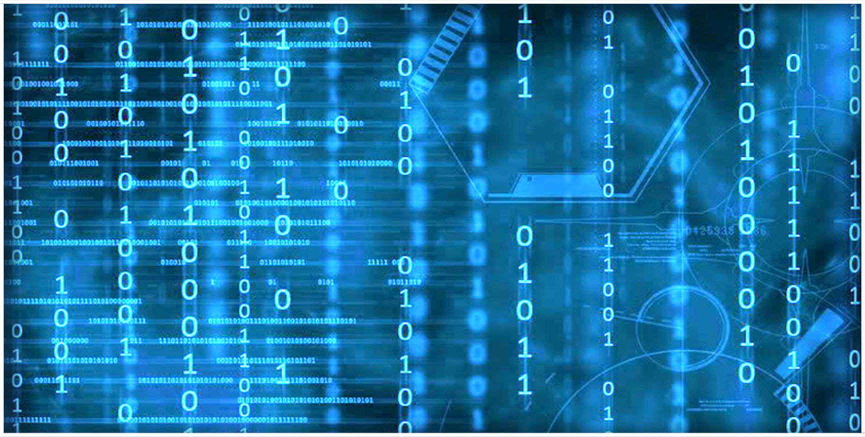
IJACSA

WHERE WISDOM SHARES

International Journal of Advanced Computer Science and Applications

Volume 14 Issue 3

March 2023



ISSN 2156-5570(Online)
ISSN 2158-107X(Print)

SAI

www.ijacsa.thesai.org

Editorial Preface

From the Desk of Managing Editor…

It may be difficult to imagine that almost half a century ago we used computers far less sophisticated than current home desktop computers to put a man on the moon. In that 50 year span, the field of computer science has exploded.

Computer science has opened new avenues for thought and experimentation. What began as a way to simplify the calculation process has given birth to technology once only imagined by the human mind. The ability to communicate and share ideas even though collaborators are half a world away and exploration of not just the stars above but the internal workings of the human genome are some of the ways that this field has moved at an exponential pace.

At the International Journal of Advanced Computer Science and Applications it is our mission to provide an outlet for quality research. We want to promote universal access and opportunities for the international scientific community to share and disseminate scientific and technical information.

We believe in spreading knowledge of computer science and its applications to all classes of audiences. That is why we deliver up-to-date, authoritative coverage and offer open access of all our articles. Our archives have served as a place to provoke philosophical, theoretical, and empirical ideas from some of the finest minds in the field.

We utilize the talents and experience of editor and reviewers working at Universities and Institutions from around the world. We would like to express our gratitude to all authors, whose research results have been published in our journal, as well as our referees for their in-depth evaluations. Our high standards are maintained through a double blind review process.

We hope that this edition of IJACSA inspires and entices you to submit your own contributions in upcoming issues. Thank you for sharing wisdom.

Thank you for Sharing Wisdom!

Kohei Arai
Editor-in-Chief
IJACSA

Volume 14 Issue 3 March 2023
ISSN 2156-5570 (Online)
ISSN 2158-107X (Print)

Editorial Board

Editor-in-Chief

Dr. Kohei Arai - Saga University

Domains of Research: Technology Trends, Computer Vision, Decision Making, Information Retrieval, Networking, Simulation

Associate Editors

Alaa Sheta
Southern Connecticut State University

Domain of Research: Artificial Neural Networks, Computer Vision, Image Processing, Neural Networks, Neuro-Fuzzy Systems

Domenico Ciuonzo
University of Naples, Federico II, Italy

Domain of Research: Artificial Intelligence, Communication, Security, Big Data, Cloud Computing, Computer Networks, Internet of Things

Dorota Kaminska
Lodz University of Technology

Domain of Research: Artificial Intelligence, Virtual Reality

Elena Scutelnicu
"Dunarea de Jos" University of Galati

Domain of Research: e-Learning, e-Learning Tools, Simulation

In Soo Lee
Kyungpook National University

Domain of Research: Intelligent Systems, Artificial Neural Networks, Computational Intelligence, Neural Networks, Perception and Learning

Krassen Stefanov
Professor at Sofia University St. Kliment Ohridski

Domain of Research: e-Learning, Agents and Multi-agent Systems, Artificial Intelligence, e-Learning Tools, Educational Systems Design

Renato De Leone
Università di Camerino

Domain of Research: Mathematical Programming, Large-Scale Parallel Optimization, Transportation problems, Classification problems, Linear and Integer Programming

Xiao-Zhi Gao
University of Eastern Finland

Domain of Research: Artificial Intelligence, Genetic Algorithms

CONTENTS

An Ontology-driven DBpedia Quality Enhancement to Support Entity Annotation for Arabic Text

Adham Kahlawi
Department of Statistics, Computer Science Applications, University of Florence, Florence, Italy

***Abstract*—Improving NLP outputs by extracting structured data from unstructured data is crucial, and several tools are available for the English language to achieve this objective. However, little attention has been paid to the Arabic language. This research aims to address this issue by enhancing the quality of DBpedia data. One limitation of DBpedia is that each resource can belong to multiple types and may not represent the intended concept. Additionally, some resources may be assigned incorrect types. To overcome these limitations, this study proposes creating a new ontology to represent Arabic data using the DBpedia ontology, followed by an algorithm to verify type assignments using the resource's title metadata and similarity between resources' descriptions. Finally, the research builds an entity annotation tool for Arabic using the verified dataset.**

Keywords—Entity annotation; semantics annotation; DBpedia; Arabic language; ontology; semantic web; linked open data

I. Introduction

The volume of unstructured text data increases daily, which increases the need for methodologies to help understand and classify this information. Since such data is not structured, the best way to understand it is by adding a metadata to it; consequently, it can be converted to semi-structured data. Part of Speech (POS) tagging, morphological annotation, structural annotation, pragmatic annotation, syntactic annotation, semantic annotation, pragmatic annotation, and stylistic annotation are several methodologies used in Natural Language Processing (NLP) to improve the results of the text analyze. One of the biggest sources of text data that is edited collaboratively from different users is Wikipedia [1]; furthermore, it covers a large number of topics, and it is a free information source. As a consequence, a specific type of semantic annotation has been involved based on Wikipedia called wikification [2]. Thus, Wikipedia has become training data for multiple models and not just a source of information. DBpedia is a project to convert Wikipedia data into structured data and make this data not only available on the web but also machine-readable [3]. DBpedia is available in different languages with cross-language being available to identify the same concept. In other words, the semantic structure of DBpedia allows the connection between more than one label written in more than one language with the same concept. One of these languages is Arabic, which makes DBpedia a good choice to be the information source to build entity annotation for Arabic text. Nonetheless, the transformation process that DBpedia underwent caused some problems, such as the use of the Wikipedia category system does not form a complete topical classification because there exist cycles between the categories, sometimes representing a loose connection between the Wikipedia articles [4]. Consequently, DBpedia data is not high quality. These problems can be summarized into two main categories. The first category, the types to which the DBpedia resources[1] belong to were extrapolated from the DBpedia ontology, but this ontology contains a large number of classes and subclasses. Thus, the single resource can belong to more than one type; meanwhile, these types do not necessarily refer to the same concept. In the second category, the process of assigning types to resources was not accurate enough; therefore, can find some resources which were assigned a type that does not correspond to their concept, or some resources which were assigned more than one type but at least one of these types does not correspond to their concept.

To build an efficient tool, the quality of the data taken from DBpedia has to be improved by building a methodology from several steps. The methodology begins from building a new ontology inspired by the DBpedia ontology, up to validating the information through the content of the text itself or by using the measurement of similarity between texts.

The rest of this paper is structured as follows; Section II discusses the related work. Section III shows what DBpedia is and how it is built. Section IV explains how to obtain data from DBpedia. Section V describes the data collected. Section VI explains the methodology that was followed in this article. Section VII evaluates the methodology. Finally, Section VIII concludes the paper.

II. Related Work

The ability to give metadata to text data has been established as a strategic task for understanding and analyzing texts. Cucerzan [5] seeks through his large-scale system for the recognition and semantic disambiguation to name entity by using data extracted from Wikipedia. Through a link between a single information presented in Wikipedia with the document in which presented and with the category tags of this document. This system passes through three stages. The first stage is to surface from entity mappings, which use the information of the titles of entity pages, the titles of redirecting pages, the disambiguation pages, and the references to entity pages in other Wikipedia articles. The second stage is category information, which takes advantage of the classifications built by Wikipedia contributors. The third stage is contexts, which use the information present in the entity page and other pages

[1] A resource is a basic object encapsulating a "real" thing like a text, image or whatever. This object is accessible within the server according to its Uniform Resource Identifier (URI) because each URI is unique at the global level. All the information that describes the resource itself is stored using the Resource Description Framework (RDF).

that explicitly refer to the same entity. Mihalcea and Csomai [6] integrate two algorithms into one system called Wikify, where the first algorithm identifies and extract the important words in the input text, and the second algorithm assigns each word of the output of the first algorithm with the correct page of Wikipedia. According to Milne and Witten [2], the Wikify system will make mistakes because of its dependence on a probabilistic value of word link that is calculated by the number of a link of the same word in different articles on Wikipedia with a specific article. For these reasons, they developed a new system that takes into consideration not only the word but also the context surrounding this word.

On the other hand, Kulkarni and others [7] proposed a new system that is not directly human interpretable, but downstream indexing, search, and mining. This system is an optimization of the previous one through investigating practical solutions based on local hill-climbing, rounding integer linear programs, and pre-clustering entities followed by local optimization within clusters. TAGME is a software system proposed by Ferragina and Scailla [8] which addresses the problem of cross-referencing text fragments with Wikipedia pages. In particular, TAGME took advantage of previous work and assigned it to small texts such as tweets. Makris et al. [9] proposed techniques that enhanced the TAGME to be applied to different approaches for Wikipedia disambiguation. These techniques improve the quality of Wikipedia by auxiliary information provided by more formal knowledge resources like WordNet [10], which is a large lexical network where concepts/senses are represented by so called synsets. In addition, these techniques employed the PageRank of the Wikipedia pages as an extra factor for disambiguation. Wikifier is a semantic annotation approach presented by Brank and others [11], which supports different languages that are available in Wikipedia; furthermore, it is suitable for parallel processing and supports various minor heuristics. Wikifier refinements is an effort to improve the performance of other approaches; indeed, it uses Wikipedia class membership to ignore certain types of concepts and use the indicator of word frequency to ignore the common words. Consequently, Wikifier reduces the noise in the annotation output. WEXEA, a Wikipedia EXhaustive Entity Annotation system [12] proposed by Strobl and others. the approach aims to create an annotated text corpus include all mentions in Wikipedia instead of simply depend on already existing links between Wikipedia pages; as a consequence, that can be more useful in downstream tasks and can introduce unnecessary errors. There have been several attempts to build tools for the Arabic language. In the following, we will list the most important of these attempts. In 2008 Benajiba et al. [13] used contextual, lexical, part-of-speech, and other features to train a Named Entity Recognition model using an SVM-based approach. The model can recognize four (Person, Location, Organization, and Miscellaneous). Next, in 2012 Al-Jumaily et al. [14] built a real-time named entity recognition system using the data from DBPedia and other sources. The system can recognize three entities only (Person, Location and Organization); the system can also extract the linguistic roots (nouns and verbs). Then, in 2014 Yosef et al. [15] used the Wikipedia Arabic version to create a Named entity disambiguation framework. The framework structure contains four principal concepts entity repository, name entity dictionary, entity description and entity-entity relation. Meanwhile, the framework uses the connection between the Arabic version and the English one of Wikipedia to control the data quality. To test the framework, the authors chose ten news Arabic articles and manually annotated them. After that, in 2015, Al-Yahya et al. [16] presented a lexical semantic annotation based on an ontology that contains six classes that build a three-level hierarchical structure. The higher level is Linguistic Concepts, and the lower level is Words. Consequently, in 2016, Al-Qawasmeh et al. [17] created a similar tool based on DBPedia, where they made an ontology consisting of three classes and filled it in the DBPedia's individuals that belong to these three classes. The authors use a similar algorism to optimize the performance where the algorism finds the nearest entity from the dataset to the input entity. Finally, in 2018, Albukhitan et al. [18] built a tool using deep learning to recognise three entities (Food, Nutrition and Health). The tool will assign the closest ontological class to the input text as a named entity using the weighted candidate vectors and word2vec model. Jarrar et al. [29] introduce Wojood, a corpus for nested Named Entity Recognition in Arabic, consisting of 550K tokens manually annotated with 21 entity types, including nested entities, with a strong inter-annotator agreement. The corpus was used to train a nested NER model achieving a micro F1-score of 0.884, and all resources are publicly available. Al-Thubaity et al. [30] present the COVID-19 Arabic Named Entities Recognition (CAraNER) dataset, consisting of 55,389 tokens from Saudi Arabian newspaper articles labelled with five named-entity tags. The paper also evaluates the dataset using four BERT-based Arabic language models, with AraBERTv0.2-large achieving the highest F1 macro measure of 0.86.

III. Baseline: Dbpedia

Wikipedia is one of the large knowledge sources, and is created by the contribution of thousands of users through writing the articles using natural languages. On the other hand, Wikipedia contains info box, images, geo-coordinates, and categorization information which can be considered as structured information. Moreover, this structure information has been presented by different languages [19]. The structured information of Wikipedia was the basis for building the DBpedia through an open-source extraction framework; as a consequence, DBpedia has been considered as a multilingual and multidomain knowledge base. Each concept of DBpedia is described by a corresponding Wikipedia page and is identified by a Uniform Resource Identifier (URI) [20]. The use of URI and the Resource Description Framework (RDF) has made the DBpedia relatively stable, machines readable, and commonly used ontology. The DBpedia ontology can be used in the integration of different languages and the organization of extracted data [21]. Lehmann and others [22] illustrated the process of data extraction from Wikipedia through Fig. 1. The data extraction process passes through several stages that start from the inputs which are Wikipedia pages; secondly, it moves to the parsing stage that transforms the pages into an Abstract Syntax Tree; thirdly, the extraction stage which converts different parts of pages to RDF by following the next steps:

- the creation of DBpedia ontology terms like the data property through the mapping of infobox structure manually;
- mapping the information in the infobox to RDF taking into consideration the DBpedia ontology;
- extracting a single feature from pages such as a label or geographic coordinates;
- extracting aggregated data from all pages like word counts as further description.

Furthermore, to improve the DBpedia by adding the Arabic language, Ismail et al. [23][23] [23] made special data extraction for Arabic Wikipedia pages through the mapping of infoboxes to the DBpedia ontology by different mapping extractors.

The data extraction based on infoboxes was of great importance in the automation of extraction; on the other hand, it causes some quality issues. Consequently, several papers have been published discussing these issues. For instance, the quality problems resulting from incorrect or missing information like incorrect values, data types, and links were discussed by Zaveri et al. [24], several automatic quality tests were provided by Kontostas et al. [25] to be applied to Linked Open Data (LOD) dataset of DBpedia, the DBpedia accessibility quality was tested by a Linked Data Quality Model that was developed by Radulović et al. [26], and the use of machine learning was proposed by Rico et al. [27] for the detection of incorrect mappings. Lakshen et al. [28] focused on identifying the quality of Arabic DBpedia since it contains problems different from the problems of other languages like the presentation of characters as symbols, the use of Hindu numerals that can create wrong values in numerical data, and occurrence of different names for the same attribute.

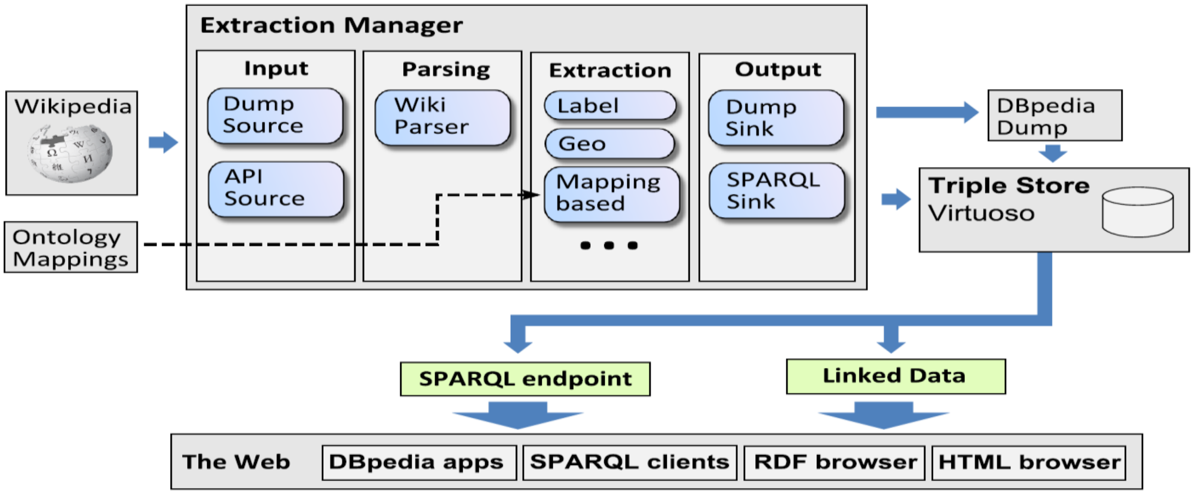


Fig. 1. Overview of DBpedia extraction framework [15].

IV. Data Collection

The latest releases of core data from en.wikipedia.org have been published by the DBpedia protect on the first of July 2020; also the collection URI is https://databus.dbpedia.org/dbpedia/collections/latest-core. However, the collection does not contain downloadable files for the Arabic language; for this reason, the data has been acquired by applying SPARQL Query at a DBpedia public SPARQL endpoint “http://dbpedia.org/sparql” using python function.

```
        def apply_query(query):
            sparql=
SPARQLWrapper("http://dbpedia.org/sparql")
            sparql.setQuery(query)
            sparql.setReturnFormat(JSON)
            results = sparql.query().convert()
            df=
pd.json_normalize(results["results"]["bindings"])
            return df
```

public SPARQL endpoint has a limit as the following[31] :
“ *ResultSetMaxRows = 10000*
MaxQueryExecutionTime = 120 (seconds)
MaxQueryCostEstimationTime = 1500 (seconds)
Connection limi = 50 (parallel connections per IP address)
maximum request rate = 100 (requests per second per IP address, with an initial burst of 120 requests)”

As a consequence, the data was collected following the steps below:

a) identify the types of all the instances.

The DBpedia collection offers a special file that contains the type of all instances.

b) building the DBpedia Arabic dataset.

At this point, a dataset containing instances URI, label, description, and type will be obtained by sending a SPARQL query to DBpedia SPARQL endpoint through a specific Python code. The SPARQL query has been built and has been sent for each type obtained in the previous step.

where_text = """ ?uri rdfs:label ?label;

```
                              rdf:type   ?type.
                        optional{}
                       filter(lang(?label)="ar")
                     filter(?type in(<{}>))
    """
optional_text = """{
                       ?uri rdfs:comment ?descrption.
                      filter(lang(?descrption)="ar")
                     }"""
for i in range(0,len(type_list)):
  arabic_dataset_query = f"""PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  SELECT distinct ?uri ?label  ?descrption ?type WHERE {'{'}
  {where_text.format(optional_text,type_list[i])}
  {'}'}
  """
  arabic_label_df=
arabic_label_df.append(apply_query(arabic_datset_query))
```

V. Data Description

In this chapter, we will discuss the validity of the dataset from two different points of view. Firstly, the dataset description and the affiliations of each resource; secondly, the formal and grammatical structure of the resources' label.

A. Description of the Dataset obtained from DBpedia

As shown in Fig. 2, the number of URI of the resources without repetition within the dataset is greater than the number of resources label without repetition. As a consequence, this information indicates the problem of the existence of a group of resources within the dataset that have the same name but have more than one URI.

On the other hand, the total data in the dataset is greater than the number of resources label without repetition. For this reason, there is a problem of the presence of some resources that belong to more than one type; in fact, Fig 3 shows that resources may belong to more than one type at the same time and the max is seven types for example: this URI http://dbpedia.org/page/Pomegranate_soup refer to Pomegranate soup resource that has two different types, the first one is dbo:Food but the second one is dbo:GivenName that usually use to indicate a person name. However, it is not necessary that these different types refer to different concepts; nonetheless, they may refer to the same concept with different names or refer to the sub-concepts of the same concept.

B. The Formal and Grammatical Structure

Arabic labels contain additional information placed in brackets to clarify the meaning of the label; on the other hand, this method of adding additional information is not considered as one of the common methods used in the Arabic language to attend this goal. For example, the URI http://dbpedia.org/resource/Pato has the Arabic label written in this way "باتو(رياضة)" which means Pato (sport); however, the English label is written with only the word Pato. The dataset contains 63747 Arabic labels contain additional information placed in brackets.

As for the linguistic rules, the rules of definite or indefinite articles in the Arabic language is different from other languages like English. The English language uses the articles that are placed before the noun, indicating whether the word is definite or indefinite. In contrast, the Arabic language adds two letters to the word at the beginning to differentiate between definite or indefinite nouns; consequently, the natural language processing will consider the definite word different from the same word in the indefinite form.

For instance, the URI that has Cartilage as English label, this label can be written inside any English text as "a cartilage" or "the cartilage"; in contrast, the Arabic label of this URI is غضروف that can be written inside any Arabic text as "غضروف" or "الغضروف".

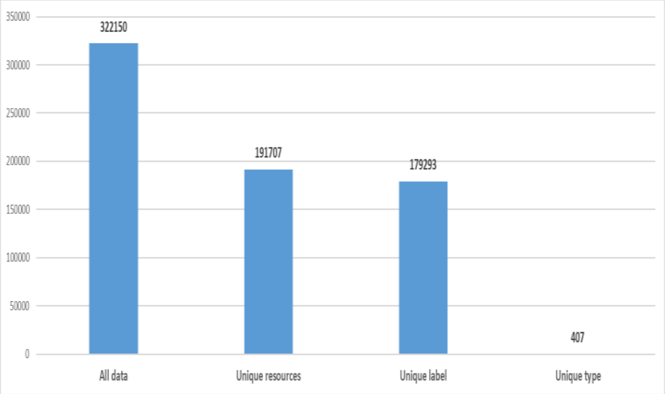


Fig. 2. Dataset description.

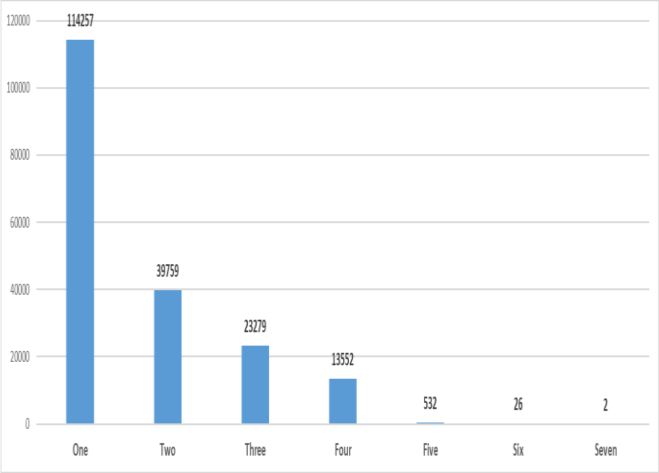


Fig. 3. Frequency of resources depending on the number of types.

VI. Methodology

The methodology seeks to reduce the number of types that some resources can belong to; in addition, seeks to verify the validity of the types that have been assigned to the resources. Finally, the data obtained from achieving the two previous goals will be used to build a tool to annotate the Arabic text. Fig. 4 shows that the methodology depends on the implementation of a set of successive steps to achieve its objective, which are described as follows:

A. Develop a New Ontology to Improve Data Quality

Each resource in the data belongs to one or more types as shown in Fig. 3; indeed, these types belong to DBpedia ontology. By referring to DBpedia ontology, it can be seen that the group of types to which the resource belongs does not necessarily belong to different concepts. However, these types may belong to the same general concept; consequently, it represents one of the oldest data quality problems faced by machine learning technologies [32]. Therefore, this study proposes to create a new ontology inspired by the DBpedia ontology and correspond to the types of Arabic dataset in order to improve the quality of data and reduce the number of types to which belongs every resource [33][34]. Afterward, the basic types to which the data belong were transformed into individuals belonging to the classes of new ontology that represent the general concept of it. We must point out that building the new ontology depends on human experience; therefore, it will be a manual process, while the process of individual transformation for it will be automatic. Fig. 5 represents the DBpedia Arabic resources ontology, while Table I represents its metrics.

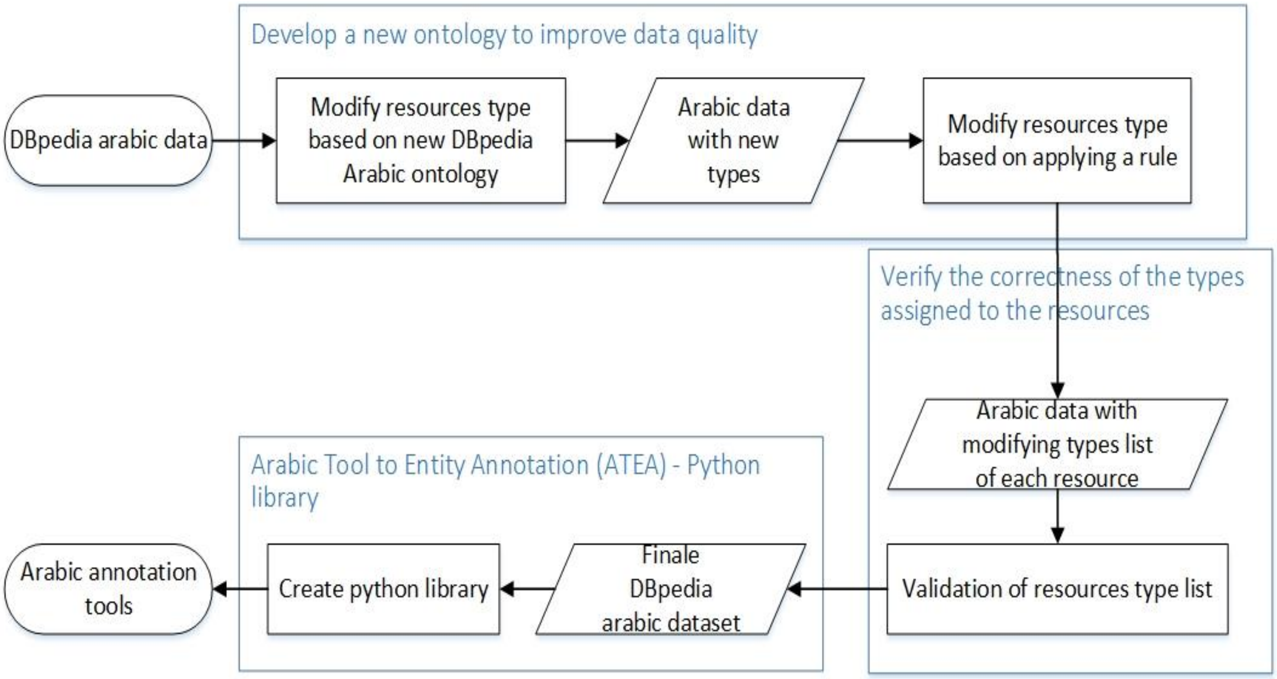

Develop a new ontology to improve data quality
DBpedia arabic data
Modify resources type based on new DBpedia Arabic ontology
Arabic data with new types
Modify resources type based on applying a rule
Verify the correctness of the types assigned to the resources
Arabic data with modifying types list of each resource
Validation of resources type list
Arabic Tool to Entity Annotation (ATEA) - Python library
Finale DBpedia arabic dataset
Create python library
Arabic annotation tools

Fig. 4. The process underlying our methodology.

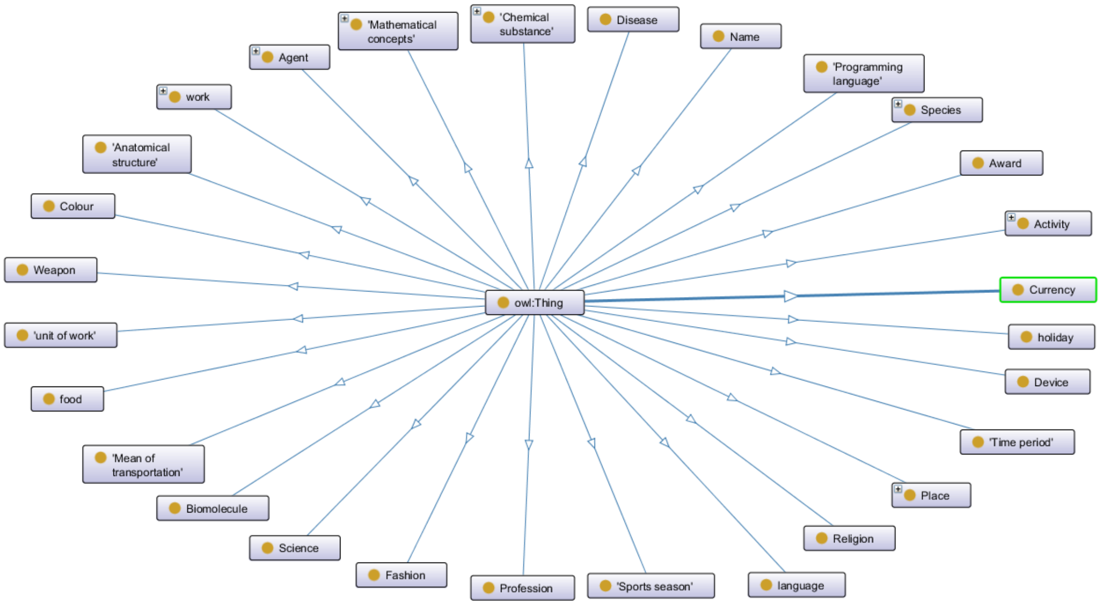


Fig. 5. DBpedia Arabic resources ontology.

TABLE I. DBPEDIA ARABIC RESOURCES ONTOLOGY METRICS

Ontology metrics	description
Metrics	
Axiom	240
Logical axiom count	106
Declaration axioms count	134
Class count	134
Class axioms	
SubClassOf	**106**

As a consequence of applying the previous mapping process to the data, each element belongs to a new type or group of types. Based on the new ontology, the following rule was applied to reduce the number of the types to which each source belongs.

Rule:

If there are two types one of which is a subclass of the other in the group of types to which a resource belongs; consequently, the type that represents a subclass is retained and the other is deleted from the group of types to which this resource belongs.

B. Verify the Correctness of the Types Assigned to the Resources

The DBpedia data is considered as structure data that was built through a process of mapping from Wikipedia; nonetheless, this process caused some errors in assigning the correct type to some resources. For this reason, this study proposed an algorithm to verify the validity of the type assigned to each resource and correct it in the event that an error is discovered. This algorithm is shown in Fig. 6 based on thirteen actions that are described as follows:

- Action 1 - Create URI_Type dictionary: A subset will be extracted from the dataset which contains the URI of each resource and its type or their types.
- Action 2 - Create URI_label dictionary: A subset will be extracted from the dataset which contains the URI of each resource and its Arabic label.
- Action 3 - Extract words in parentheses: The words written between parentheses in some labels of resource will be extracted.
- Action 4 - Associate words with the type they refer to: The list of words that result from the previous action will be checked manually if it refers to a specific type; then, a dataset of words and the types that refer to will be built.
- Action 5 - Assign to the resource URI the extracted word from parentheses presented in its label: Will build a dataset of URI of each resource and the extracted word written between parentheses in the label of resources, or the value of Null if the label does not include words between parentheses.
- Action 6 - Assign the word's type to the URI: The sources whose label contains an explanatory word and this word refer to a specific type. In this action, this type will be assigned to the resource URI.
- Action 7 - Create URI_Description dictionary: A subset will be extracted from the dataset which contains the URI of each resource and its English description.
- Action 8 - Calculate the similarity between the descriptions: The similarity will be calculated using the Latent Semantic Indexing (LSI) model. LSI is an indexing and retrieval technique. LSI is a text mining that is able to calculate the similarity between the text data by projecting it into space with latent semantic dimensions. In other words, the similarity is calculated by the co-occurrence of each word and every single word in the documents. The measure ranges from 0 to 1 (the greater the more similar)[35, 36]. LSI model was trained by using the English description of each resource to calculate the percentage of similarity between the resources, and then this dataset was constructed by linking each resource with the best five resources similar to it.
- Action 9 - Get for each URI the type of the best five similar URI: The type of five most similar resources of each resource will be gotten; indeed, the type of similar resources will be gotten from two sources. The first "verified URI_type dictionary" will be used if a similar resource is one of the resources whose type has been verified in the action 6. The second "URI_type dictionary" will be used if a similar resource is one of the resources that have not been verified its type yet.
- Action 10 - Get the type list for each URI: the type list of each resource will be gotten.
- Action 11 - Calculate the intersection between the two list: The intersection value between the two lists resulted in action 9 and action 10 will be calculated.
- Action 12 - Assign the intersection values as type to the URI: The intersection value will be assigned to the resource URI as a type for it.
- Action 13 - Human verification to the URI type: All resources that were the results of the calculated intersection in the action 11 are Null, its type will be verified manually. Manual verification faces two cases. The first case, the type that was assigned to the resource is correct, so no action will be taken. The second case, the type that was assigned to the resource is not appropriate for it, so the type will be changed to a type that corresponds to the nature of the resource.

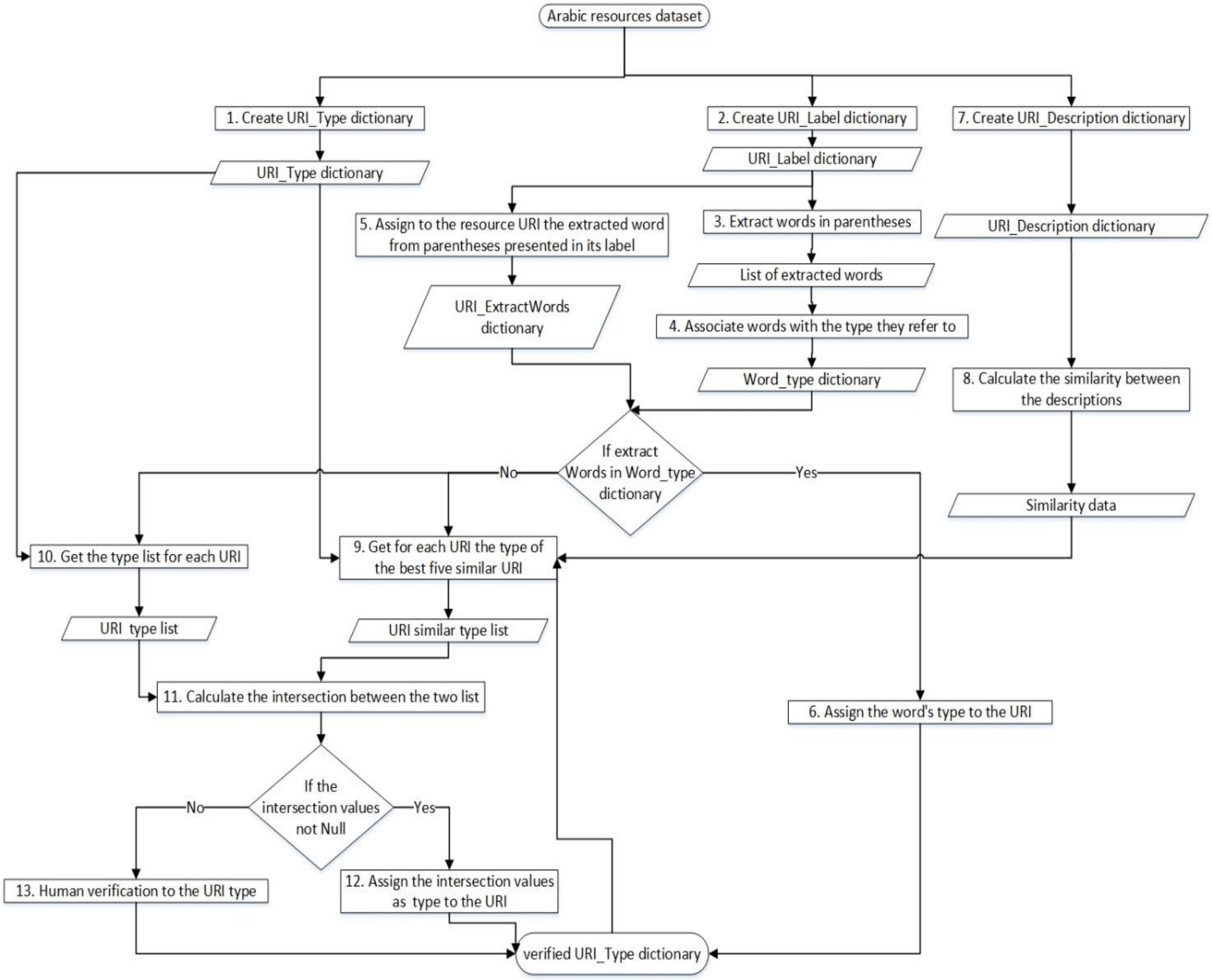

Arabic resources dataset
1. Create URI_Type dictionary
2. Create URI_Label dictionary
7. Create URI_Description dictionary
URI_Type dictionary
URI_Label dictionary
5. Assign to the resource URI the extracted word from parentheses presented in its label
3. Extract words in parentheses
URI_Description dictionary
List of extracted words
URI_ExtractWords dictionary
4. Associate words with the type they refer to
Word_type dictionary
8. Calculate the similarity between the descriptions
If extract Words in Word_type dictionary
No
Yes
Similarity data
10. Get the type list for each URI
9. Get for each URI the type of the best five similar URI
URI type list
URI similar type list
11. Calculate the intersection between the two list
6. Assign the word's type to the URI
If the intersection values not Null
No
Yes
13. Human verification to the URI type
12. Assign the intersection values as type to the URI
verified URI_Type dictionary

Fig. 6. An algorithm to verify the correctness of the types assigned to the resources.

C. *Arabic Tool to Entity Annotation (ATEA) - Python Library*

Fig. 7 illustrates the mechanism of ATEA library's work as it takes short text lake input; in contrast, its output contains different types of data. The first one represents the DBpedia Arabic resource ontology's types that belong to each entity detected in the text. The second type represents the DBpedia URI and DBpedia types that belong to each entity detected in the text. The last type represents the interpretation of the label of each entity detected in the text to different languages available in DBpedia.

ATEA includes four data set:

- URI_Label dataset: This dataset contains the entire Arabic resources label after applying two adjustments. First, all additional information attached to the primary label has been deleted; second, all the labels have been stemmed.
- Verified URI_Type dataset: This dataset contains all the resources types; indeed, these types follow the DBpedia Arabic resources ontology.
- URI_Type dataset: This dataset contains all the resources types obtained directly from DBpedia.
- This dataset contains the English label of all resources included in URI_Label dataset.

ATEA performed seven actions as following:

- Action 1 - Split text: This action is divided into two steps; first one is splitting the text based on the punctuation marks. The second step is tokenizing each subtext to all possible n gram word where n is between one and the number of words of the longest label in DBpedia Arabic resources.
- Action 2 - Find all entities: The intersection between the label list of DBpedia Arabic resources and the tokenization list will be calculated.
- Action 3 - Get the DBpedia Arabic resource ontology's type of each entity: Each element in the intersection list will be assigned to its type of the DBpedia Arabic resource ontology.

- Action 4 - Get the DBpedia URI and types of each entity: Each element in the intersection list will be assigned to its type and URI of the DBpedia.
- Action 5 - Interpret each entity to English language: Will be assigned to each element in the intersection list the English label of its represented resource in DBpedia.
- Action 6 - Integration of the three categories of data: The results of each item in the intersection list will be combined and organized into a dictionary that will represent the output of ATEA.

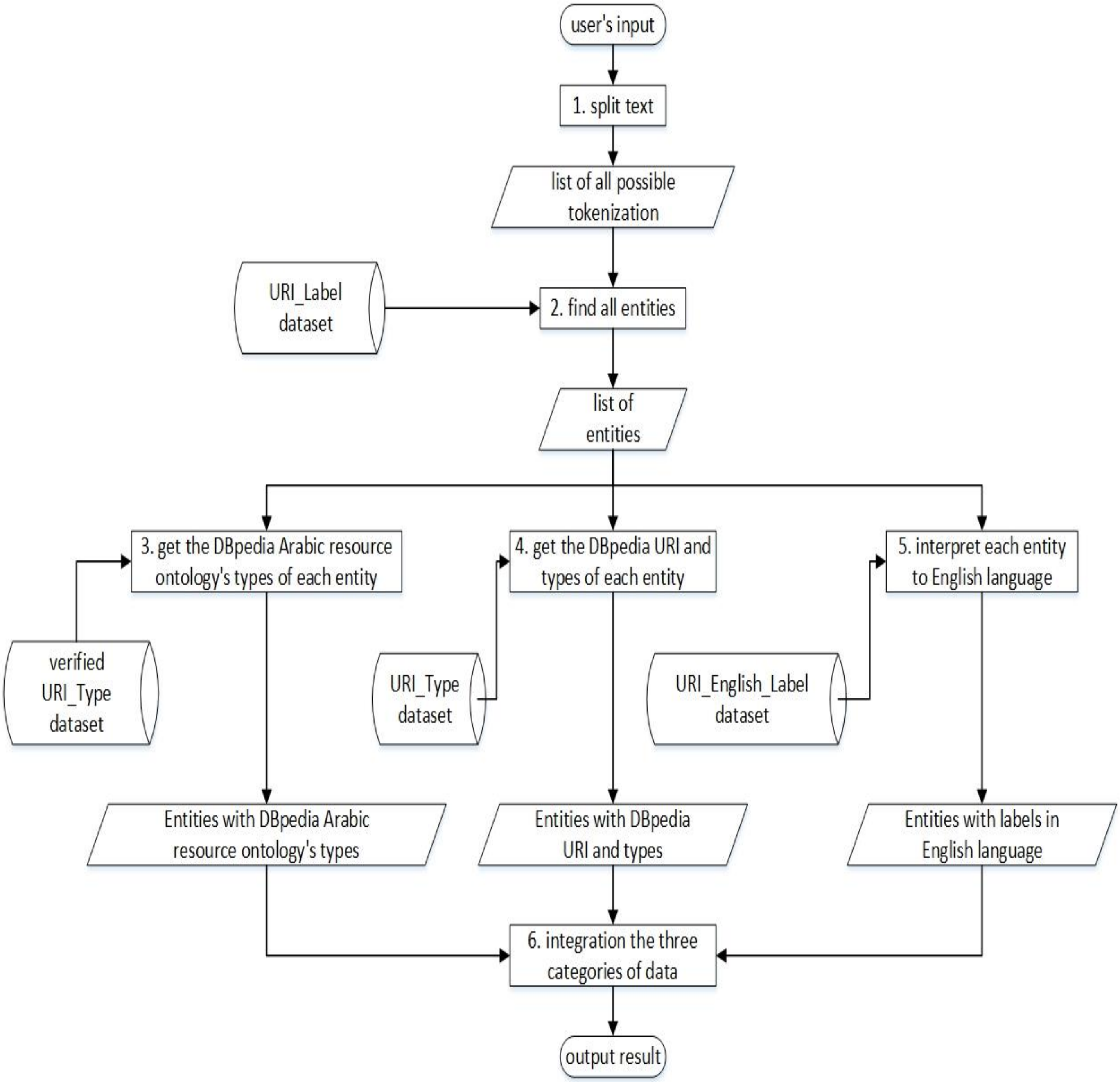

user's input
1. split text
list of all possible tokenization
URI_Label dataset
2. find all entities
list of entities
3. get the DBpedia Arabic resource ontology's types of each entity
4. get the DBpedia URI and types of each entity
5. interpret each entity to English language
verified URI_Type dataset
URI_Type dataset
URI_English_Label dataset
Entities with DBpedia Arabic resource ontology's types
Entities with DBpedia URI and types
Entities with labels in English language
6. integration the three categories of data
output result

Fig. 7. Arabic tool to entity annotation.

VII. Evaluation

Arabic resources collected from DBpedia had 407 different types and the resource may have up to seven types; consequently, each resource belongs to 1.68 types on average. The use of our developed DBpedia Arabic resources ontology reduced the number of types that the resources could belong to; as a result, DBpedia Arabic resources ontology grouped the types that belong to one concept together; thus, the number of types was reduced from 407 to 106. Indeed, Fig. 8 shows that the max number of types that any resource could belong to is five, and each resource belongs to 1.33 types on average. On the other hand, the use of DBpedia Arabic resources ontology left 739 resources without type, because these resources were related to the "Thing" type such as the resource of Whale that has the URI "http://dbpedia.org/resource/Whale". In addition, the implementation of the rule resulted in the max number of types that any resource could belong to reduce to three and each resource belongs to 1.08 types on average. At the end of the first step of the methodology, it is noted the optimization in the number of types that one resource can belong to; in fact, about 92% of the resources belong to one type.

In the second step of the methodology, firstly, 1688 words indicating specific types were extracted from information contained in parentheses in Arabic labels. Examples of these words are Novel, Book, Journal, Writer, Poet, Rocket, Tank, TV series, wrestling, Scientist, king, Restaurant, Church, programming language, Plant and bird. These words were used to verify the validity of the types to which the resources belong; indeed, if the Arabic label of any resource contains a word indicating a specific type and this type does not correspond to the resource type, the source type is modified to the type to which the word refers, as shown in Table II.

Secondly, the similarity between the descriptions of the resources was used. If the resource type corresponds to at least one of the five types of sources that are most similar to it; consequently, the resource type is considered correct. Otherwise, it will be manually verified. Table III shows some resources that have been verified manually.

As a result of the second step of the methodology, the max number of types that any resource could belong to is three types, but each resource belongs to 1.01 types on average. Moreover, about 99% of the resources belong to one type.

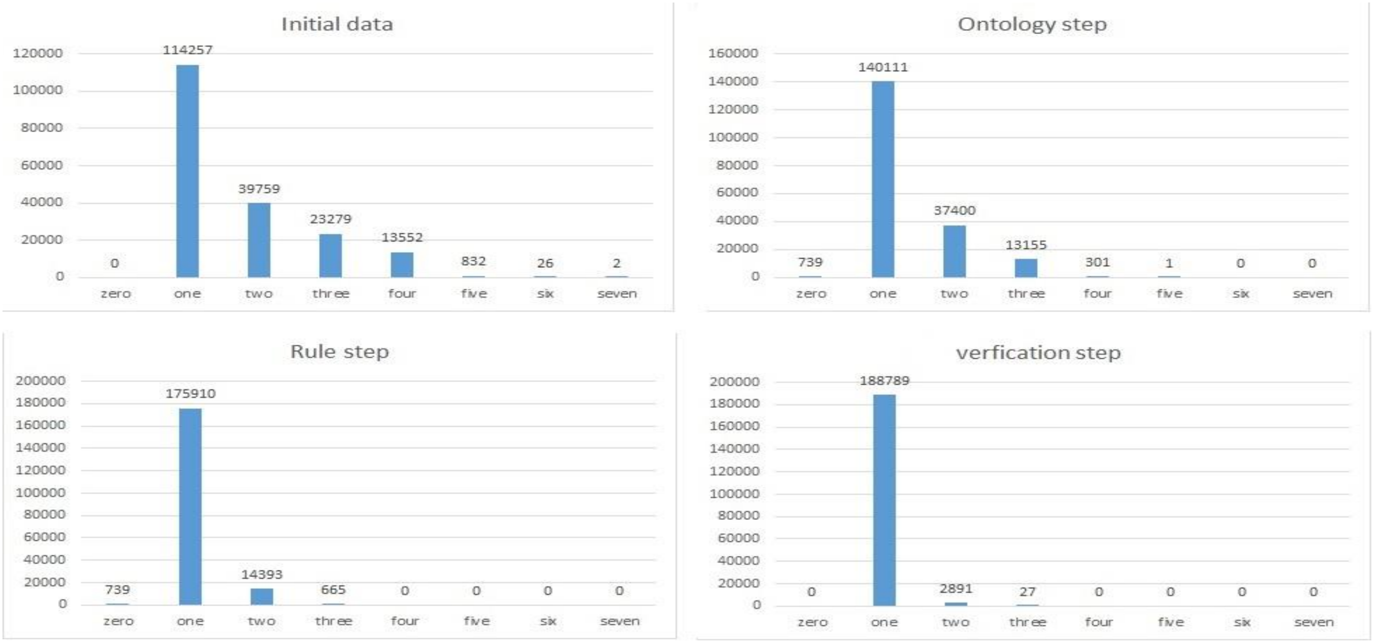


Fig. 8. Frequency of resources depending on the number of types that belong to for each methodology's step.

TABLE II. Examples of Text Verification Results

Resource URI	Arabic label	English label	DBpedia type	verify type
http://dbpedia.org/resource/Anthony_Wong_(Hong_Kong_actor)	أنتوني وونغ (ممثل هونغ كونغي)	Anthony Wong (Hong Kong actor)	Artist, Film	Artist
http://dbpedia.org/resource/Antalya_Province	أنطاليا (محافظة)	Antalya Province	PopulatedPlace, Athlete	PopulatedPlace
http://dbpedia.org/resource/Bell_AH-1Z_Viper	فايبر (مروحية)	Bell AH-1Z Viper	MeanOfTransportation	Weapon
http://dbpedia.org/resource/George_Hamilton_(actor)	جورج هاميلتون (ممثل)	George Hamilton (actor)	Film	Artist
http://dbpedia.org/resource/Ballade_(classical_music)	بالاد (موسيقى كلاسيكية)	Ballade (classical music)	Thing	MusicalWork

TABLE III. EXAMPLES OF HUMAN VERIFICATION RESULTS

Resource URI	**Arabic label**	**English label**	**DBpedia type**	**verify type**
http://dbpedia.org/resource /Orient_News	أورينت نيوز	Orient News	Band, Company	Media
http://dbpedia.org/resource/König-class_battleship	بارجة فئة كونيغ	König-class battleship	Band, MeanOfTransportation	Weapon
http://dbpedia.org/resource/ Sovereign_Military_Order_of_Malta	فرسان مالطة	Sovereign Military Order of Malta	EducationalInstitution, PopulatedPlace	MilitaryUnit
http://dbpedia.org/resource/Gezer	تل الجزر	Gezer	Event, MilitaryConflict	HistoricPlace
http://dbpedia.org/resource/ Burgundians	برغنديون	Burgundians	Language, Name	EthnicGroup
http://dbpedia.org/resource/Sabines	سابينيون	سابينيون	Language, Animal	EthnicGroup
http://dbpedia.org/resource/ Amr_Sobhy	عمرو صبحي	Amr Sobhy	Award, Writer	Writer
http://dbpedia.org/resource/Whale	حوت	Whale	Thing	Animal
http://dbpedia.org/resource/Narcotic	ناركوتي	Narcotic	Thing	Drug
http://dbpedia.org/resource/ Tank	دبابة	Tank	Thing	Weapon
http://dbpedia.org/resource/ Beef	لحم بقر	Beef	Thing	Food
http://dbpedia.org/resource/ Operetta	أوبريت	Operetta	Thing	Play
http://dbpedia.org/resource/ Train	قطار	Train	Thing	MeanOfTransportation

In the third step of the methodology, ATEA is built which takes short text as its input and then performs the following steps:

Firstly, the length of the text is measured, if the length of the text is equal or greater than 17 words, the text will be divided into n_ gram where n = 17; because of the max length Arabic label present in the data equal to 17. On the other hand, if the length of the text is less than 17 then the text will be divided into n_ gram where n equal to the length of the text.

Secondly, all n_gram obtained from the previous step are searched for; afterward, if any of them was found in the database. First of all, it will be annotated then it will be deleted from the text; last of all, if n equal to one, the finale results will be returned to the user; in contrast, if n greater than one, the previous steps will be repeated. This is the application of your method to an example. The example talks about Syria.

“ سورية دولة عربية يبلغ عدد سكانها ما يقارب 23 مليون منهم 93% عرب و 5% كرد و 2% أعراق أخرى مثل أرمن و تركمان. يعتنق السوريون أكثر من ديانة فنجد أن 90% إسلام و 7% مسيحية و 3% طوائف أخرى. يتكلم غالبية السكان لغة عربية. . تعاقبن على سوريا حضارات ممتالية ومن أشهرها إبلا وكانت هذه الحضارة العريقة والقوية قد ازدهرت في منتصف الألف الثالث قبل الميلاد و التي تم إكتشافها من قبل بعثة أثرية من جامعة روما سابينزا برئاسة باولو ماتييه. ”

“Syria is an Arab country with a population of approximately 23 million, of which 93% Arabs, 5% Kurds, and 2% other ethnicities such as Armenians and Turkmen. Syrians have more than one religion; indeed, we find that 90% Islam, 7% Christian, and 3% other sects. The majority of the population speaks Arabic. Successive civilizations took place in Syria, the most famous of which is Ebla. This ancient and powerful civilization had flourished in the middle of the third millennium BC, which was discovered by an archaeological mission from the University of Rome Sapienza headed by Paolo Mattei.”

The automatic entity annotation results are in JSON format as the following:

```
{
'جامعة روما سابينزا':
{'DBpedia_URI':
['http://dbpedia.org/resource/Sapienza_University_of_Rome'],
  'label': {'ar': ['جامعة روما سابينزا'],'en': ['Sapienza University of
Rome']},
  'type': {'ar': ['موسسة اكاديمية'], 'en': ['Educational institution']},
  'DBpedia_type': ['http://dbpedia.org/ontology/University',
   'http://dbpedia.org/ontology/Organisation',
   'http://dbpedia.org/ontology/EducationalInstitution']},
 'لغة عربية':
{'DBpedia_URI': ['http://dbpedia.org/resource/Arabic'],
  'label': {'ar': ['لغة عربية'], 'en': ['Arabic']},
  'type': {'ar': ['لغة'], 'en': ['language']},
  'DBpedia_type': ['http://dbpedia.org/ontology/Language']},
 'إبلا':
{'DBpedia_URI': ['http://dbpedia.org/resource/Ebla'],
  'label': {'ar': ['إبلا'], 'en': ['Ebla']},
  'type': {'ar': ['مكان تاريخي'], 'en': ['Historic place']},
  'DBpedia_type': ['http://dbpedia.org/ontology/Country']},
  'سوريا':
{'DBpedia_URI': ['http://dbpedia.org/resource/Syria'],
  'label': {'ar': ['سوريا'], 'en': ['Syria']},
  'type': {'ar': ['مكان مأهول'], 'en': ['populated place']},
  'DBpedia_type': ['http://www.w3.org/2002/07/owl#Thing',
   'http://dbpedia.org/ontology/Place',
   'http://dbpedia.org/ontology/Country',
   'http://dbpedia.org/ontology/MusicalArtist',
   'http://dbpedia.org/ontology/PopulatedPlace']},
 'عرب':
{'DBpedia_URI': ['http://dbpedia.org/resource/Arabs'],
  'label': {'ar': ['عرب'], 'en': ['Arabs']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type': ['http://dbpedia.org/ontology/EthnicGroup',
   'http://dbpedia.org/ontology/Band']},
 'تركمان':
{'DBpedia_URI': ['http://dbpedia.org/resource/Turkmens'],
  'label': {'ar': ['تركمان'], 'en': ['Turkmens']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type':
['http://dbpedia.org/ontology/EthnicGroup']},
 'أرمن':
{'DBpedia_URI': ['http://dbpedia.org/resource/Armenians'],
  'label': {'ar': ['أرمن'], 'en': ['Armenians']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type': ['http://dbpedia.org/ontology/EthnicGroup',
   'http://dbpedia.org/ontology/Band']},
 'مليون':
{'DBpedia_URI': ['http://dbpedia.org/resource/Million'],
  'label': {'ar': ['مليون'], 'en': ['Million']},
  'type': {'ar': ['وحدة قياس'], 'en': ['Measure unit']},
  'DBpedia_type':
['http://dbpedia.org/ontology/Organisation']},
'مسيحية':
{'DBpedia_URI': ['http://dbpedia.org/resource/Christianity'],
  'label': {'ar': ['مسيحية'], 'en': ['Christianity']},
  'type': {'ar': ['دين'], 'en': ['Religion']},
  'DBpedia_type':
['http://dbpedia.org/ontology/EthnicGroup']},
 'إسلام':
{'DBpedia_URI': ['http://dbpedia.org/resource/Islam'],
  'label': {'ar': ['إسلام'], 'en': ['Islam']},
  'type': {'ar': ['دين'], 'en': ['Religion']},
  'DBpedia_type':
['http://dbpedia.org/ontology/EthnicGroup']},
 'كرد':
{'DBpedia_URI': ['http://dbpedia.org/resource/Kurds'],
  'label': {'ar': ['كرد'], 'en': ['Kurds']},
  'type': {'ar': ['مجموعة عرقية'], 'en': ['Ethinc group']},
  'DBpedia_type': ['http://dbpedia.org/ontology/EthnicGroup',
   'http://dbpedia.org/ontology/Band']}
}
```

The Arabic language suffers from the scarcity of databases that can be used to evaluate the performance of such tools. In order to evaluate the performance of this tool, we will use the ANERcorp dataset [37, 38] that includes five entities (Person, Location, Organization, Miscellaneous, and others). It is necessary to point out some of the difficulties facing the evaluation process, including:

The tool provided by this work is able to distinguish more than 100 entities; therefore, we must perform a manual comparison process between the type predicted by the tool and the real type.

The database contains single words. In some cases, this single word is a part of a name of an organization or a location; thus, the assigned entity will be the same entity of the full name even if the single name refers to another entity. According to the way the proposed tool works “as shown in the previous example", the entity that the tool will predict will correspond to the nature of the single word and not the full name.

Since we will do the evaluation process manually, we have selected a sample of 2000 words from the ANERcorp dataset to complete the evaluation process. Consequently, the proposed tool could annotate words belonging to more than fifty entities, but only fifty entities contain at least three words.

We measured the annotation accuracy for each entity separately; therefore, we can say that the annotation accuracy was between 75% and 100%. It should be noted here that the entities with 100% accuracy are entities that contain less than six words.

In conclusion, we can say that the tool presented by this research is comparable in its accuracy to other tools that do the same work, but it surpasses them in its ability to annotate more entities. Thus, giving a more accurate description.

VIII. Conclusion

This research aimed to reach an automatic entity annotation tools for Arabic texts. This tool uses the DBpedia database as an information source; afterwards, work was done to improve its quality in terms of the number of types that a single resource could belong to. In addition, the research focused on verifying the validity of the attribution that was made for each resource; in other words, whether the type to which the resource belongs matches the concept that this resource represents. This is with the aim of obtaining structured data from unstructured data; thus, improving the quality of the output that can be obtained through applying NLP models. The contributions made by this research were:

First, devise a new ontology from the DBpedia ontology that represents the resources with Arabic label; besides, the new ontology aims to improve the quality of automatic entity annotation by unifying the resources that belong to similar concepts within the same type;

Second, build an algorithm to verify the validity of the types that were given to the resources;

Third, build an ATEA that provides the user with the final results in a format that enables him to use them in a different field.

This tool can be developed by increasing the size of its database in two ways, the first is to use the existing resources in DBpedia but have not an Arabic label after translating their English label to the Arabic language. The second is to use a new database and expanding the classes included in the DBpedia Arabic resources ontology. Thus, it will increase the efficiency of the output of this tool; then, the possibility of using their output as inputs for classification models and cluster models.

In future works, we will develop the tool to build semantic relationships to convert text into a RDF triple store. We will use different datasets after performing all data quality checks to achieve this goal. In addition, we aim to expand languages that can be automatically annotated; consequently, this will allow the use of different texts written in different languages as an input of different NLP models like similarity model without the need for translation by the use of the information of the words annotated.

REFERENCES

[1] Völkel M, Krötzsch M, Vrandecic D, et al. Semantic wikipedia. In: Proceedings of the 15th international conference on World Wide Web. 2006, pp. 585–594.

[2] Milne D, Witten IH. Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management. 2008, pp. 509–518.

[3] Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data. In: The semantic web. Springer, 2007, pp. 722–735.

[4] Hossain BA, Salam A, Schwitter R. A survey on automatically constructed universal knowledge bases. J Inf Sci 2021; 47: 551–574.

[5] Cucerzan S. Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 708–716.

[6] Mihalcea R, Csomai A. Wikify! Linking documents to encyclopedic knowledge. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. 2007, pp. 233–242.

[7] Kulkarni S, Singh A, Ramakrishnan G, et al. Collective annotation of Wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009, pp. 457–466.

[8] Ferragina P, Scaiella U. Fast and accurate annotation of short texts with wikipedia pages. IEEE Softw 2011; 29: 70–75.

[9] Makris C, Plegas Y, Theodoridis E. Improved text annotation with Wikipedia entities. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. 2013, pp. 288–295.

[10] Fellbaum C, Miller G. Wordnet: An electronic lexical database (language, speech, and communication). Epub ahead of print 2000. DOI: 10.2307/417141.

[11] Brank J, Leban G, Grobelnik M. Semantic annotation of documents based on wikipedia concepts. Informatica; 42, http://www.informatica.si/index.php/informatica/article/view/2228 (2018).

[12] Strobl M, Trabelsi A, Zaiane OR. WEXEA: Wikipedia EXhaustive Entity Annotation. In: Proceedings of The 12th Language Resources and Evaluation Conference, pp. 1951–1958.

[13] Benajiba Y, Diab M, Rosso P, et al. Arabic named entity recognition: An svm-based approach. In: Proceedings of 2008 Arab international conference on information technology (ACIT). 2008, pp. 16–18.

[14] Al-Jumaily H, Martínez P, Martínez-Fernández JL, et al. A real time Named Entity Recognition system for Arabic text mining. Lang Resour Eval 2012; 46: 543–563.

[15] Yosef MA, Spaniol M, Weikum G. {AIDA}rabic A Named-Entity Disambiguation Framework for {A}rabic Text. In: Proceedings of the {EMNLP} 2014 Workshop on {A}rabic Natural Language Processing ({ANLP}). Doha, Qatar: Association for Computational Linguistics, pp. 187–195.

[16] Al-Yahya M, Al-Shaman M, Al-Otaiby N, et al. Ontology-Based Semantic Annotation of Arabic Language Text. Int J Mod Educ Comput Sci 2015; 7: 53–59.

[17] Al-Qawasmeh O, Al-Smadi M, Fraihat N. Arabic named entity disambiguation using linked open data. In: 2016 7th International Conference on Information and Communication Systems (ICICS). 2016, pp. 333–338.

[18] Albukhitan S, Alnazer A, Helmy T. Semantic Annotation of Arabic Web Documents using Deep Learning. Procedia Comput Sci 2018; 130: 589–596.

[19] Mendes PN, Jakob M, Bizer C. DBpedia: A Multilingual Cross-domain Knowledge Base. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association, pp. 1813–1817.

[20] Bizer C, Lehmann J, Kobilarov G, et al. DBpedia-A crystallization point for the Web of Data. J web Semant 2009; 7: 154–165.

[21] Ismayilov A, Kontokostas D, Auer S, et al. Wikidata through the Eyes of DBpedia. Semant Web 2018; 9: 493–503.

[22] Lehmann J, Isele R, Jakob M, et al. DBpedia--a large-scale, multilingual knowledge base extracted from Wikipedia. Semant Web 2015; 6: 167–195.

[23] Ismail AS, Al-Feel H, Mokhtar HMO. Introducing a new arabic endpoint for DBpedia internationalization project. In: Proceedings of the 20th International Database Engineering & Applications Symposium. 2016, pp. 284–289.

[24] Zaveri A, Kontokostas D, Sherif MA, et al. User-driven quality evaluation of dbpedia. In: Proceedings of the 9th International Conference on Semantic Systems. 2013, pp. 97–104.

[25] Kontokostas D, Westphal P, Auer S, et al. Test-driven evaluation of linked data quality. In: Proceedings of the 23rd international conference on World Wide Web. 2014, pp. 747–758.

[26] Radulovic F, Mihindukulasooriya N, Garc\'\ia-Castro R, et al. A comprehensive quality model for Linked Data. Semant Web 2018; 9: 3–24.

[27] Rico M, Mihindukulasooriya N, Kontokostas D, et al. Predicting incorrect mappings: a data-driven approach applied to DBpedia. In: Proceedings of the 33rd annual ACM symposium on applied computing. 2018, pp. 323–330.

[28] Lakshen GA, Janev V, Vraneš S. Challenges in quality assessment of Arabic DBpedia. In: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics. 2018, pp. 1–4.

[29] Jarrar M, Khalilia M, Ghanem S. Wojood: Nested {A}rabic Named Entity Corpus and Recognition using {BERT}. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. Marseille, France: European Language Resources Association, pp. 3626–3636.

[30] Al-Thubaity A, Alkhereyf S, Alzahrani W, et al. {CA}ra{NER}: The {COVID}-19 {A}rabic Named Entity Corpus. In: Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP). Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 1–10.

[31] Public SPARQL Endpoint. DBpedia, https://wiki.dbpedia.org/public-sparql-endpoint (2020, accessed 22 July 2020).

[32] Zhang Z, Krawczyk B, Garcìa S, et al. Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. Knowledge-Based Syst 2016; 106: 251–263.

[33] Kahlawi A. An Ontology Driven ESCO LOD Quality Enhancement. Int J Adv Comput Sci Appl; 11. Epub ahead of print 2020. DOI: 10.14569/IJACSA.2020.0110308.

[34] Zhu L, Ghasemi-Gol M, Szekely P, et al. Unsupervised Entity Resolution on Multi-type Graphs. In: Groth P, Simperl E, Gray A, et al. (eds) The Semantic Web -- ISWC 2016. Cham: Springer International Publishing, 2016, pp. 649–667.

[35] Rahman NA, Mabni Z, Omar N, et al. A Parallel Latent Semantic Indexing (LSI) Algorithm for Malay Hadith Translated Document Retrieval. In: Berry MW, Mohamed A, Yap BW (eds) Soft Computing in Data Science. Singapore: Springer Singapore, 2015, pp. 154–163.

[36] Kahlawi A, Martelli C, Buzzigoli L, et al. A similarity matrix approach to empower ESCO interfaces for testing , debugging and in support of users ' experience. In: Pollice A, Salvati N, Spagnolo FS (eds) Riunione Scientifica della Società Italiana di Statistica -SIS. Pisa: Pearson, pp. 904–909.

[37] Benajiba Y, Rosso P, BenedíRuiz JM. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In: Gelbukh A (ed) Computational Linguistics and Intelligent Text Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 143–153.

[38] Obeid O, Zalmout N, Khalifa S, et al. {CAM}e{L} Tools: An Open Source Python Toolkit for {A}rabic Natural Language Processing. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. Marseille, France: European Language Resources Association, pp. 7022–7032.

Illicit Activity Detection in Bitcoin Transactions using Timeseries Analysis

Rohan Maheshwari[1], Sriram Praveen V A[2], Shobha G[3], Jyoti Shetty[4], Arjuna Chala[5], Hugo Watanuki[6]
Computer Science and Engineering Department, R. V. College of Engineering, Bengaluru, India[1, 2, 3, 4]
HPCC Systems LexisNexis Risk Solutions, Alpharetta, USA[5, 6]

***Abstract*—A key motivator for the usage of cryptocurrency such as bitcoin in illicit activity is the degree of anonymity provided by the alphanumeric addresses used in transactions. This however does not mean that anonymity is built into the system as the transactions being made are still subject to the human element. Additionally, there is around 400 Gigabytes of raw data available in the bitcoin blockchain, making it a big data problem. HPCC Systems is used in this research, which is a data intensive, open source, big data platform. This paper attempts to use timing data produced by taking the time intervals between consecutive transactions performed by an address and make an identification of the nature of the address (illegal or legal). With the use of three different goodness of fit run tests namely Kolmogorov–Smirnov test, Anderson-Darling test and Cramér–von Mises criterion, two addresses are compared to find if they are from the same source. The BABD-13 dataset was used as a source of illegal addresses, which provided both references and test data points. The research shows that time-series data can be used to represent transactional behaviour of a user and the algorithm proposed is able to identify different addresses originating from the same user or users engaging in similar activity.**

***Keywords*—Bitcoin; time-series analysis; HPCC systems; random time interval; illicit activity detection**

I. Introduction

The rapid growth of Cryptocurrencies has proved to be a major regulatory challenge. They are used in various illegal activities including illegal trade of drugs, hacks, thefts, illicit pornography and other major crimes. According to the 2022 Crypto Crime report by Chainalysis [1], a total of $14 billion was involved in Cryptocurrency based illicit activities in the year 2021. Although it represents only 0.15% of the total volume of crypto transactions, digital currencies and trading of these assets is becoming increasingly mainstream. The tracked volume of illicit activity is likely to rise in the future as more bad actors are identified. Bitcoin is the first and the most established cryptocurrencies in the world. Bitcoin provides pseudo-anonymity in the form of a 26-35 length alphanumeric addresses. This makes the identification of users difficult. However, it is possible to link transactions to users due to the public nature of the Bitcoin blockchain. After linking the addresses to users, it is further possible to identify which of these users were involved in activities that are of criminal nature [2].

Consider a user who has a dedicated set of Bitcoin addresses which they use to perform some kind of illegal activity. Over time, as they perform their transactions, the timestamps are available publicly and can be used to reveal the identity of the user. Behavioral biometrics rely usually on a rich stream of information to identify a user. RTI is one such biometric which can be considered as any sequence of time intervals between successive events. As we have the timestamps from the bitcoin blockchain we can obtain the RTI of transactions by taking the difference between the timestamp of the current transaction and the previous transaction for the sequence of transactions made by the user. This work uses the random time-interval (RTI) biometric [3].

This paper presents a method to recognise individual users from their behavioral metrics which is the time-series data and is publicly available on the bitcoin blockchain. A review of the related work is presented on the next section. In Section III, we present the methodology used which encompasses the methods and components involved in the research. Finally, the results and discussion of the three goodness-of-fit tests are presented in Section IV followed by conclusion in Section V.

II. Related Work

The most important datapoint that is generated by a bitcoin address for this technique of fraud detection is the RTI or Random Time Interval data which was first made use of in the paper by Laskaris, Zafeiriou, and Garefa [4]. Subjects were given a button to press randomly and it was shown that the time series produced could be used as a biometric signature of a person's cognitive thought process. Monaco [3] made use of 12 total time signatures unique to any address and used Takens's theorem for phase space reconstruction followed by the approximate multivariate wald-wolfowitz test to check whether the two samples originated from the same sources. The best identification rate was 76.

Other types of biometric identification have been attempted such as signature verification using template matching [5]. In this a novel variation of the DTW algorithm is used which produced verification errors of as low as 1.34% at a very rapid speed. Another application of timing data for biometric identification is in gait analysis. Mekruksavanich and Jitpattanakul [6] were able to use the timing data of 22 subjects from portable devices and create a CNN model which achieved an identification accuracy of as high as 93.9%.

Dynamic Time Warping (DTW) [7] is used to align two different time series data to recognize the origin of data. The only fundamental difference between various time series analysis and bitcoin time series analysis is that the measurement is event driven versus being measured continuously by sensors. Often transactions also occur over various weeks or even months whereas with a specific application such as gait analysis data can be continuously captured over a period of day or two. Another difference is the accuracy of the time stamps being used, with sensor data the data is quite precise, but since transaction level timestamps are not available in the blockchain, block level time stamps are used which reduces accuracy.

Many techniques have been developed for bitcoin fraud detection apart from the usage of timestamp data such as clustering [8], various techniques such as trimmed k-means, DBSCAN etc. are tested to this end. Various network analysis techniques have also been analyzed [9] such as use of LOF (local outlier factor). The difference between these and time series analysis is that network analysis attempts to find anomalies in behavior globally whereas time series attempts to show that two addresses originate from the same user with no mention of the nature of the transactions being made.

III. Methodology

Bitcoin is the oldest and most widely adopted cryptocurrency to date. It becomes the natural choice for study given the amount of transactional data available publicly. Having 389 Gigabytes of raw data to work with immediately makes this a big data problem as bare metal resources cannot handle this much data. Hence, HPCC Systems is used. Another reason to choose HPCC is its ROXIE delivery engine which let us query new addresses quickly.

A. Bitcoin

On October 28th, 2008 Satoshi Nakamoto made public his research on a trustless peer to peer electronic cash system called bitcoin [10]. The system has its origins in the growing distrust around centralized financial institutions [11]. It relies on the proof of work model where the longest chain is the one that is trusted first amongst all the miners. Bitcoin also makes its transaction history easily accessible to the public making it easy to perform analysis.

The two most relevant data points on the chain to this particular research effort are the bitcoin addresses and bitcoin transactions. Every bitcoin user has access to a private key and a public key. The bitcoin address is derived from the public key by the use of one-way cryptographic hashing [12]. It represents the origin and destination of bitcoin in transactions. RTI data will be aggregated

Bitcoin transactions represent a transfer of bitcoin from one address to another. The transactions are then added to a mempool and miners race to add it to their blocks and obtain the block reward. The transactions are contained in blocks which are chained together.

B. HPCC Platform

As of April 2022, the amount of raw blk data a bitcoin client would download is 389 Gigabytes and if the solution is expanded to other crypto such as ethereum it can take as much as 658 Gigabytes of data. This makes the problem of fraud detection a big data problem where regular computational resources will struggle to keep up with the demands of scaling data. HPCC (High-Performance Computing Cluster) is a big data platform developed by LexisNexis Risk Solutions. It supports both parallelized batch computing and online querying using a declarative and data centric programming language called ECL (Enterprise Control Language) [13].

The HPCC platform has three main components as shown in Fig. 1, namely THOR, Roxie and ECL. The THOR cluster is used for batch processing, while ROXIE is used to run multiple online queries. ECL is the language used to interact with these two clusters. There is a plethora of other components used for housekeeping and maintenance of logical files and work units such as Dali, Sasha etc. The clusters themselves can be made up of commodity hardware to supercomputers as nodes. Thus, HPCC provides for scalability.

C. Goodness-of-Fit Tests

These statistical tests check whether a given set of observations were drawn from the same distribution. Generally, these tests check whether the observations were drawn from the normal distribution. For our purposes, a two-sample variation of the tests is used, to compare the underlying continuous distributions of two sets of independent observations.

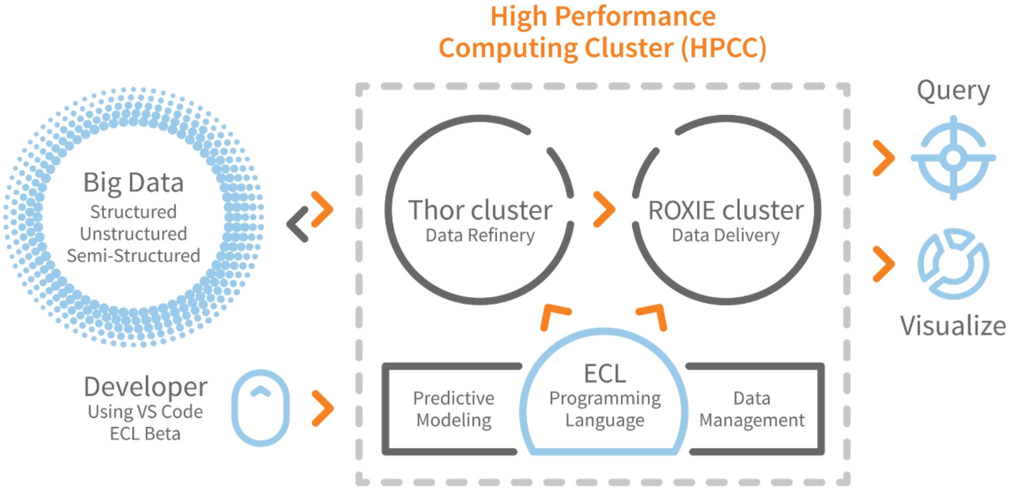


Fig. 1. HPCC systems architecture.

The three tests chosen show a varying degree of bias in different situations and have varying statistical power measures. Various studies have been conducted which include a series of goodness of fit tests and compare its statistical powers on different distributions. The Kolmogorov-Smirnov test has more statistical power against the distributions with n<100. The Anderson Darling test shows a better power against the distributions with higher sample sizes. The Cramer-von-Mises test performs similar to the KS test but for certain distributions with n>50, this test outperforms the Kolmogorov-Smirnov test [14].

1) Kolmogorov-smirnov test: A variation of the one sample test is used which compares a set of two observations [15]. It creates a cumulative distribution for both of the sets of observations after sorting them. The difference between the two distributions is obtained and the maximum of these differences is compared against the critical value. Should the Kolmogorov test statistic be less than critical value, the null hypothesis that both the observations originate from the same distribution will be rejected. In this paper the scipy implementation of this test is used [16]. We use the 'two-sided' option for the alternative parameter which states that the alternative hypothesis is that two distributions are not identical, F(x) is not equal to G(x) for all x; and the statistic is given by the maximum absolute difference between the empirical distribution functions of the samples. The method parameter is set to 'auto' option which means that for small arrays, it uses the exact distribution of test statistic; and for large arrays, it uses asymptotic distribution of test statistic.

2) Anderson-darling test: The Anderson Darling test is another test to check for data coming from a particular distribution. K-sample Anderson-Darling test is a modification that tests the null hypothesis that k-samples are drawn from the same population without any specification of the distribution function of that population [17]. The critical values depend on the number of samples. In this paper the scipy implementation of the k-sample Anderson-Darling test is used, taking the value of k as 2 [18]. The *midrank* parameter is set to *True* which computes the test using the midrank empirical distribution function applicable to continuous and discrete data.

3) Cramér–von mises criterion: Cramér–von Mises is another goodness of fit test. The two-sample Cramér-von Mises test is a test where the null hypothesis is that the samples are from the same, unspecified continuous distribution [19]. The scipy implementation of the Cramér-von Mises test is used in this paper [20]. The only parameter is *method* which is set to '*auto'* option working similar to the parameter in the Kolmogorov-Smirnov test.

D. Dataset

The Bitcoin Address Behavior Dataset [21] contains 13 categories of bitcoin addresses, each containing a list of addresses from a different type of crime. The reference illegal addresses have been picked from these. Specifically, to make the task of obtaining RTI simpler, 1500 addresses were randomly picked from labels 0, 10 and 11 and their RTI data was obtained by using a simple python script using the JSON-RPC interface to fetch all the information about transactions made by that address. The BABD-13 is used as it is a robust dataset which addresses not only the crime that the address was involved in, but also provides the degree of certainty that the illegal activity has occurred. Not only does this allow for the selection of illegal activities of interest, but it also enables the selection of only those addresses that are labeled as illegal with a high degree of certainty.

The other dataset in use is the one generated by the parser described later on, it has been obtained by parsing the binary data to CSV. There are many more data points that can be extracted from the raw blk data such as Merkle root, block hash etc. but have been ignored as the crucial data point for the research is the timing data.

E. Methods

The main idea was to retrieve all the bitcoin raw data using bitcoin core followed by the usage of a parser to retain the data in CSV format. This is followed by the generation of Random Time Interval data. This data is also generated for the addresses extracted from the BABD-13 dataset which will be part of both reference and test dataset. Each sample in the test dataset is then run using all three goodness of fit tests against each of the reference addresses. The statistic values are then averaged and if the average is greater than 0.5, the test returns a positive hit for suspicious activity from the tested address. The workflow is shown in Fig. 2.

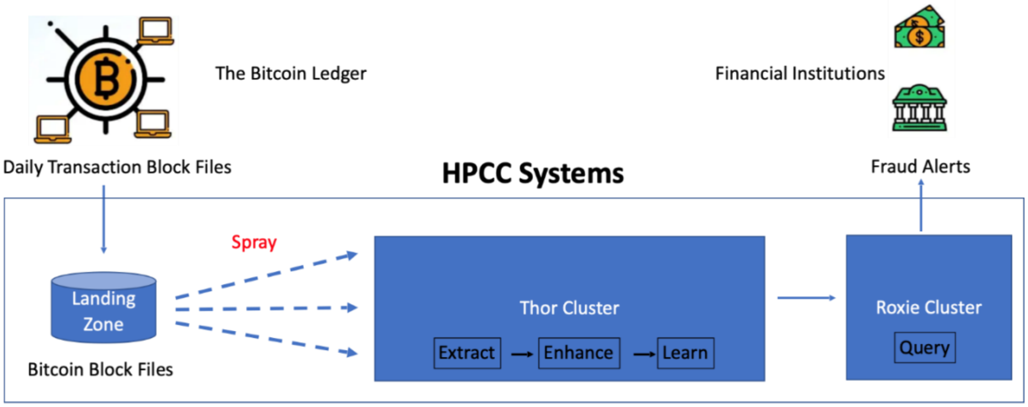


Fig. 2. Algorithm methodology.

The bitcoin data is obtained by initializing a bitcoin core node which downloads the data in the form of blk.dat files which are then sprayed onto the thor cluster for processing after passing through the landing zone. Here it goes through various ETL operations after which financial institutions will be able to submit queries of particular addresses to check for fraud on Roxie.

1) Parser: A bitcoin core node was set up to fetch the bitcoin blockchain raw data. The blockchain has been broken down into multiple blk.dat files (3185 as of writing this paper). The blk files consist of multiple blocks and are limited to 128 MiB. This data is all in binary form and a parser must be used to convert to CSV, JSON or some other structured format.

A modified Bitcoin parser [22] written in python is implemented using the HPCC Systems platform. The parser is embedded into ECL to take advantage of the parallel architecture available on the platform. This embedded python parser showed a significant improvement over a single node parser. For a set of 50 random blk.dat files, the HPCC parser ran at 5% of the time of a single node parser for 41 minutes.

This first phase of the parser extracted the following data points

a) Transaction Hash

b) Input Index

c) Input Transaction Hash

d) Output Index

e) Output Address

f) Output Value

g) Timestamp

The only missing data here is the input address, this data is obtained by using ECL. The blockchain stores the input transaction hash and the corresponding output index to refer to the inputs of a transaction. Thus, to find the actual input address ECL's self-join operation is used. Here the transaction in question and the previous transaction are joined, the corresponding output address of the previous transaction is the input address of the current transaction. This is shown in Fig. 3. This process leads to all the data being collected as required by the various algorithms used in this paper.

2) RTI Generation: Random Time Interval (RTI) data [4] can be generated from the timestamp data obtained from the blockchain. The data is generally precise and captured by sensors in cases such as gait analysis [7], this is where bitcoin presents a slight hurdle. Transaction level timestamps are not recorded and the closest substitute is the block level timestamp. Even with this rougher granularity however, some promising results do arise as will be seen later on.

The RTI data is calculated by taking consecutive timestamps in UNIX time of transactions made by a given address and subtracting them from their successor. Thus, if an address has made n transactions it will have an RTI of length n-1. RTI generation was done by making use of ECL's ITERATE function.

3) Algorithm: A set of reference addresses are taken which can contain legal and illegal addresses. A test query is then taken and one of the three aforementioned run tests is used to evaluate the test address against each and every reference address. If the reference address is illegal and returns a p value greater than the threshold or if the reference address is legal and the test returns a p value less than the threshold it is added to the total number of hits. Finally, if the fraction of hits is greater than a threshold value, the query will return as illegal.

This threshold on the fraction can be thought of as a level of risk tolerance, where a lower threshold means that financial institutions would want to investigate addresses even with a small fraction of hits and vice versa.

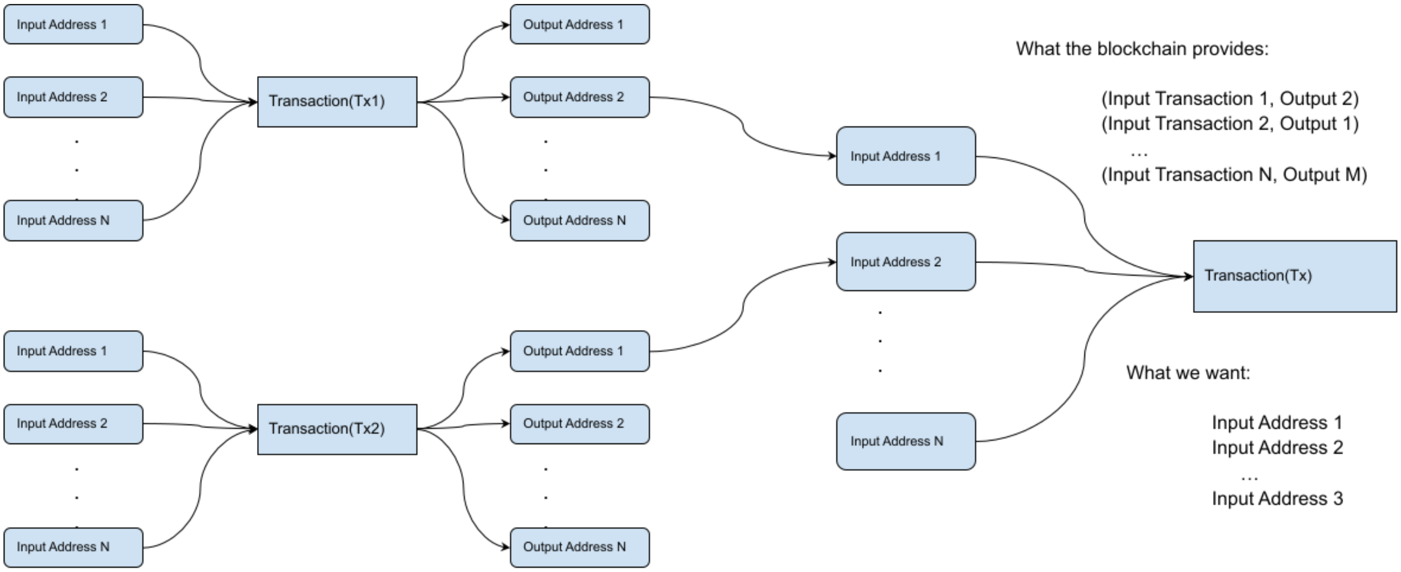


Fig. 3. Input to output address relation.

These illegal reference addresses are taken from the Bitcoin Address Behavior Dataset (BABD) [21] and the legal addresses are addresses which have been randomly taken with the assumption that 1% of bitcoin addresses are illegal. There is very high variability in the consideration of percentage of illegal activity in bitcoin [2][23]. The implementation of this querying will eventually be done on ROXIE for financial institutions to use.

IV. Results and Discussion

Before there are three run tests that have been used to evaluate distributional similarities between the RTI data of two addresses. They are:

- Kolmogorov–Smirnov test
- Anderson–Darling test
- Cramér–von Mises criterion

Based on the composition of the reference and test query dataset, three evaluations were made,

- Illegal only test vs illegal only reference
- Illegal only test vs illegal and legal reference
- Illegal and legal test vs illegal only reference

The results obtained were evaluated based on precision, accuracy, f1-score and recall and only those addresses with at least 25 transactions were considered in both reference and query sets.

A. *Method 1*

In this method the test dataset consisted of only known illegal addresses and the references also consisted of only known illegal addresses. The size of the reference set was 263 and the size of the query set was 66. Table I summarizes these findings.

TABLE I. Illegal Only Test Vs Illegal Only Reference

Algorithms	Precision	Recall	f1-score	Accuracy
Kolmogorov–Smirnov test	1.00	0.85	0.92	0.85
Anderson–Darling test	1.00	0.77	0.87	0.77
Cramér–von Mises criterion	1.00	0.80	0.89	0.80

B. *Method 2*

In this method the test dataset consisted of only known illegal addresses but the references consisted of both known illegal addresses and random addresses which were considered legal. The size of the reference set was 304 and the size of the query set was 66. Table II summarizes these findings.

TABLE II. Illegal Only Test Vs Illegal and Legal References

Algorithms	Precision	Recall	f1-score	Accuracy
Kolmogorov–Smirnov test	1.00	0.85	0.92	0.85
Anderson–Darling test	1.00	0.79	0.88	0.79
Cramér–von Mises criterion	1.00	0.80	0.89	0.80

C. *Method 3*

In this method the test dataset consisted of known illegal addresses and legal addressee and the references consisted of only known illegal addresses. The size of the reference set was 263 and the size of the query set was 107. Table III summarizes these findings.

The following observations can be made from the above data. The Kolmogorov–Smirnov test consistently across all three methods either outperforms or is at par with the other two run tests in terms of accuracy. However, the f1-score is lower in general.

Given the f1-score is lower a further look into the recall and precision of the methods is called for. In the use case of detecting illegal addresses for financial institutions with virtually unlimited resources, it is possible to look into false positives and so a higher recall would be preferable. Here as well, the Kolmogorov–Smirnov test gives the highest recall except in the detection of legal addresses in method 3 where the Anderson –Darling test does better with 33%.

From method 3 we see a sharp contrast in detection between legal and illegal addresses. The f1 is consistently over double for all three run tests. This is due to the initial assumption that only 1% of addresses being involved in illegal activity is wrong and random selection of addresses has led to addresses that are not legal to be included making the query and test dataset impure. It is also because the activity that legal addresses are involved in and thereby the time series patterns generated vary too widely for a small sampling to represent them. This can cause legal addresses to be misclassified as illegal, leading to lower overall detection rates for legal addresses.

TABLE III. Illegal and Legal Test, Illegal Only Reference

Algorithms	Kolmogorov–Smirnov test					Anderson–Darling test					Cramér–von Mises criterion				
	class		*accuracy*	*macro avg*	*weighted avg*	*class*		*accuracy*	*macro avg*	*weighted avg*	*class*		*accuracy*	*macro avg*	*weighted avg*
	0	*1*				*0*	*1*				*0*	*1*			
Precision	0.47	0.64		0.56	0.57	0.46	0.65		0.55	0.58	0.41	0.62		0.52	0.54
Recall	0.22	0.85		0.53	0.61	0.32	0.77		0.54	0.6	0.22	0.8		0.51	0.58
f1-score	0.3	0.73	0.61	0.51	0.56	0.38	0.7	0.62	0.54	0.58	0.29	0.7	0.58	0.49	0.54
Support	41	66	107	107	107	41	66	107	107	107	41	66	107	107	107

The number of addresses with more than 25 transactions is a smaller fraction of the number of data points collected. This also points to other transactional behavior where bitcoin addresses are not reused as much or that bitcoin is often unused and parked in addresses. The limited number of addresses with more than 25 transactions suggests that the behavior of bitcoin addresses may vary widely and may not be accurately represented in the dataset.

The lack of data on illegal transactions and addresses limits the certainty of identifying illegal addresses. Therefore, the accuracy of the results may be affected, and false positives may occur.

V. Conclusion

The work supports the original hypothesis that bitcoin transaction behavior is nonrandom. The paper presents strong evidence that users engaging in illegal activity can be detected by use of previously known illegal addresses.

The simple use of timing data has proven to be quite effective. Paired with external information about the owner of an address or even network information available on the blockchain such as the amount being sent, the effectiveness could go higher.

One problem with any bitcoin illegal detection approach is the lack of data on illegal transactions and addresses. The dataset used in this paper does provide illegal addresses but only with a limited degree of certainty. Legal addresses are also seen to have a lower overall detection rate. This is likely due to the assumption of a percent of addresses being illegal being off while picking random addresses. While the Kolmogorov-Smirnov test may outperform the other run tests in terms of accuracy, the limitations of the dataset and assumptions made in the paper must be considered when interpreting the results. Further research and more comprehensive datasets are needed to improve the accuracy and reliability of illegal address detection in bitcoin transactions.

The method is only dependent on one feature which is the time interval data being generated, and since this is not a feature specific to bitcoin it can very easily be extended to other cryptocurrencies such as Ethereum, Luna etc. Similarly, it can be extended to other applications such as credit card fraud detection which have the added advantage of having precise transaction timestamp information and easy access to a person's history.

References

[1] Grauer, K., Updegrave, H. and Kueshner, W. (2022) The chainalysis 2022 crypto crime report, Chainalysis. Retrieved from https://go.chainalysis.com/2022-Crypto-Crime-Report.html (Accessed: December 2, 2022).

[2] Foley, S., Karlsen, J. R., & Putnin¸s, T. J. (2019). Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? The Review of Financial Studies, 32(5), 1798-1853.

[3] Monaco, J. V. (2015, May). Identifying bitcoin users by transaction behavior. In Biometric and surveillance technology for human and activity identification XII (Vol. 9457, pp. 25-39). SPIE.

[4] Laskaris, N. A., Zafeiriou, S. P., & Garefa, L. (2009). Use of random time-intervals (RTIs) generation for biometric verification. Pattern Recognition, 42(11), 2787-2796.

[5] Okawa, M. (2019). Template matching using time-series averaging and DTW with dependent warping for online signature verification. IEEE Access, 7, 81010-81019.

[6] Mekruksavanich, S., & Jitpattanakul, A. (2021). Convolutional neural network and data augmentation for behavioral-based biometric user identification. In ICT Systems and Sustainability: Proceedings of ICT4SD 2020, Volume 1 (pp. 753-761). Springer Singapore.

[7] Permanasari, Y., Harahap, E. H., & Ali, E. P. (2019, November). Speech recognition using dynamic time warping (DTW). In Journal of physics: Conference series (Vol. 1366, No. 1, p. 012091). IOP Publishing.

[8] Shayegan, M. J., Sabor, H. R., Uddin, M., & Chen, C. L. (2022). A Collective Anomaly Detection Technique to Detect Crypto Wallet Frauds on Bitcoin Network. Symmetry, 14(2), 328.

[9] Turner, A. B., McCombie, S., & Uhlmann, A. J. (2020). Analysis techniques for illicit bitcoin transactions. Frontiers in Computer Science, 2, 600596.

[10] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260.

[11] De Filippi, P., Mannan, M., & Reijers, W. (2020). Blockchain as a confidence machine: The problem of trust & challenges of governance. Technology in Society, 62, 101284.

[12] Antonopoulos, A.M. (2014) Mastering bitcoin, O'Reilly Online Learning. O'Reilly Media, Inc. Retrieved from https://www.oreilly.com/ library/view/mastering-bitcoin/9781491902639/ch04.html (Accessed: December 2, 2022).

[13] Karthik, A., Mishra, H., Jayanth, S., Shobha, G., & Shetty, J. (2022, January). Performance skew prediction in HPCC systems. In 2022 12th International Conference on Cloud Computing, Data Science Engineering (Confluence) (pp. 94-97). IEEE.

[14] Fusek, M. (2023). Statistical Power of Goodness-of-Fit Tests for Type~ I Left-Censored Data. Austrian Journal of Statistics, 52(1), 51-61.

[15] Hodges, J. L. (1958). The significance probability of the Smirnov twosample test. Arkiv for Matematik, 3(5), 469-486.¨

[16] scipy.stats.ks_2samp (2022). SciPy v1.9.3 Manual. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks\ _2samp.html (Accessed: December 2, 2022).

[17] Scholz, F. W., & Stephens, M. A. (1987). K-sample Anderson–Darling tests. Journal of the American Statistical Association, 82(399), 918-924.

[18] scipy.stats.anderson_ksamp (2022). SciPy v1.9.3 Manual. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats. anderson\ ksamp.html (Accessed: December 2, 2022).

[19] Anderson, T. W. (1962). On the distribution of the two-sample Cramervon Mises criterion. The Annals of Mathematical Statistics, 1148-1159.

[20] scipy.stats.cramervonmises_2samp (2022) SciPy v1.9.3 Manual. Retrieved from https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.cramervonmises\ 2samp.html (Accessed: December 2, 2022).

[21] Xiang, Y., Ren, W., Gao, H., Bao, D., Lei, Y., Li, T., Yang, Q., Liu, W., Zhu, T., & Choo, K. K. R. (2022). BABD: A Bitcoin Address Behavior Dataset for Address Behavior Pattern Analysis. arXiv preprint arXiv:2204.05746.

[22] Calvez, A.L. (2022) Alecalve/python-bitcoin-blockchain-parser: A python 3 bitcoin blockchain parser, GitHub. alecalve. Retrieved from https://github.com/alecalve/python-bitcoin-blockchain-parser (Accessed: December 3, 2022).

[23] Maheshwari, R. (2022) Breaking down Bitcoin blockchain using HPCC systems, HPCC Systems. HPCC Systems. Retrieved from https:// hpccsystems.com/blog/RVCE-RohanM-Blockchain (Accessed: December 3, 2022).

Strategic Monitoring for Efficient Detection of Simultaneous APT Attacks with Limited Resources

Fan Shen, Zhiyuan Liu, Levi Perigo
Department of Computer Science, University of Colorado Boulder
Boulder, Colorado 80309

***Abstract*—Advanced Persistent Threats (APT) are a type of sophisticated multistage cyber attack, and the defense against APT is challenging. Existing studies apply signature-based or behavior-based methods to analyze monitoring data to detect APT, but little research has been dedicated to the important problem of addressing APT detection with limited resources. In order to maintain the primary functionality of a system, the resources allocated for security purposes, for example logging and examining the behavior of a system, are usually constrained. Therefore, when facing multiple simultaneous powerful cyber attacks like APT, the allocation of limited security resources becomes critical. The research in this paper focuses on the threat model where multiple simultaneous APT attacks exist in the defender's system, but the defender does not have sufficient monitoring resources to check every running process. To capture the footprint of multistage activities including APT attacks and benign activities, this work leverages the provenance graph which is constructed based on dependencies of processes. Furthermore, this work studies the monitoring strategy to efficiently detect APT attacks from incomplete information of paths on the provenance graph, by considering both the "exploitation" effect and the "exploration" effect. The contributions of this work are two-fold. First, it extends the classic UCB algorithm in the domain of the multi-armed bandit problem to solve cyber security problems, and proposes to use the malevolence value of a path, which is generated by a novel LSTM neural network as the exploitation term. Second, the consideration of "exploration" is innovative in the detection of APT attacks with limited monitoring resources. The experimental results show that the use of the LSTM neural network is beneficial to enforce the exploitation effect as it satisfies the same property as the exploitation term in the classic UCB algorithm and that by using the proposed monitoring strategy, multiple simultaneous APT attacks are detected more efficiently than using the random strategy and the greedy strategy, regarding the time needed to detect same number of APT attacks.**

Keywords*—*Advanced persistent threats; intrusion detection; LSTM; multi-armed bandit

I. Introduction

Multiple malicious activities can happen simultaneously on a host or system, especially when it performs mission-critical tasks [1]. If the monitoring capacity (also referred to as monitoring resources throughout this paper) is limited, without thoughtful allocation of these resources, it is possible that some malicious activities will not be captured and identified. A generic way of allocating limited resources is to assign the monitoring resources to the most important or suspicious objects, which is also called a greedy strategy. One dilemma of the greedy strategy, however, is that, if the early perceptions about objects are not reliable, benign objects might consume more monitoring resources than malicious objects. Unfortunately, unreliable early detection of Advanced Persistent Threats is common, because such advanced attacks are stealthy and it is possible that an APT attack in its earlier stages is less suspicious than a benign activity. When using a greedy strategy to allocate resources, the Matthew Effect can cause some of the attacks to be undetected, because less monitoring resources are assigned to them. Therefore, an enhanced method to allocate monitoring resources, compared with the greedy strategy, is needed for the detection of simultaneous long-term attacks with limited resources, because this threat model has not been extensively addressed by existing works in the area of anomaly detection.

The research in this paper focuses on the detection of a sophisticated cyber attack, APT[2], and proposes a strategy to allocate limited security resources for monitoring, in order to efficiently detect APT activities when multiple APT activities are ongoing concurrently in a system.

II. Related Work

APT has attracted the attention of industry and academia since the 2010s, when several unseen yet powerful APT malware were discovered including Stuxnet, Duqu, Flame and Gauss. Although it is a multistage, complicated attack, the typical stages and behavioral patterns of APT are extracted by existing studies. For example, attack chain models based on Cyber Kill Chain [3] and the attack pyramid model [4] are proposed to characterize multiple APT stages and their relations. MITRE ATT&CK [5] constantly publish common tactics, techniques, and procedures (TTP) of APT attacks. Then according to the characteristics of APT, researchers use different methods to effectively detect it from illegitimate system access, suspicious network traffic patterns, and abnormal system resource utilization. Based on the categories of methods, APT detection studies are mainly divided into two categories: signature-based and behavior-based.

Signature-based detection methods match system behaviors with known attack patterns, once a match is found, pre-configured actions will be taken. Snort [6] is an example of signature-based intrusion detection system, which generates explicit rules from known attacks, if any rule is matched, it will trigger actions such as SNMP traps, event logging, and allow/deny traffic. The author [7] leverages the Intrusion Kill Chain [8] model defining rules to identify each APT stage by its attack mechanisms from multiple sources of logs and build the kill chain by comparing the timestamps of found APT stages. HOLMES [9] defines explicit rules to map a low-level system event to an abstracted APT TTP of MITRE ATT&CK and use the rules to identify each APT stage. As Command and Control (C2) communication is common in an APT campaign, network packet inspection or network flow analysis is used in some works to identify the C2 communication. The authors of [10] found consistent patterns in the network traffic between the APT malware of interest and the C2 server, and used protocol-aware rules, special strings in a URL, and malformed images to effectively detect C2 communication. Signature-based detection methods are effective to attacks whose characteristics are known and can be well-represented by a set of matching rules, and when compared with behavior-based methods, signature-based methods are less complicated and easier to implement. However, the performance of signature-based methods can degrade quickly when dealing with variants of known attacks and novel or zero-day attacks.

Behavior-based detection methods do not rely on known patterns, and they profile behaviors, either benign or malicious, using statistical or machine learning methods [11]. The advantage of such detection methods is being able to accommodate variations and uncertainties of behaviors and characterize behaviors that can not be represented by explicit matching rules. To identify suspicious hosts that might be involved in C&C communications, the authors in [12] characterized each host periodically using features extracted from network flows, then assigned a risk score to each host based on the deviation from its historical positions, distances from other hosts, and magnitude of increment in the feature space. The authors [13][14] focus on detection of spear phishing emails that are used for initial penetration in APT. The authors of [13] extract static and dynamic features of PDF files attached in emails, and propose a classification model using Support Vector Machine, to detect malicious PDF files. Active learning is also integrated in their model to cope with unseen PDF files. The researchers in [14] used Naïve Bayes theorem to detect spam emails containing links that redirect the victim to malicious websites, which could help APT attacker establish backdoors inside the victim's system. The study [15] proposes an ensemble RNN-based model to detect different APT steps by analyzing network traffic data. Host-level system logs are analyzed in [16] to identify different APT phases. The authors of [16] first translated system log sequences into abstracted states by using the hidden Markov model, then fed high-level state sequences to three multi-classification models: LSTM, one-dimensional CNN, and SVM, to predict the APT phase. Since the detection of a single APT stage does not necessarily indicate a multistage APT activity, some works [4][17][18][19] detect APT by correlating detection results of different APT stages based on models that describe dependencies between stages.

In existing studies on APT detection, the resource allocation problem is rarely addressed; whereas, it is a popular topic in game-theoretic studies [20][21] on APT prevention. Resource constraints, however, do exist in APT detection, either only a portion of CPU and memory can be used for monitoring activities in a system, or only limited insights can be obtained from enormous monitoring data in a timely manner. In a worse-case scenario, if multiple attacks concurrently exist in a system, and monitoring resources are not enough to cover all attacks, it is critical that the system operator allocate limited resources efficiently so that the most attacks possible are detected. Therefore, this work focuses on multistage APT attacks, and proposes a novel strategy that allocates monitoring resources, not only based on the current malevolence of activities, but also introducing an exploration mechanism to eliminate the side effect of a greedy strategy when the malevolence calculation based on early stage information of APT is not reliable.

III. Threat Model

In this work, simultaneous and continuous APT attack activities exist in the defender's system. For each multistage APT attack, each of its stages can be detected if relevant behavior is monitored. It is assumed that a whole-system provenance graph is used to obtain all paths including non-APT paths and APT paths. However, to reflect a real-world scenario, the monitoring capability of the defender's system is limited, so that only the activity on some, but not all, paths can be monitored and analyzed at each timestamp. Every monitoring timestamp provides a classification result indicating that the monitored activity is benign or one of APT stages. Therefore, for each path on the provenance graph, the defender has a sequence of temporal but incomplete detection results about the activities on the path. Thus, the goal of the defender is to efficiently detect as many simultaneous APT attacks as possible, while utilizing limited resources.

IV. Problem Setting

To capture the footprint of system activities, this work leverages the provenance graph which is constructed based on the dependencies of processes. It is assumed that the provenance graph is complete and not compromised in this paper. Therefore, the movement of a multistage APT attack is represented by one of the paths on the provenance graph, and in the threat model of this paper, the defender needs to detect those attack paths efficiently with limited resources. In addition, to decide the identity of a path, benign or APT, monitoring and investigation is used to check the state of nodes on a path. At each timestamp, however, not all running nodes can be checked because the monitoring resources are limited. Therefore, the research question addressed in this paper is, how to allocate limited monitoring resources, (i.e. select which running nodes should be evaluated at each timestamp), so APT attacks are detected effectively in terms of the number of detected attacks, and efficiently in terms of time needed? And a strategic monitoring model which considers both the estimated malevolence of system activities and the uncertainty of that estimation is proposed to solve the problem.

The problem setting of this paper is described as follows. There are n processes running on a host, which correspond to n paths on the provenance graph, but at each timestamp, only k ($k < n$) processes can be monitored and investigated to get the current state of the corresponding path. In this paper, when a process is chosen to be monitored, its local structure is fed into the detection engine developed by [11] which outputs a classification result. Mathematically, for each path i ($i \in [1, n]$), at time j, if the process corresponding to path i is monitored, the current state of path s_{ij} is obtained from the detection engine and $s_{ij} = \{0, 1, 2, 3, 4, 5, 6\}$; if the process corresponding to path i is not monitored, the current state of path s_{ij} is unknown and is represented by $s_{ij} = -1$. In this paper, the defender decides which k processes are selected to be monitored at each timestamp, based on historical and temporal state information of each path.

V. Strategic Monitoring Model

In this paper, a scoring mechanism is proposed to decide which k processes to monitor at each timestamp, in other words, to solve a sequential decision making problem to identify all simultaneous APT attacks as early as possible. The proposed scoring rule is derived from a classic multi-armed bandit algorithm Upper Confidence Bound (UCB) [22]. The purpose of a multi-armed bandit problem is to maximize the cumulative reward by sequentially selecting arms to pull, by assuming the existence of uncertainties in the reward of each arm [23]. The UCB algorithm integrate both the exploitation effect (observed mean reward of each arm so far) and the exploration effect (number of times each arm has been pulled) when prioritizing arms. The index of arm i at time t is calculated as follows:

$$I_i(t) = \bar{\theta}_i(t) + \sqrt{2\frac{\log t}{\tau_i(t)}} \tag{1}$$

where $\bar{\theta}_i(t)$ is the sample mean reward of arm i, which is the exploitation factor meaning that arms with higher historical reward are prioritized; $\tau_i(t)$ is the number of times arm i has been pulled, which is the exploration factor meaning that the less pulled arms are prioritized.

It is appropriate to apply UCB to solve the monitoring strategy in the threat model of this paper, because the detection of multistage APT relies on sequential decisions of monitoring to obtain temporal state of each system behavior path. Therefore, the proposed scoring mechanism is a variation of the UCB algorithm, replacing the sample mean reward with the current malevolence value of a path. One novelty of this paper is that, it extends the UCB algorithm to the multistage APT detection scenario by proposing to use a long short-term memory (LSTM) based malevolence value as the exploitation driver, and showing that this modification to the UCB algorithm is reasonable. The proposed scoring rule is formulated as follows:

$$I_i(t) = \alpha \cdot f_i([s_{i1}, s_{i2}, ..., s_{it}]) + (1-\alpha) \cdot \sqrt{2\frac{\log t}{\tau_i(t)}} \tag{2}$$

where $I_i(t)$ is the index or score of path i at time t which is a weighted sum of an exploitation term and an exploration term; $f(\cdot)$ is a neural network that takes the temporal state information of path i as input and outputs the current malevolence value the path i; the exploration term is the same as in Equation 1; and α is a constant in $[0, 1]$.

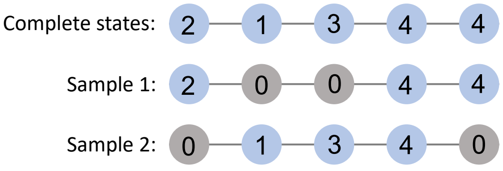


Fig. 1. Different samples generated from a sequence of complete states of a system behavior path.

A LSTM neural network is used to assign a malevolence value to a system behavior path based on its temporal states information. Samples of paths with various temporal states are used to train the neural network. In this paper, these samples are generated from complete state sequences of both benign behavior paths and APT behavior paths, by randomly selecting some states and making them unknown. For example, as shown in Fig. 1, the top row indicates that the state of an "APT attack" path at every timestamp is known, thus the sequence of states is complete (1 represents the state is benign, 2-4 represent different APT stages). To generate samples e.g. "Sample 1" and "Sample 2" resulted from incomplete monitoring in a scenario where resource constraints exist, some timestamps of the complete sequence are masked with value 0 representing that the state information at those timestamps become unknown. Note that, the class of "Sample 1" and "Sample 2" is still "APT attack". By choosing different values for the number of timestamps that are masked, samples with different incompleteness are generated, and together with their classes they are used to train the LSTM model which outputs how likely a sequence of states is an APT attack. There are two classes in the data: "1" for APT attack behavior paths; "0" for benign behavior paths. The LSTM neural network is trained to output a value in $[0, 1]$. A higher LSTM output value indicates a more suspicious behavior path, and this value is used as the exploitation factor in Equation 2: the more suspicious the behavior path is, the more likely it would be monitored next time. The second term in Equation 2 is the exploration factor, meaning that the less frequently the behavior path was monitored, the more likely it would be monitored next time.

VI. Experiments

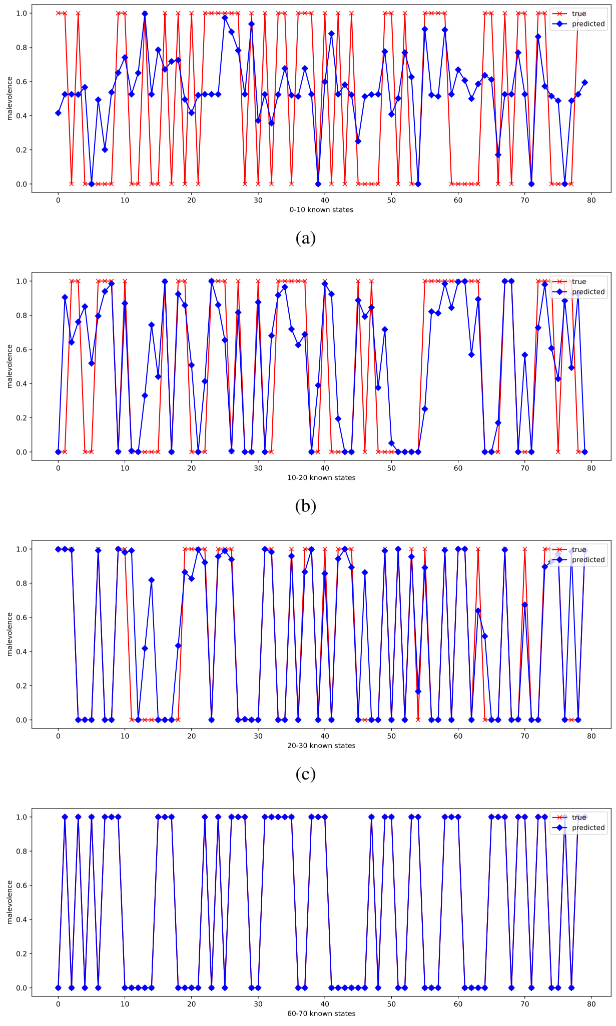
To evaluate the performance of the proposed monitoring strategy developed in this research, synthetic behavior paths with temporal states for both benign scenario and APT attack scenario are generated and implemented. Two principles are followed when generating the synthetic data: (1) all states at a timestamp ($\{1$: "benign"; 2: "APT stage 1"; 3: "APT stage 2"; 4: "APT stage 3"; 5: "APT stage 4"; 6: "APT stage 5"; 7: "APT stage 6"$\}$) can appear on a behavior path of benign scenario and APT attack scenario; (2) the temporal order of states is differentiated on a behavior path of benign scenario and on a behavior path of APT attack scenario. More specifically, the states corresponding to APT stages on the behavior path of a benign scenario are uncorrelated; however, the states corresponding to APT stages on the behavior path of an APT attack scenario are correlated in the sense that, without interruption an APT attacker gradually moves from lower APT stages to higher stages because the attacker does not have incentive to move from higher APT stages to lower stages. In addition, a random number of benign states appear between APT related states on behavior paths of both benign scenarios and APT attack scenarios. By following these principles, 100 attack paths and 100 benign paths with 80 complete temporal states are generated. Then, each complete path generates 79 incomplete paths by randomly hiding i ($i \in [1, 79]$) states. Eventually, 8000 APT attack paths and 8000 benign paths with various degrees of incompleteness are generated and used to train and test the LSTM model in this paper.

The first part of the experiment is to demonstrate that using the output of a novel LSTM neural network as the exploitation term in the proposed model is effective, in other words, the trained neural network should satisfy the same property as the exploitation term defined in the classic UCB algorithm. In the classic UCB algorithm, the property of the exploitation term is that, as an arm is pulled more often, the estimation to its reward becomes more accurate. Therefore, the proposed LSTM neural network is tested on behavior paths of which the defender has different degrees of information incompleteness. If the LSTM neural network is effective, it is desired that its estimation to the malevolence of paths is more accurate as the defender's information incompleteness of paths decreases.

The second part of the experiment is to demonstrate the performance of the proposed monitoring strategy. Before evaluation, 20 behavior paths are randomly selected from the data set, including four APT attack paths and 16 benign paths. Then, the the proposed monitoring strategy, a random strategy and a greedy strategy are evaluated respectively, in terms of when the attack paths are detected and the number of false positives. In the experiment setting, only five paths can be monitored at each timestamp, a random strategy means that the five monitored paths are randomly selected; a greedy strategy means that the paths with five highest malevolence scores are selected, in other words, only the exploitation effect in the proposed strategy is considered; the proposed strategy selects the 5 paths with the highest scores where the score is defined as the weighed sum of the exploitation factor and the exploration factor are selected. In addition, two termination conditions are applied when implementing the proposed strategy: (1) when the malevolence score of a path is greater than β, the path is determined as an APT attack and will no longer be a candidate of being monitored; (2) when the malevolence score of a path is smaller than γ, the path is determined as a benign scenario and will no longer be a candidate to be monitored. By testing different sets of parameter values, the best parameter values of the proposed strategy are used to compare the proposed strategy with other strategies, including the weight parameter α in Equation 2 ($\alpha = 0.82$), and two threshold parameters β ($\beta = 0.9$) and γ ($\gamma = 0.05$).

VII. Results and Analysis

Fig. 2 shows the performance of the LSTM neural network in the proposed strategic monitoring model. From Fig. 2(a) to Fig. 2(d), the number of known states of paths increases, in other words, the defender's information incompleteness of paths decreases. The red line represents the true malevolence of paths, and the blue line represents the predicted malevolence of paths by the LSTM neural network. It can be seen that the difference between the true malevolence values and predicted malevolence values by the LSTM neural network becomes smaller, when the number of known states increases. Therefore, the estimation of the LSTM neural network to the malevolence of a behavior path becomes more accurate along the path that is monitored more frequently, making the output of the LSTM neural network an effective exploitation term in the proposed monitoring strategy model.



(a)

(b)

(c)

(d)

Fig. 2. Comparison of true and predicted malevolence of paths with various number of known states.

To demonstrate the performance of the proposed monitoring strategy, it is compared with a random strategy and a greedy strategy, and the results of the three strategies including the paths monitored at each timestamp as well as when attacks are detected are visualized in Fig. 3, Fig. 4, Fig. 5 respectively. Note that, in Fig. 3 to 5, each row i represents a behavior path and the rows with light red shade means that the row represents an APT path. Each column j represents a timestamp, and at each time j, the defender can pick $\max(5, u_j)$ paths to monitor where u_j is the number of undetermined paths at time j. For a position (i, j) where $i \in [1, 20]$ and $j \in [1, 80]$, if it is in blue, it means path i is monitored at timestamp j; if it is in red, it means path i is classified as an APT attack scenario at timestamp j and it will no longer be monitored which is represented by marking its future states as grey; if it is in green, it means path i is classified as a benign scenario at timestamp j and it will no longer be monitored which is represented by marking its future states as grey. Therefore, the number of undetermined paths u_j is the number of paths that are not in grey at time j.

Fig. 3 shows the results of using a random strategy, which means that the defender randomly picks paths to monitor at each timestamp. The four APT attacks paths are detected at timestamp 57, 50, 49 and 41. In addition, the number of false positives is 2.

Fig. 4 shows the result of using a greedy strategy, which

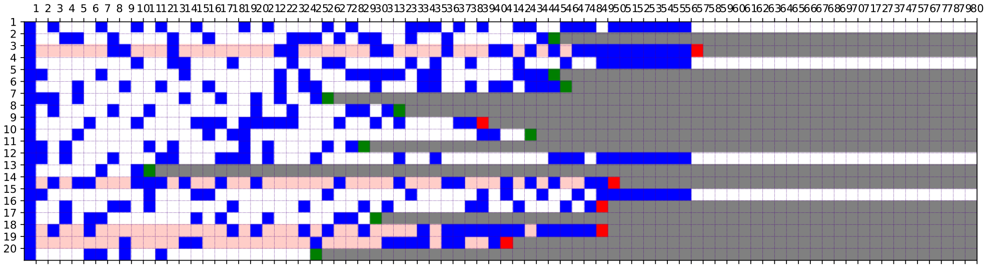


Fig. 3. Performance of the random monitoring strategy.

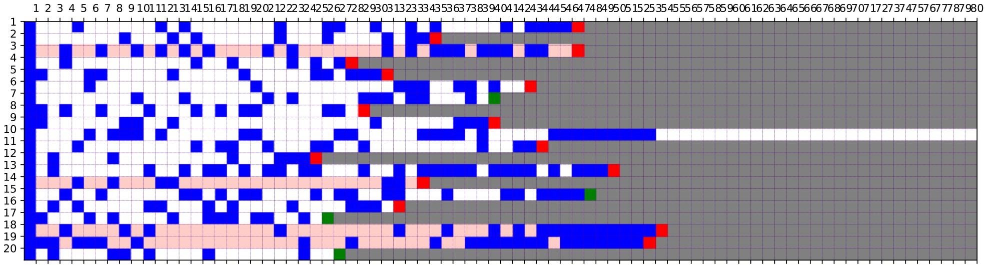


Fig. 4. Performance of the greedy monitoring strategy.

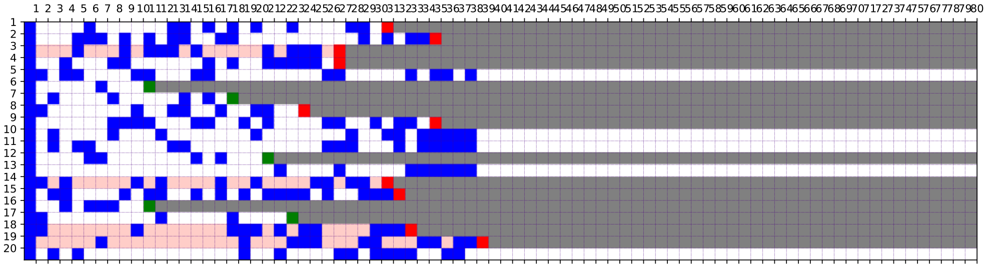


Fig. 5. Performance of the proposed monitoring strategy.

means that the defender picks paths with the highest malevolence values predicted by the LSTM neural network. The 4 APT attacks paths are detected at timestamp 47, 34, 54 and 53. In addition, the number of false positives is 11.

Fig. 5 shows the result of using the strategy developed by this paper, which means the defender considers both exploitation and exploration then picks paths with the highest values calculated by Equation 2. The 4 APT attacks paths are detected at timestamp 27, 31, 33 and 39. In addition, the number of false positives is 6.

Compared with the other two strategies, the monitoring strategy proposed in this paper detects all 4 APT attacks significantly earlier, specifically nearly 17 timestamps earlier than the random strategy, and 15 timestamps earlier than the greedy strategy. The key to the efficiency of the proposed strategy is that by considering exploration when the malevolence estimation is not as accurate in early timestamps, the defender is able to identify some benign paths quickly, which reduces the number of competitors of limited monitoring resources. However, the random strategy treats benign paths and attacks paths equally and the greedy strategy can be misled by inaccurate malevolence in earlier stamps, thus they are less efficient in the detection of simultaneous APT attacks.

When facing advanced attacks like APT, false positive is more acceptable compared to false negative, because missing an APT attack is more devastating than looking into a benign activity which is falsely classified as attack. Regarding the number of false positives in the experiments, the proposed strategy is better than the greedy strategy, but is worse than the random strategy. This is as expected, because from Fig. 3, it takes longer and relies on more information for the defender to determine the identity of a path when using the random strategy. With more information, the malevolence estimation to a path is more accurate at shown in Fig. 2, however, the random strategy is the least efficient regarding the time needed to detect simultaneous APT attacks. Therefore, overall the proposed strategy outperforms the other two strategies regarding the metric of efficient detection of simultaneous APT attacks with limited resources. And the improvement on other metrics is left as a future extension.

VIII. Conclusion

The work in this paper addresses the issue of resource constraints in the detection of multiple simultaneous APT attacks. It proposes a monitoring strategy to efficiently detect APT attacks with incomplete information about activities in a system. The key of the proposed strategy is that it considers both the “exploitation” effect and the “exploration” effect in resource allocation, which is beneficial for finding the optimal strategy in circumstances with high uncertainties. The novel contributions of this work to address the research question are as follows.

First, differing from existing works that allocate security resources based on the estimated malevolence of system activities only, this work emphasizes the importance of “exploration” in APT detection, because the perception to advanced and stealthy attacks based on its earlier stage information is usually not accurate. This work is the seminal paper to consider both the “exploitation” effect and the “exploration” effect in monitoring resource allocation, and apply the classic multi-armed bandit algorithm, UCB, to solve optimal resource allocation problems in APT defense.

Second, this work proposes a novel LSTM neural network to measure the malevolence of a path on the provenance graph based on its incomplete temporal information, and replaces the exploitation term in the classic UCB algorithm with this malevolence value. The experimental results show that by using the proposed monitoring strategy, multiple simultaneous APT attacks are detected more efficiently than using a random strategy and a greedy strategy, regarding the time needed to detect same number of attacks.

Although the proposed model shows the advantage of detecting simultaneous APT attacks efficiently with limited resources, a future extension to this work is to enhance the model in terms of more metrics, for example, reducing false positives by exploring more features of APT to differentiate it from benign activities more effectively.

References

[1] H. Zhang *et al*, “Efficient strategy selection for moving target defense under multiple attacks,” *IEEE Access*, vol. 7, pp. 65982-65995, 2019.

[2] A. Alshamrani *et al.*, “A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851-1877, 2019.

[3] Lockheed Martin, “The cyber kill chain,” Available at https://www.lockheedmartin.com/us/what-we-do/aerospace-defense/cyber/cyber-kill-chain.html, Accessed Dec 2022.

[4] P. Giura and W. Wang, “A context-based detection framework for advanced persistent threats,” In *Proc. 2012 International Conference on Cyber Security*, pp. 69-74, 2012.

[5] MITRE ATT&CK, Available at https://attack.mitre.org, Accessed Dec 2022.

[6] M. Roesch, “Snort: Lightweight intrusion detection for networks,” *Lisa*, vol. 99, no. 1, pp. 229-238, 1999.

[7] P. Bhatt *et al.*, “Towards a framework to detect multi-stage advanced persistent threats attacks,” In *IEEE 8th international symposium on service oriented system engineering*, pp. 390-395, 2014.

[8] E. M. Hutchins, M. J. Cloppert and R. M. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, Apr. 2011.

[9] M. M. Sadegh *et al.*, “Holmes: real-time apt detection through correlation of suspicious information flows,” In *Proc. 2019 IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, May 2019, pp. 1137-1152.

[10] N. Villeneuve and J. Bennett, “Detecting APT activity with network traffic analysis,” *Trend Micro Incorporated Research Paper*, pp. 1-13, 2012.

[11] F. Shen, L. Perigo and J. Curry, “SR2APT: A Detection and Strategic Alert Response Model Against Multistage APT Attacks,” *Security and Communication Networks* (forthcoming), DOI:10.1155/1969/6802359.

[12] M. Marchetti *et al.*, “Analysis of high volumes of network traffic for advanced persistent threat detection,” *Computer Networks*, vol. 109, pp. 127-141, Nov. 2016.

[13] N. Nissim, “Detection of malicious PDF files and directions for enhancements: A state-of-the art survey,” *Computers & Security*, vol. 48, pp. 246-266, Feb. 2015.

[14] J. V. Chandra, N. Challa and S. K. Pasupuleti, “A practical approach to E-mail spam filters to protect data from advanced persistent threat,” In *Proc. 2016 international conference on circuit, power and computing technologies (ICCPCT)*, pp. 1-5, 2016.

[15] H. N. Eke *et al*, “Framework for Detecting APTs Based on Steps Analysis and Correlation,” *Security and Resilience in Cyber-Physical Systems*, pp. 119-147, 2022.

[16] M. AbuOdeh *et al.*, “A novel AI-based methodology for identifying cyber attacks in honey pots,” In *Proc. AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 15224-15231, May 2021.

[17] J. Sexton, C. Storlie and J. Neil, “Attack chain detection,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 8, no. 5-6, pp. 353-363, Oct. 2015.

[18] G. Brogi and V. V. T. Tong, “Terminaptor: Highlighting advanced persistent threats through information flow tracking,” In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-5, 2016.

[19] I. Ghafir *et al.*, “Detection of advanced persistent threat using machine-learning correlation analysis,” *Future Generation Computer Systems*, vol. 89, pp. 349-359, Dec. 2018.

[20] M. V. Dijk *et al.*, “FlipIt: The game of stealthy takeover,” *Journal of Cryptology*, vol. 26, no. 4, pp. 655-713, 2013.

[21] M. Zhang *et al.*, “A game theoretic model for defending against stealthy attacks with limited resources,” In *International Conference on Decision and Game Theory for Security*, pp. 93-112, 2015.

[22] P. Auer, N. Cesa-Bianchi and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2, pp. 235-256, 2002.

[23] Z. Liu *et al.*, “Incentivized exploration for multi-armed bandits under reward drift,” In *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4981-4988, 2020.

Deep Learning Algorithm based Wearable Device for Basketball Stance Recognition in Basketball

Lan Jiang[1], Dongxu Zhang[2]*
Basic Teaching Department, Sichuan Film and Television University, Chengdu, 610000, China[1]
College of Modern Services, Mianyang City College, Mianyang, 621000, China[2]

***Abstract*—With the continuous improvement of technology, modern sports training is gradually developing towards precision and efficiency, which requires more accurate identification of athletes' sports stances. The study first establishes a classification structure of basketball stance, then designs a hardware module to collect different stance data by using inertial sensors, thus extracting multidimensional motion stance features. Then the traditional convolutional neural network (CNN) is improved by principal component analysis (PCA) to form the PCA+CNN algorithm. Finally, the algorithm is simulated and tested. The outcomes demonstrated that the average discrimination error rate of the improved PCA+CNN algorithm in the Human 3.6M dataset was 3.15%, which was a low error rate. In recognition of basketball sports pose, the wearable based on the improved algorithm had the highest accuracy of 99.4% and took the quietest time of 18s, which was better than the other three methods. It demonstrated that the method had high discrimination precision and recognition efficiency, which could provide a reliable technical means to improve the science of basketball sports training plan and training effect.**

Keywords*—*Deep learning; wearable devices; basketball; sports pose; CNN

I. Introduction

When basketball players training and contesting program is to be devised then developing a scientific and rational training program based on individual circumstances is the basis for improving their skills[1]. Considering player's actual ability, traditional training methods use experience and theory as a reference for the development of the program. However, this model is highly subjective and requires much time to analyze athletes' gestures, which is hard to meet the requirements of modern sports training. When coaches control the various postures of different athletes accurately, the training effect will be greatly raised, so it is important to collect and analyze posture data to achieve accurate identification. Basketball posture is one of the human postures, and today's human posture recognition is based on image acquisition and inertial sensor-based posture recognition [2]. The image acquisition stance recognition is achieved through camera image, video acquisition or different classifiers, whose technical maturity is high. But it has drawbacks that it needs a large amount of equipment and it is difficult to be applied widely. On the other hand, inertial sensor recognition is achieved by wearing a data acquisition sensor, which transmits the collected data in real time. Finally, the processing terminal completes the recognition. Its high recognition efficiency and low requirements for the use of the environment have become a research hotspot in this field [3]. In addition, with the increasing penetration of deep learning in various fields, the CNN, it contains, is used in discrimination widely. Therefore, in order to improve the recognition accuracy of the deep learning algorithm in basketball posture, wearable devices are applied in basketball training. This paper studies the recognition of basketball posture based on inertial sensors and convolution neural network. At the same time, the convolution neural network is improved by principal component to provide more high-performance auxiliary training technology for basketball.

II. Related Works

Deep learning algorithms are highly capable of learning, cover a wide range of areas, and can exhibit greater stability as more data is available. Through deep learning algorithms, the recognition of motion gestures has attracted huge attention from many professionals recently. And a series of forward-looking and practical research outcomes have been achieved. Hu B et al. developed a gesture recognition system based on UAV control for dynamic gesture recognition instead of human-machine interaction. This system used 8-layers CNN, 5-layers fully connected network and 2-layers fully connected network that could convert 4-D spatio-temporal data into 1-D and 2-D matrices to model the data of gesture sequences. The experimental outcomes demonstrated an average accuracy of 89.1% in the unscaled dataset [4]. Azad R's team designed a multi-level temporal sampling method based on depth sequence key frames for gesture recognition in computer vision. And they combined it with weighted depth motion maps to extract spatio-temporal information in the sequences by accumulating weighted absolute differences in successive frames. The outcomes demonstrated that the method had great precision [5]. Sun Y et al. made a posture discrimination processing framework based on a radar system with a shallow CNN whose input is a feature cube. And it could feed gesture contours into the CNN through lower latency. Experimental outcomes demonstrated that the framework achieved 92.08% accuracy in performing real-time classification of 12 gestures [6]. Zhang Y's team addressed the human-computer interaction in wearable devices. A benchmark dataset called EgoGesture was proposed to address the problem of human-computer interaction in wearable devices. In static and dynamic gesture recognition of different scenarios, it performed well with sufficient variability and realism when training deep neural networks [7]. The outcomes demonstrated that the method could achieve the discrimination of different body movements in basketball with a high accuracy rate [8]. Pan T Y et al. developed a gesture discrimination way based on multiple inertial measurement unit transducers, which could record the rotation information and acceleration of hand joints. And the average precision in the discrimination of basketball referee signals was 90.02% [9].

Zhao Y et al. designed a 3D position estimation method with an integrated ankle sensing device consisting of a magnetometer, accelerometer, gyroscope and barometer. And they added vertical variables to the adaptive multimodal stride in combination with it. Experimental outcomes demonstrated that it raised the applicability and precision of pedestrian horizontal position estimation [10]. Neethu P S's team applied a convolutional neural network classification method to human hand gesture detection and recognition, which extracted fingertip features in hand images by a connected component analysis algorithm. The outcomes demonstrated that the method had good operational performance [11]. Zhang W et al. applied deep learning networks to gesture recognition for human-computer interaction in hand gestures. It learned short-term and long-term features in the video input and feeded into a CNN for feature extraction. Experiments on the Jester and Nvidia datasets demonstrated its high accuracy [12]. Lian C et al. designed an IoT wristband to facilitate quantitative shooting action guidance for basketball players, which used a miniature inertial measurement unit sensor and was capable of collecting. The outcomes demonstrated that it was able to achieve 98.0% accuracy for layups, free throws, positioning shots and jump shots. And the overall accuracy was nearly 97.5% for 18 out of 18 shooting motions [13]. Kamel A's team developed a way for discriminating depth maps and pose data through convolutional neural networks, which used two inputs to describe the action representation. That is, a depth motion image that accumulates human movements and a motion joint descriptor that represents body joints over time. And the action prediction outcomes are generated from three CNN channels, which are experimentally demonstrated to be around 6.84% higher than general recognition methods [14]. Adithya V et al. applied CNN in deep learning to solve the problem of automatic sign language recognition by capturing gesture images and deriving complex feature descriptors. The outcomes demonstrate that it has good recognition accuracy on the hand gesture dataset [15].

To sum up, most researchers had proposed corresponding recognition methods for motion, gesture and human posture recognition. And these methods had improved the convolutional neural network in deep learning, which had achieved good application results. However, few people combined convolution neural network with inertial sensors, and there was a lack of research on the application of wearable devices in basketball motion recognition. And there was little research on the recognition of basketball posture. Therefore, the research first improves the action recognition effect of convolutional neural network, and combines it with wearable devices to apply it in basketball training, so as to improve the level of basketball training.

III. Basketball Pose Recognition based on Deep Learning Algorithms

A. *Basketball Posture Recognition Model and Wearable Device Design*

There are very complex and variable body movements during basketball. To carry out effective and accurate posture recognition, it is necessary to establish a scientific and comprehensive posture classification [16]. Based on the various limb states of the basketball player, the first two categories are static and athletic [17]. The athletic state is athletes' performance of different basketball actions, where the limbs remain in motion. While the resting state is the state in which the athlete's limbs do not perform any action and are strictly static. To effectively discern sporting gestures, the study has implemented a progression of two levels. In the first level, there are two types of movement gestures, namely transient and continuous, based on whether they are cyclical or not. And in the second level, it is further divided into seven stances, namely running, jumping, shooting, catching, passing, walking and dribbling, according to whether the movement state is lower or upper limb. The recognition of basketball movement stance actually becomes a recognition of the movement stance proposed by the automatic recognition. The classification of basketball sports stance is demonstrated in Fig. 1.

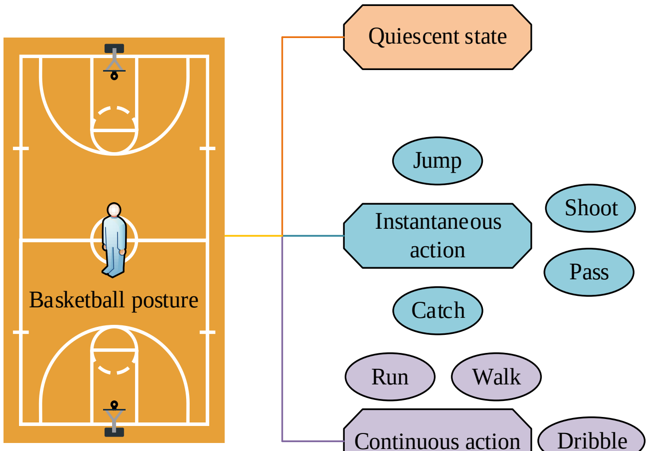


Fig. 1. Classification structure of basketball posture.

After determining the individual basketball postures, the wearable device is optimized. Timely and accurate gesture data acquisition is essential for accurate identification, so the research designs a basketball gesture data acquisition module based on inertial sensors in the wearable device. The data is first collected by using angular velocity, magnetic and acceleration sensors fixed to the basketball player. Then it is transmitted to the terminal device using wireless sensors for gesture discrimination. The hardware of the data acquisition includes data acquisition and transmission, containing a base station for data transmission and four nodes for information collection. Together form the data acquisition points which contains the acceleration and palstance information of the human body, the tri-axial accelerometer, tri-axial gyroscope MPU3050 and magnetometer LSM303DLH are collected. The wireless transceiver nRF24L01 is the core component of the data transmitting site and is for receiving information from the nodes and transmitting this data to the data terminals via the wireless network. For the information collection module, the microcontroller STM32F103 is for the core processing functions. At the same time, a 3.7V lithium-ion battery completes the energy supply for this module, the hardware structure is shown in Fig. 2.

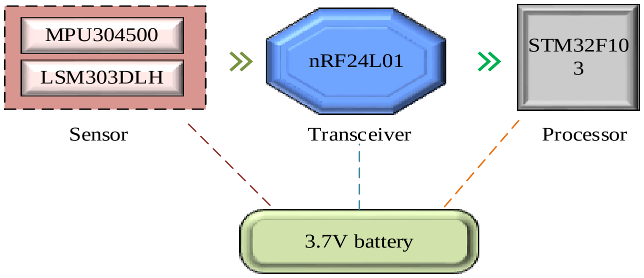


Fig. 2. Hardware structure diagram of basketball player posture data acquisition module.

In the data acquisition module, the signal transmission is carried out containing two parts. Firstly, the information conveying site receives the human posture data from the sensor nodes. Secondly, it is transmitted to the processing terminal. This part needs to minimize data collision rates and retain data to prevent large amounts of data loss, thus improving the accuracy of the collected data. The star topology network is the basis for the signal conveying between the processing terminal and the information conveying station, which is carried out via a time division multiplexing protocol. And the calibration of the clock deviations of the different nodes to maintain time uniformity is key to this. In conjunction with the classification of the basketball stance, four sensor nodes are set up in the legs and arms to obtain accurate data on the legs and arms of the athlete to collect magnetic field strength, acceleration and palstance data. The acceleration vector addition and the palstance vector addition for the n sampling point are calculated as demonstrated in equation (1).

$$\begin{cases} a_n = \sqrt{(a_n^x)^2 + (a_n^y)^2 + (a_n^z)^2} \\ p_n = \sqrt{(p_n^x)^2 + (p_n^y)^2 + (p_n^z)^2} \end{cases} \tag{1}$$

In Eq. (1), a_n^x, a_n^y and a_n^z correspond to the axial accelerations of x, y and z at the first n point, respectively, while p_n^x, p_n^y and p_n^z represent the corresponding angular velocities. a_n and p_n are the combined motion of the accelerations and palstance, respectively. The vector sums and three vectors of acceleration and angular velocity are then combined to form an 8-dimensional feature matrix. When the sampling point is N, each sample owns a feature matrix of $N \times 8$. The mean and variance are the time domain characteristics identified, and the mean and variance of each point is calculated as demonstrated in Eq. (2).

$$\begin{cases} \delta^2 = \frac{1}{N}\sum_{n=1}^{N}(a_n - \mu_a)^2 \\ \mu_a = \frac{1}{N}\sum_{n=1}^{N} a_n \end{cases} \tag{2}$$

In equation (2), δ^2 represents the mean value and μ_a is the variance. The time domain data is then transformed into the frequency domain data by the Fourier transform, which is calculated as demonstrated in Eq. (3).

$$S(n) = \sum_{i=0}^{N-1} a_i e^{-j\frac{2\pi}{N^n}} \tag{3}$$

In Eq. (3), $S(n)$ represents the value of adoption point n, where the frequency domain is located. The time domain features required for basketball gesture discrimination is actually the peaks of the Fourier transform, as demonstrated in equation (4).

$$f = \frac{Kf_s}{N} \tag{4}$$

In equation (4), K represents the quantity of frequency domain sampling point and f_s represents the sampling frequency. After the motion stance features are extracted, a 32-dimensional feature parameter set is obtained. For these parameters as a whole, there are some features with low or even no correlation with basketball motion pose. And there are also some features with redundant information between them, which seriously affect the classification efficiency and performance. Therefore, the study needs to make feature dimensionality further being reduced to achieve better recognition outcomes.

B. Motion Pose Recognition based on Convolutional Neural Networks

After obtaining the basketball pose data features, a CNN is used to downscale and identify the features. A CNN includes convolutional layers, pooling layers, activation functions and fully connected layers [18]. In general, the upper layers are convolutional layers, and then one or more fully connected layers are cascaded. All fully connected and convolutional layers are followed by an activation function, which is used as a non-linear transform. At the same time, pooling layer is after convolutional layer, which serves to reduce the amount of data contained in the intermediate outcomes. In a convolutional neural network, the convolutional layer is the most basic unit [19]. Under the action of mapping, a well-trained convolutional layer automatically and efficiently extracts features from the data and transfers the original data to the hidden feature space. The fully connected layer is the classifier in CNN, which maps the learned features to the sample data labeling space. At the same time, the essence of the operation in this layer is the multiplication of vectors and matrices. If it is transformed into vector form after the straightening operation, only the convolutional layer feature map can be input to the fully connected layer [20]. The pooling layer differs significantly from the fully connected and convolutional layers. Because it has no bias or weight parameters and is mostly used to lowering feature map. Thereby, the redundancy is eliminated significantly in the feature map. Among the pooling approaches, maximum and average pooling are more widely used. Similar to the convolutional layer, the pooling mode slides through a box n_1 high and n_2 wide, over all the input channels of the feature map. And it finds the maximum or average value in this box and the size of the box are usually consistent with the step size of the slide. The activation function acts as the key step in the convolutional neural network to complete the non-linear mapping. And almost all fully connected and convolutional layers must go through it. The missing activation function will cause the stack of fully connected layers and multiple volume layers. It is equivalent to the multiplication of multiple matrices to finally get a matrix, resulting in the original data not being activated to obtain a strong fit. The basic structure of CNN is demonstrated in Fig. 3.

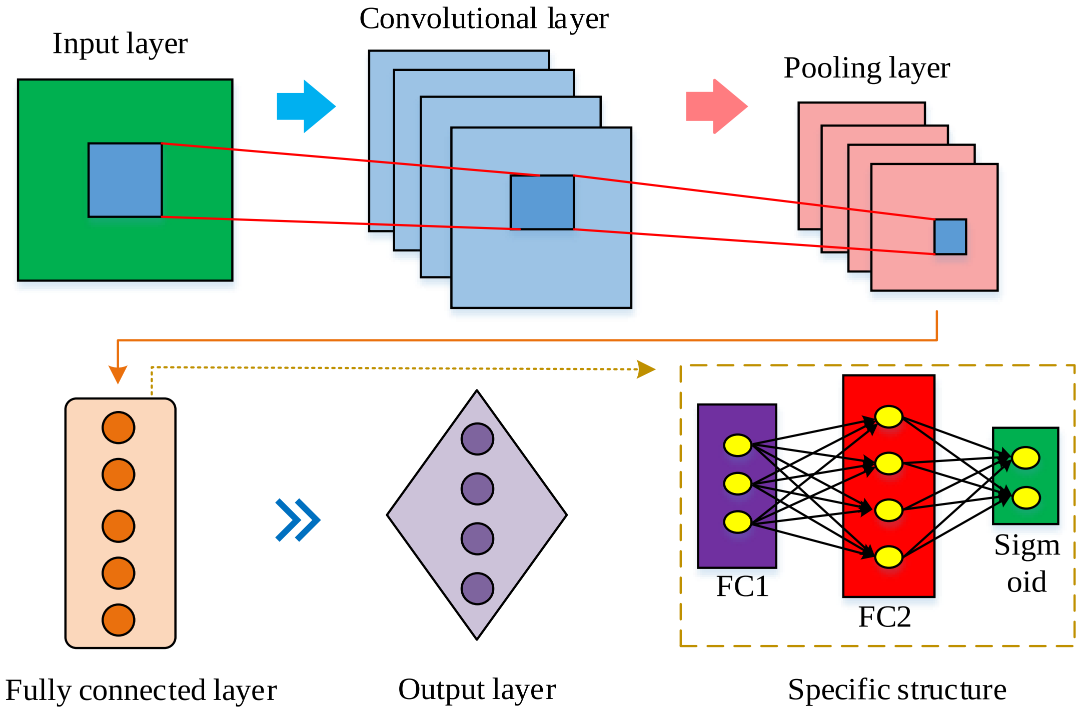


Fig. 3. Basic structure of convolutional neural network.

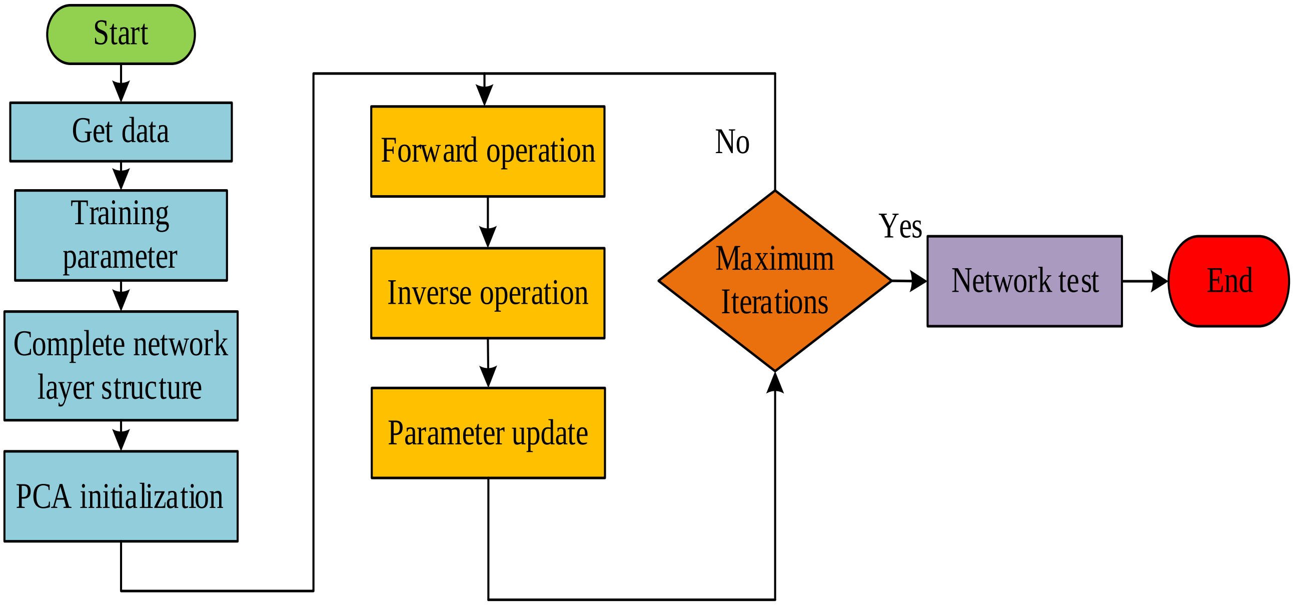


Fig. 4. Algorithm flow of CNN with principal component analysis.

The multi-layer perceptron is the basic structure of a fully connected layer in CNN, including input and output layer, with a weight matrix connecting the layer to the layer [21]. The input layer is formed by the upper layer feature map after vectorization. After the inner product of the input vector and the weight matrix, and mapped by the activation function, the outcome is obtained for the output layer. The calculation is demonstrated in Eq. (5).

$$y_{pj}^{l} = f\left(\sum_{i=0}^{N^{l-1}} X_{pj}^{l-1} \cdot b_{j}^{l} + w_{ji}^{l}\right) \tag{5}$$

In Eq. (5), f represents activation function, N is the quantity of neurons, b represents the neuron bias, w is the weights. l is the convolutional layer sequence, j is the feature map sequence, and p is the training sample sequence. As the number of layers in the network increases, the extracted features become more abstract and discriminative, and these features facilitate the classification of basketball poses. At the same time, the features give feedback to the extraction of shallow features [22]. The main problem with convolutional neural networks is the network parameters training and updating. To raise the ability of basketball posture recognition, principal component analysis is introduced to integrate with convolutional neural networks. Principal component analysis is a statistically based feature extraction method that can filter features with a high correlation, i.e. selecting the optimal sign. At the same time, the principal component analysis method has a high similarity with the learning outcomes of self-coding neural networks. Therefore, the study uses this method to perform multiple calculations on all convolutional kernel sets in layers. The convolutional kernel parameters are initialized to optimize the convolutional neural network performance. The optimized algorithm is demonstrated in Fig. 4.

In Fig. 4, the improved convolutional neural network first completes all network layer structures based on training and test data, then the training parameters can be defined. From there, the initialized network parameters are analyzed by principal component analysis, followed by forward and backward operations for network training. Then parameter updates are implemented. Finally the network is tested and concluded by determining whether iterations has been reached to complete the network. The principal component analysis method calculates the input layer feature vector set V and then orients the first L of the vector set. These can be used as the principal component feature vector of the input sample data set, thus forming the convolutional kernel set $V1$ of the $C1$ layer, as demonstrated in Eq. (6).

$$\begin{cases} \min \left\| Y - V_1 V_1^T Y \right\|_F^2 \\ V_1^T V_1 = I \end{cases} \tag{6}$$

In Eq. (6), I represents the unit matrix and V_1 represents the training data fed into the convolutional neural network. In the convolutional layer $C2$, the same method is used to treat the feature maps from the previous layer as the sample set for the principal component analysis. So it outcomes in the convolutional group $V2$ of the $C2$ layer. According to the convolution and size, and used as the initialization values for the corresponding layer's convolution kernel, all the column vectors contained in all the computed feature vectors are arranged, which finally completes the convolution kernel initialization. The optimized convolutional neural network model is in Fig. 5.

Finally, the optimized convolutional neural network is applied to the wearable device classifier to obtain different basketball motion pose types, thus completing the output of the recognition outcomes.

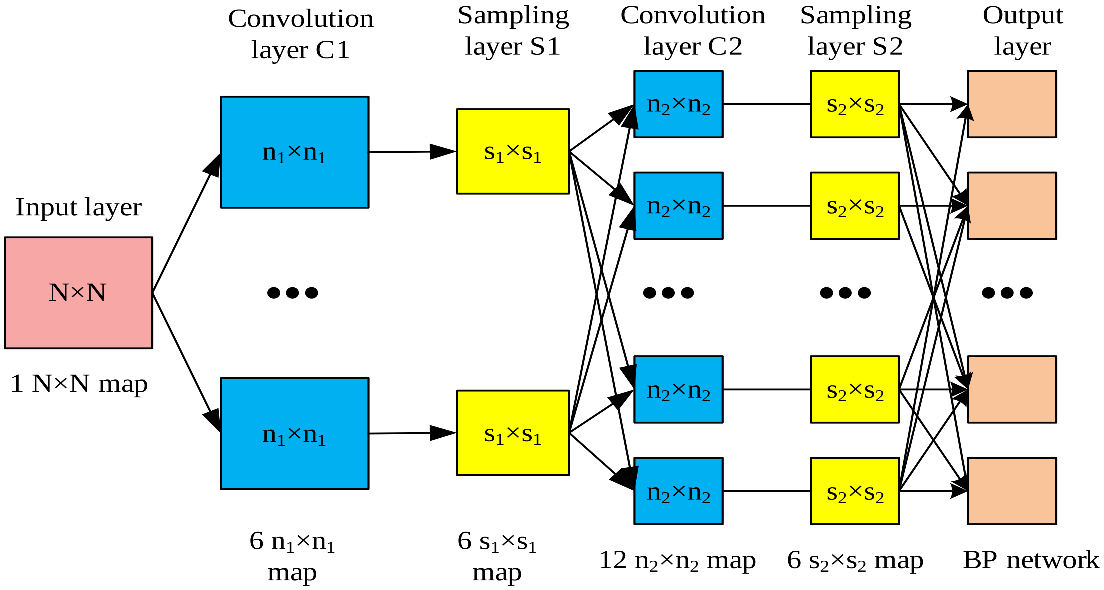


Fig. 5. CNN optimized by introducing principal component analysis.

IV. Analysis of Basketball Posture Recognition Outcomes based on Improved CNN

The function of the PCA-improved CNN is first examined. The simulation environment is a discrete graphics card GeForce MX150, 7th generation Intel i5 processor, Matlab (R2016b), 2 GB GDDR5, 256 GB PCIe SSD, memory 8 GB DDR4, Deep Learning-Toolbox. The PCA+CNN is compared with the classical CNN algorithm and the selected dataset is Human3.6M dataset. This dataset is a large public dataset for 3D human pose estimation, containing 3.6 million human poses and corresponding images. A comparison of the outcomes of the two algorithms is in Table I.

Table I demonstrates the comparison outcomes of the error rate and time between CNN algorithm and PCA+CNN for five runs on the Human3.6M dataset. In Table I, in terms of running time, the average time of the PCA+CNN and the classical CNN over the five experiments was 291.60s and 289.11s respectively, which shows a small difference. In terms of error rate, the average error rate of PCA+CNN algorithm and classical CNN were 3.15% and 4.68% respectively. The former had a lower error rate and the error rate of each experiment was lower than that of the traditional CNN. Then the number of iterations with mean square error outcomes were taken from one of the five experiments. And the outcomes of the other four experiments were graphically similar to the selected experimental outcomes. The comparison of the two algorithms was demonstrated in Fig. 6.

In Fig. 6, the red line was the experimental run of the classical CNN algorithm, while blue line represented the run of PCA+CNN. The horizontal coordinate was 6000 samples after 5 iterations with 50 training samples input each time. And the vertical coordinate was the mean square error for each input training sample number of 50. From Fig. 6, in the initial period of iteration, the mean square error obtained by the CNN algorithm was greater than that produced by PCA+CNN. The difference was most significant when the number of iterations was between 145 and 1000. While after the number of iterations exceeds 1000, the mean square error of both algorithms changes less. But the error convergence curve of PCA+CNN always stayed below the CNN experimental outcomes. At the same time, when the mean square error was 0.0479, iterations of PCA+CNN was only 4261. While the number of iterations of conventional CNN was 5893, which is 1632 times higher than the former. So the error rate obtained by PCA+CNN algorithm was smaller than that of the classical CNN algorithm. The two algorithms were then simulated for the MPII Human Pose dataset, evaluated by loss function curves and accuracy curves. 410 human activities were included in the MPII Human Pose dataset, and each image was clearly labelled with an activity, making it a dataset for human pose recognition. For the accuracy metric, it was the ratio of the number of correctly recognised poses to the total. The loss function was an important metric for evaluating how well the deep learning training was done. Here, the loss function was mainly for the test set and was judged on the basis that the loss functions of both the test and training sets converge and the difference was small. The loss function and precision curves are in Fig. 7.

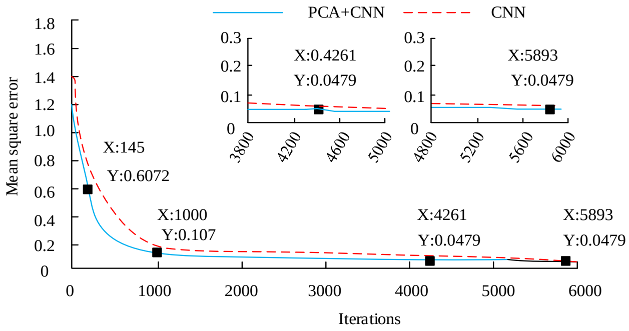
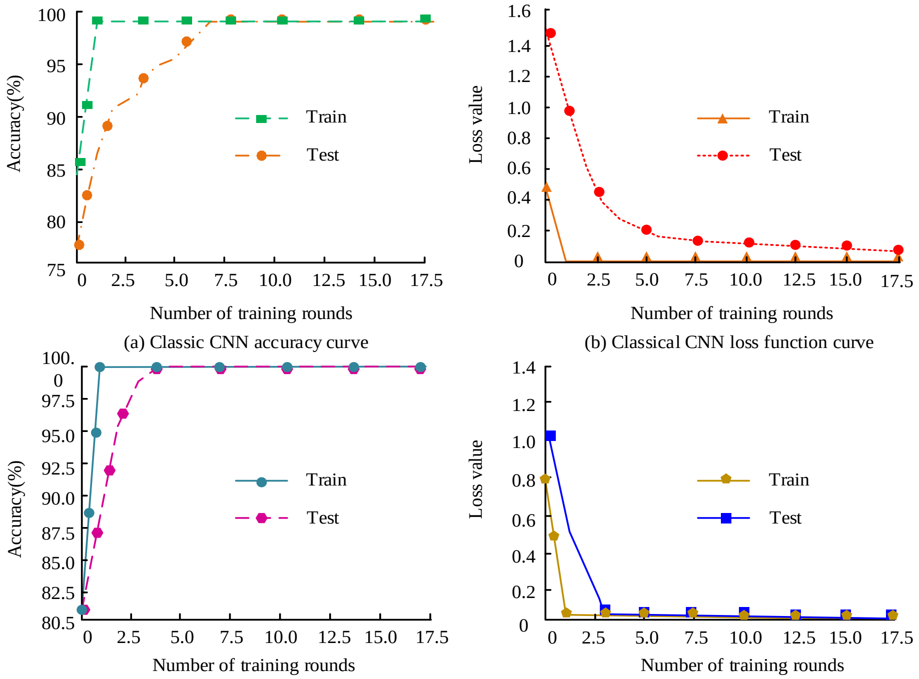


Fig. 6. Error convergence curve outcomes of CNN algorithm and PCA+CNN algorithm.

TABLE I. Comparison of Running Outcomes of PCA+CNN and Classical CNN Algorithms on Human 3.6M Dataset

Algorithm	Number of experiments	Running time/s	Error rate/%
PCA+CNN	1	291.53	3.04
	2	290.14	3.22
	3	293.52	3.13
	4	292.37	3.28
	5	290.45	3.10
CNN	1	289.12	4.35
	2	287.56	4.47
	3	289.73	4.92
	4	290.51	5.01
	5	288.64	4.63



(a) Classic CNN accuracy curve

(b) Classical CNN loss function curve

(c) PCA+CNN accuracy curve

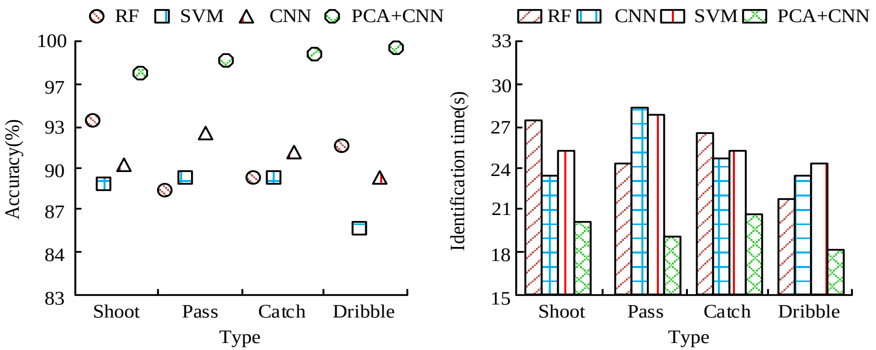
(d) PCA+CNN loss function curve

Fig. 7. Loss function and accuracy curve outcomes of CNN and PCA+CNN algorithms.

Fig. 7(a) and Fig. 7(b) demonstrated the accuracy curve and loss function curve of the classical CNN algorithm, respectively, while Fig. 7(c) and Fig. 7(d) demonstrated the precision and loss-function curve of the PCA+CNN algorithm. From Fig. 7(a), the classical CNN achieved a fit at round 7, where the accuracy was 99%. In Fig. 7(b), the loss function curve starts to smooth out and converge to 0 in round 8. In Fig. 7(c) and Fig. 7(d), the PCA+CNN converged near round 5 with 100% accuracy, which was faster and more accurate than the CNN algorithm, and the loss function also converged faster than the CNN algorithm, which has better performance. Finally, for example validation, sample data was collected from 10 athletes in eight stances: dribbling, walking, shooting, no movement, jumping, passing, catching and running. 80 sets of repeated data were collected for each movement, and a total of 6400 data samples can be obtained. The data was collected by performing a series of pre-determined basketball stances in conjunction with the participant's own exercise habits. All postures included both upper and lower limb movements, and the lower and upper limb movements were also analyzed separately for identification purposes. For checking the validity of the posed way, Support Vector Machine (SVM) and Random Forest (RF) were for comparison in the example validation. The outcomes of the four methods for the recognition of upper limb basketball postures are demonstrated in Fig. 8.

Fig. 8(a) and Fig. 8(b) the upper limb recognition accuracy and recognition time outcomes for the four algorithms respectively, with the horizontal coordinates all representing the types of upper limb basketball sports poses, i.e. shooting, catching, passing and dribbling. From Fig. 8(a), it can be seen that among the four types of upper limb sports pose recognition, RF, SVM and CNN are stable between 88~94%, 86~89% and 89~93% respectively, while the accuracy of PCA+CNN is above 97% for all of them, and the recognition accuracy for passing is as high as 99%. From Fig. 8(b), among the four methods, the PCA+CNN algorithm recognized the four types of upper limb movement poses in 20s, 19s, 21s and 18s, respectively, all of which were faster than the other three methods, with a maximum lead of 10s and high operational efficiency. The recognition outcomes of the four methods for the three lower limb motion postures of jumping, walking and running are demonstrated in Fig. 9.

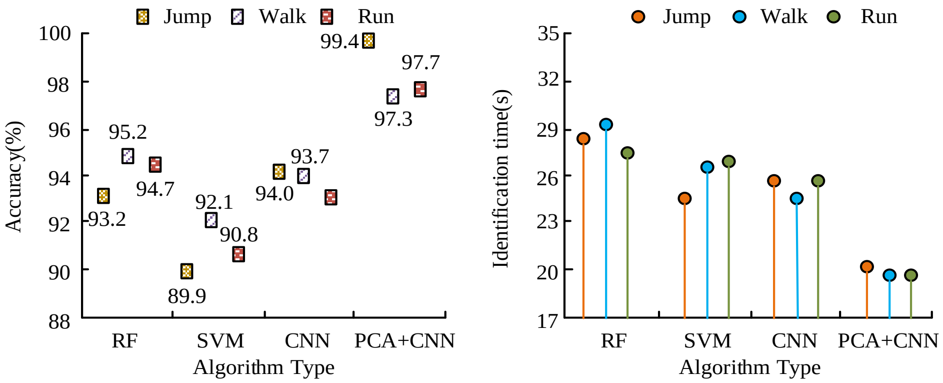
Fig. 9(a) and Fig. 9(b) represent the accuracy and time outcomes of the four methods for lower limb motion pose recognition, respectively, and the horizontal coordinates represent the four algorithms compared. From Fig. 9(a), the accuracy of the proposed PCA+CNN among the four methods is as high as 99.4% in the recognition of jumping, and the accuracy of 97.7% and 97.3% for running and walking respectively, which is higher than the other three methods. From Fig. 9(b), the recognition time of the Random Forest algorithm was above 28s, which was the worst performance among the four methods. In contrast, the PCA+CNN algorithm was stable at around 20s, which was more efficient than the other three methods and had better recognition performance. To sum up, the method proposed in the study can identify the basketball posture with high accuracy and maintain high recognition efficiency, which can provide more scientific and effective methods for basketball training.



(a) Accuracy of upper limb basketball posture recognition

(b) Recognition time of upper limb basketball posture

Fig. 8. Recognition outcomes of upper limb basketball movement posture using four methods.



(a) Accuracy of the legs basketball posture recognition

(b) Recognition time of the legs basketball posture

Fig. 9. Four methods of lower limb basketball posture recognition.

V. Conclusion

Traditional basketball training is based on the coach's personal training experience and theory, making it difficult to evaluate the training effect objectively. With the continuous improvement of deep learning algorithms, neural networks are increasingly widely used in data reduction and classification processing. The study proposes a scientific sports stance classification structure for basketball's complex and variable stance characteristics and thus establishes a data information collection module based on inertial sensors. The convolutional neural network is then improved using principal component analysis, and finally, the improved algorithm is applied to the recognition of basketball postures. Experimental outcomes demonstrate that the proposed PCA+CNN algorithm has an average recognition error rate of 3.15% in the Human3.6M dataset. Compared to 4.68% of the traditional CNN, the difference is 1.53%, and the error convergence curve of PCA+CNN is consistently below the CNN outcomes. In the MPII Human Pose dataset, the PCA+CNN converged in only the 5th round achieved 100% accuracy with better performance. In recognition of the upper limb pose for basketball, the method performed above 97% accuracy for all four types of dribbling, passing, catching, and shooting. Its running time is 10s faster and the accuracy for the lower limb motion pose is 99.4%. It has higher running efficiency and accuracy. However, the performance of CNN algorithm in the case of increased learning rate has not been analyzed in the study, and the power consumption caused by wearable devices has not been optimized. Therefore, it is necessary to further reduce the power consumption and increase the learning rate to achieve better results.

References

[1] Zhao B, Liu S. Basketball shooting technology based on acceleration sensor fusion motion capture technology[J]. EURASIP Journal on Advances in Signal Processing, 2021, 2021(1): 1-14.

[2] Pfitscher M, Welfer D, Do Nascimento E J, Cuadros M A D S L & Gamarra D F T. Article users activity gesture recognition on kinect sensor using convolutional neural networks and fastdtw for controlling movements of a mobile robot.Inteligencia Artificial, 2019, 22(63): 121-134.

[3] Li B, Xu X. Application of artificial intelligence in basketball sport.Journal of Education, Health and Sport, 2021, 11(7): 54-67.

[4] Hu B, Wang J. Deep learning based hand gesture recognition and UAV flight controls.International Journal of Automation and Computing, 2020, 17(1): 17-29.

[5] Azad R, Asadi-Aghbolaghi M, Kasaei S & Escalera S.Dynamic 3D hand gesture recognition by learning weighted depth motion maps.IEEE Transactions on Circuits and Systems for Video Technology, 2018, 29(6): 1729-1740.

[6] Sun Y,Fei T,Li X,Warnecke A,Warsitz E & Pohl N.Real-time radar-based gesture detection and recognition built in an edge-computing platform.IEEE Sensors Journal, 2020, 20(18): 10706-10716.

[7] Zhang Y, Cao C, Cheng J, & Lu H.EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition.IEEE Transactions on Multimedia, 2018, 20(5): 1038-1050.

[8] Li J, Gu D. Research on basketball players' action recognition based on interactive system and machine learning.Journal of Intelligent & Fuzzy Systems, 2021, 40(2): 2029-2039.

[9] Pan T Y, Chang C Y, Tsai W L & Hu M C. Multisensor-based 3D gesture recognition for a decision-making training system.IEEE Sensors Journal, 2020, 21(1): 706-716.

[10] Zhao Y, Liang J, Cui Y, Sha X & Li W J.Adaptive 3D position estimation of pedestrians by wearing one ankle sensor.IEEE Sensors Journal, 2020, 20(19): 11642-11651.

[11] Neethu P S, Suguna R, Sathish D. An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks.Soft Computing, 2020, 24(20): 15239-15248.

[12] Zhang W, Wang J, Lan F. Dynamic hand gesture recognition based on short-term sampling neural networks.IEEE/CAA Journal of Automatica Sinica, 2020, 8(1): 110-120.

[13] Lian C, Ma R, Wang X,Zhao Y, Peng H, Yang T & Li W J. ANN-enhanced IoT wristband for recognition of player identity and shot types based on basketball shooting motion analysis. IEEE Sensors Journal, 2021, 22(2): 1404-1413.

[14] Kamel A, Sheng B, Yang P.Deep convolutional neural networks for human action recognition using depth maps and postures.IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 49(9): 1806-1819.

[15] Adithya V, Rajesh R. A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition.Procedia Computer Science, 2020, 171(12):2353-2361.

[16] Han C. RETRACTED ARTICLE: Urban air pollution resolution and basketball training optimization based on time convolution network.Arabian Journal of Geosciences, 2021, 14(18): 1-7.

[17] Rast F M, Labruyère R. Systematic review on the application of wearable inertial sensors to quantify everyday life motor activity in people with mobility impairments.Journal of NeuroEngineering and Rehabilitation, 2020, 17(1): 1-19.

[18] Mangiarotti M, Ferrise F, Graziosi S, TAMBURRINO F & Bordegoni M. A wearable device to detect in real-time bimanual gestures of basketball players during training sessions.JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING, 2019, 19(1): 1-13.

[19] Murao K,Yamada H,Terada T & Tsukamoto M.Estimating Timing of Specific Motion in a Gesture Movement with a Wearable Sensor.Sensors and Materials, 2021, 33(1): 109-126.

[20] Pai N S, Chen Y H, Hung C P, Chen P Y, Kuo Y C & Chen J Y.Dual-input Control Interface for Deep Neural Network Based on Image/Speech Recognition.Sensors and materials, 2019, 31(11):3451-3463.

[21] Jain S, Rustagi A, Saurav S, Saini, R & Singh S.Three-dimensional CNN-inspired deep learning architecture for Yoga pose recognition in the real-world environment.Neural Computing and Applications, 2021, 33(12): 6427-6441.

[22] Hu Y, Lu M, Lu X. Driving behaviour recognition from still images by using multi-stream fusion CNN. Machine Vision and Applications, 2019, 30(5): 851-865.

Dynamic Hardware Redundancy Approaches Towards Improving Service Availability in Fog Computing

Sara Alraddady[1], Ben Soh[2], Mohammed AlZain[3], Alice Li[4]

Department of Computer Science and Information Technology-School of Computing, Engineering and Mathematical Sciences, La Trobe University, Australia[1, 2]

Department of Information Technology-College of Computers and Information Technology, Taif University, Kingdom of Saudi Arabia[3]

Department of Management & Marketing-La Trobe Business School, La Trobe University, Australia[4]

***Abstract*—The distributed nature of fog computing is designed to alleviate bottleneck traffic congestion which happens when a massive number of devices try to connect to more powerful computing resources simultaneously. Fog computing focuses on bringing data processing geographically closer to data source utilizing existing computing resources such as routers and switches. This heterogeneity nature of fog computing is an important feature and a challenge at the same time. To enhance fog computing availability with such nature, several studies have been conducted using different methods such as placement policies and scheduling algorithms. This paper proposes a fog computing model that includes an extra layer of duplex management system. This layer is designated for operating fog managers and warm spares to ensure higher availability for such a geographically disseminated paradigm. A Markov chain is utilized to calculate the probabilities of each possible state in the proposed model along with availability analysis. By utilizing the standby system, we were able to increase the availability to 93%.**

Keywords—Fog computing; fault tolerance; Markov chain; hardware redundancy

I. INTRODUCTION

The main idea that led researchers to introduce fog computing was to increase systems availability and delivered quality of service to overcoming cloud computing limitations. Fog computing focuses on distributing data processing instead of relying on centralized computing resources which is cloud data centers. It was introduced in 2012 by Cisco and defined as "an extremely virtualized environment that delivers networking, storage, and compute resources between outdated cloud computing information centers, usually, but not entirely situated at the network edge [1]." The reference structure of fog computing was introduced later in 2017. The structure consists of three layers. Highest layer is for cloud data centers, lowest layer is where end users reside, and between these two layers, a layer designed for fog node devices [2]. Fog nodes are computing devices with limited capabilities compared to data centers. These nodes can be routers, switches, access points, vehicles, or personal computers. On the other hand, end users can be mobile devices, sensors, actuators, or vehicles. The name fog reflects that fog in weather is closer to the ground than clouds. Fog computing brings the computation process closer to the end user by leveraging all available computing resources at the periphery of the network. The decentralised nature of fog computing reduces the amount of data that needs to be sent to the cloud. Thus, higher quality of service (QoS) can be achieved. Given the nature of fog computing which includes heterogeneity and end users mobility, numerous studies were conducted by researchers to explore the potential of this new computing paradigm aiming to support users mobility and design context aware fog computing paradigms. However, fog systems availability did not receive much attention in cases of faults occurrence. fog computing systems availability must be properly addressed to mitigate service disruption impact since such interruption can be very financially expensive and, in some cases, can lead to fatalities.

Availability of fog computing paradigm is concerned with ensuring that the system is reachable to end users as much as possible. In such highly heterogeneous paradigm, system availability is divided into two main parts. The first part is related to the availability of cloud layer, which is the sustainability of cloud data centers and their networking. This part has been investigated by scholars over the years as surveyed in [3], [4], and [5]. Different mechanisms were incorporated to improve the availability of cloud computing such as hot migration, load balancing, resource management, and traffic management. The second part is concerned with fog nodes and their communication. Researchers have done some work related to enhance the overall performance of fog computing by resource management mechanisms, load balancing algorithms, designing mobility, energy, and context aware fog environment, reviewing different fog architectures, and only a few incorporated fault tolerance techniques into fog computing.

Fault tolerance FT concept is defined as systems survival attribute in which it can operate with the existence of faults in any part of it. FT is a mechanism used to achieve high availability, scalability, resilience, and reliability and can be found in many fields including aviation, military, telecommunication, and space missions. Faults can be results of several factors - some are external factors and other internal ones like incorrect algorithms. Various Fault Tolerance Techniques FTT have been introduced to minimize faults manifestation in computing systems such as checkpointing, watchdog, and redundancy [6]. Fig. 1 summarizes fault tolerance techniques which include software and hardware techniques.

Aiming to increase fog computing availability, two fault tolerance techniques FTT are used. First, a management layer as a software FTT. This layer should be able to function independently from main cloud where contacting main cloud in the proposed work should be minimal. Main cloud communication is allowed to perform certain tasks such as long-term storage, history analysis, or complex computation. The efficiency of this layer is discussed in detail in [7]. Second, a standby system deployed at management layer as a hardware FTT since the management layer is considered as a backbone in the proposed model.

The remaining of the paper is structured as follows: Section II discusses related studies. Section III briefly, describes the software FTT in the proposed model followed by a duplex management system as a hardware FTT. Section IV provides a quantitative analysis for the proposed model using an example followed by an availability analysis in Section V. The comparison of proposed model with other models is given in Section VI and Section VII concludes the paper and highlights future work.

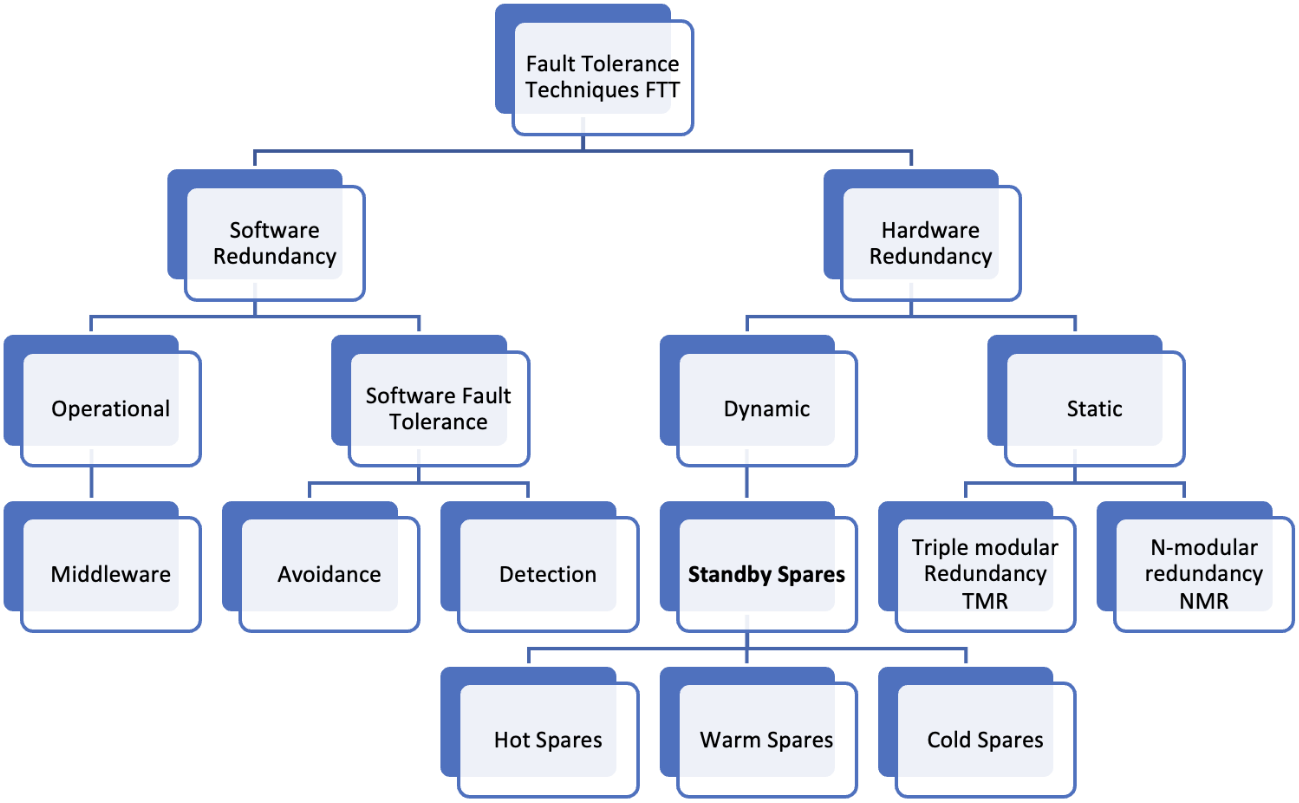


Fig. 1. Fault tolerance techniques FTT.

II. Literature Review

Fog computing is becoming popular since it provides a computing experience with low latency at low cost, and it is applicable to be deployed in various domains like healthcare, industries, and smart cities. To reach its potentials, researchers have been exploring several aspects of this new computing paradigm including: first, placement policies as in [8], [9], and [10]. Placement polices are responsible for deciding the most suitable computing node to process each single request. These policies can be context aware, energy aware or mobility aware. Second, several studies have been conducted on fog computing architecture as in [11], [12], [13], [14], and [15]. Some researchers increased the number of layers to reach six layers aiming to enhance the performance and robustness of fog computing. Third, scholars have been investing in fog resources management. Because of the heterogeneous nature of fog computing, balancing resources utilization and achieving the required quality of service must be addressed. Hence, authors in [16] propose load balancing algorithm to maximize resource utilization. However, searching for studies that focus on fault tolerance in fog computing, showed that service replication was the only FTT investigated by researchers. The main idea of service replication is to create replicas of data at the edge of the hierarchy. These replicas can be used in cases of fault occurrence. Authors in [17], propose a multi-tiers fog model, in which data replicas are generated at the same tier. The proposed model includes mobile agents that can roam all tiers to investigate failures when occurred, facilities communication between devices in different tiers and most importantly, fetch jobs with high priorities to be assigned. Simulating this model showed better performance and fault tolerance. Additionally, Javed et al. in [18], propose a fault tolerant architecture for edge applications which can save data when connection to cloud is lost. This architecture consists of three layers which are: application isolation, data transport, and management layer.

They tested the performance of the proposed architecture on surveillance cameras and found that the model was able to tolerate losing two nodes out of five. Furthermore, Authors in [19] developed a service replication scheme that can sense what services need to be replicated and chooses the most suitable node to perform and store the replica.

After reviewing all the listed studies, we found out that hardware redundancy did not receive much attention. One of the reasons behind this goes to the fact that hardware redundancy involves extra cost. Also, incorporating hardware redundancy increase the level of complexity of any system. However, integrating extra hardware ensures higher availability and higher fault tolerance level because faults are inevitable whether internal or external ones. Accordingly, the idea of the proposed model in this paper arose, which addresses heterogeneity and resource management in fog computing and improve system availability using hardware FTT as the next section illustrates.

III. Proposed Model Design

Fog computing architecture mainly consists of three layers as mentioned before. In this paper, one more layer is added to the hierarchy that resides between fog node layer and cloud layer. The purpose of this layer is to governs fog nodes and their communications among each other and with cloud data centres. It consists of fog nodes that are called fog managers. Each fog manager has three modules: processing module, tracking module, and allocating module. It also, has a sending and receiving units for communication purposes as depicted in Fig. 2 [7].

Since this management layer is considered the backbone of the proposed hierarchy, any fault that occurs at this layer will lead to expensive consequences. Therefore, a standby system for this layer is designed. Generally, standby systems were designed for extremely high availability systems such as aircraft where service disruptions are not acceptable. Standby systems consist of three components: active node, standby node, and a switching unit as illustrated in Fig. 3. Techniques to implement a standby system fall into three categories: hot, warm, and cold. Tradeoffs between cost, energy consumption, and availability rate must be considered during design stage. When switching time is critical as the case in aircrafts, hot standby is the most suitable technique. In this technique, a completely fired up node is always ready to take over the failing node. Energy consumption for this technique is very high, yet higher availability is guaranteed. On the other hand, cold standby requires higher switchover time, consumes less energy since the standby node is not fully functioning and in sometimes is not fired up, and provides less availability than hot standby. It is suitable for industrial use and weapon systems. Lastly, warm standby which falls between hot and cold standby technique [20]. In this paper, a standby system is designed at management layer to increase availability in fog computing environment. To the best of our knowledge, designing a highly available fog computing environment using this fault tolerance technique was not investigated yet.

Accordingly, at management layer in the proposed model, a group of fog nodes reside in charge of managing fog nodes connected to it. Each cluster of fog nodes in a relatively small area is managed by a single fog node. In this proposed model, a number of standby nodes is added to optimize the availability as described in Fig. 4.

The proposed model is an active/passive standby system which denotes that only one node is in control. Passive/standby node steps into failover the active node. The switching technique between nodes is the common switching policy as illustrated in Fig. 5.

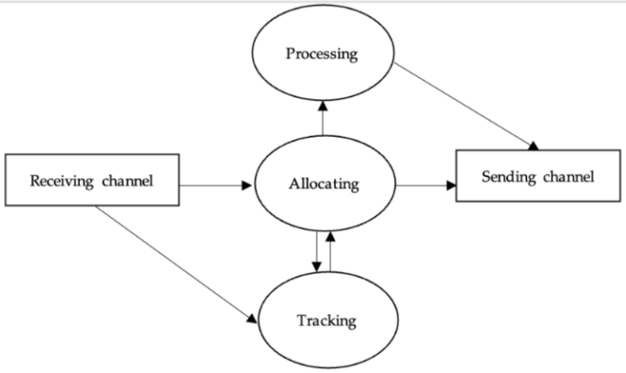


Fig. 2. Fog manager components.

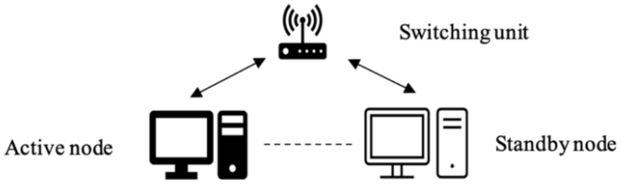


Fig. 3. Standby systems components.

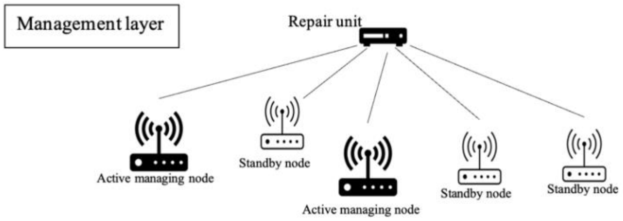


Fig. 4. Proposed standby system at management layer.

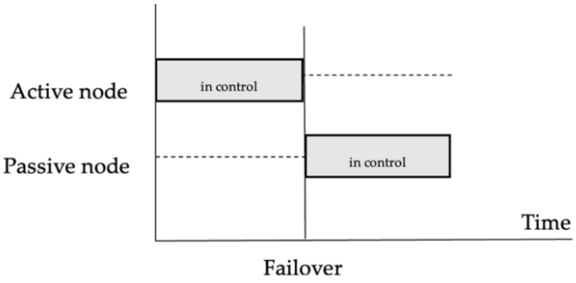


Fig. 5. Common switching policy for standby systems.

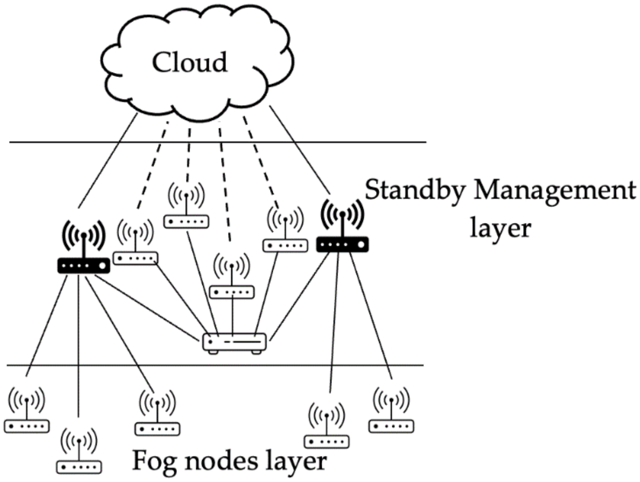


Fig. 6. Network connectivity in the proposed model.

Both the active/operating manager O_m and warm manager W_m are identical hardware components but differ in the failure rate. The failure rate of O_m is λ_o while $\lambda_o > 0$, and the W_m failure rate is λ_w where $0 < \lambda_w < \lambda_o$. Each standby manager has a unit with a Boolean value either up or down. When the unit value is up (one) the manager can be in an active mode or idle mode. A down or (zero) value indicates that the manager is not operating. Network connectivity assumptions for the proposed standby system are depicted in Fig. 6 as follows:

- Both active managers and standby mangers are securely connected the repair unit.
- Active managers are securely connected to cloud data centers.
- Standby managers have an established connection to cloud centers which will activated once the standby manager status switched to active.
- Active fog managers are connected to fog nodes at the next lower level of the models.
- Fog managers can establish secure connections among each other for higher resource utilization.

To illustrate and evaluate the switching mechanism between active and standby managers at the management layer, a Colored Petri net is designed. Petri nets PN are used to evaluate and analyze embedded systems performance, and they consist of places, transitions, and arcs. Places, which are in a circle, represent the states of a system; transitions, symbolized as rectangles, exist between places; and arcs demonstrate the workflow [21]. Usually, some places contain tokens that demonstrates the dynamics of the system. Even though Petri nets were invented at an earlier time, it is still used for its proficiency in designing sophisticated distributed computing systems as in [22]. Fig. 7 describes how the proposed standby system works in case an active manager fails. The red places represent failed managers and blue places signifies the completion of a process. Additionally, green transitions indicate the start of the moving process.

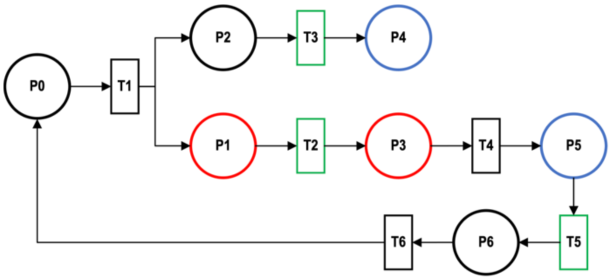


Fig. 7. PN design of the proposed standby system.

<u>PN notations_1:</u> There are six states for the proposed system components as follow:

P0: initial state

P1: failed fog manager moving to repair unit

P2: setting W_m in full operating mode with failure rate of λ_o instead of λ_w

P3: failed managers are moving to repair unit

P4: fog manager is active with a failure rate of λ_o

P5: failed managers are being fixed and restored as new components

P6: the manager is ready to use as a warm standby manager

<u>PN notations_2:</u>

T1: is enabled when an active manager has failed

T3: start moving the activated manager into fully operating manager

T2: start moving the failed manager to the repair unit

T4: end moving the failed manager to the repair unit

T5: enabled when the failed manager has been fixed

T6: start using fixed manager as a warm standby manager with failure rate of λ_w which takes the system to its initial state.

The initial state of the proposed model is represented by P0 where the active manager is in a fully functioning state with a failure rate of λ_o and the standby manager in warm state and with a failure rate of λ_w. When the active manager fails, which is represented by T1, the system performs two steps. First, it moves the failed manager to the repair unit (P1) and sends an order to the standby manager to be fully operating (P2). P3 represents the system state during moving the failed manager to the repair unit, and in P4, standby manager is now fully active with failure rate of λ_o and referred to as the active manager. States P3 and P4 happen at the same time as well as P1 and P2. Next, P5 represents the system state when the failed manager is being fixed or replaced depending on its condition. When P5 is completed, T5 is enabled. T5 transits the system to P6, in which the failed manager is fixed and ready to be active again. Subsequently, T6 is activated by P6 which takes the system back to its initial state.

IV. Qualitative Analysis of the Proposed Model

The proposed model consists of cloud data centers, operating fog manager O_m, warm fog managers W_m, fog manager, and one repair unit. In the next lower layer, fog nodes reside to provide the required services to end users. Operating fog managers manage the fog nodes with a failure rate of λo, and warm fog managers are in standby mode with a failure rate of λw. There are two chains of circles representing all possible states as illustrated in Fig. 8. Each circle has a number inside of it representing the number of failed managers in that state. The values on the arcs flowing in and out is associated with the status of the devices (failed/fixed) that changes the system state. Two chains of states coded as 1 and 2 to represent the possible state transitions for operating managers O_m and warm managers W_m where μ is the repair rate of the repair unit. Chain 1 represents the system states with 0, 1, …, O_m failed operating managers, and chain 2 represents the system states with failed warm managers starting from O_m+1 until $O_m + W_m$, which represents the total number of managers T.

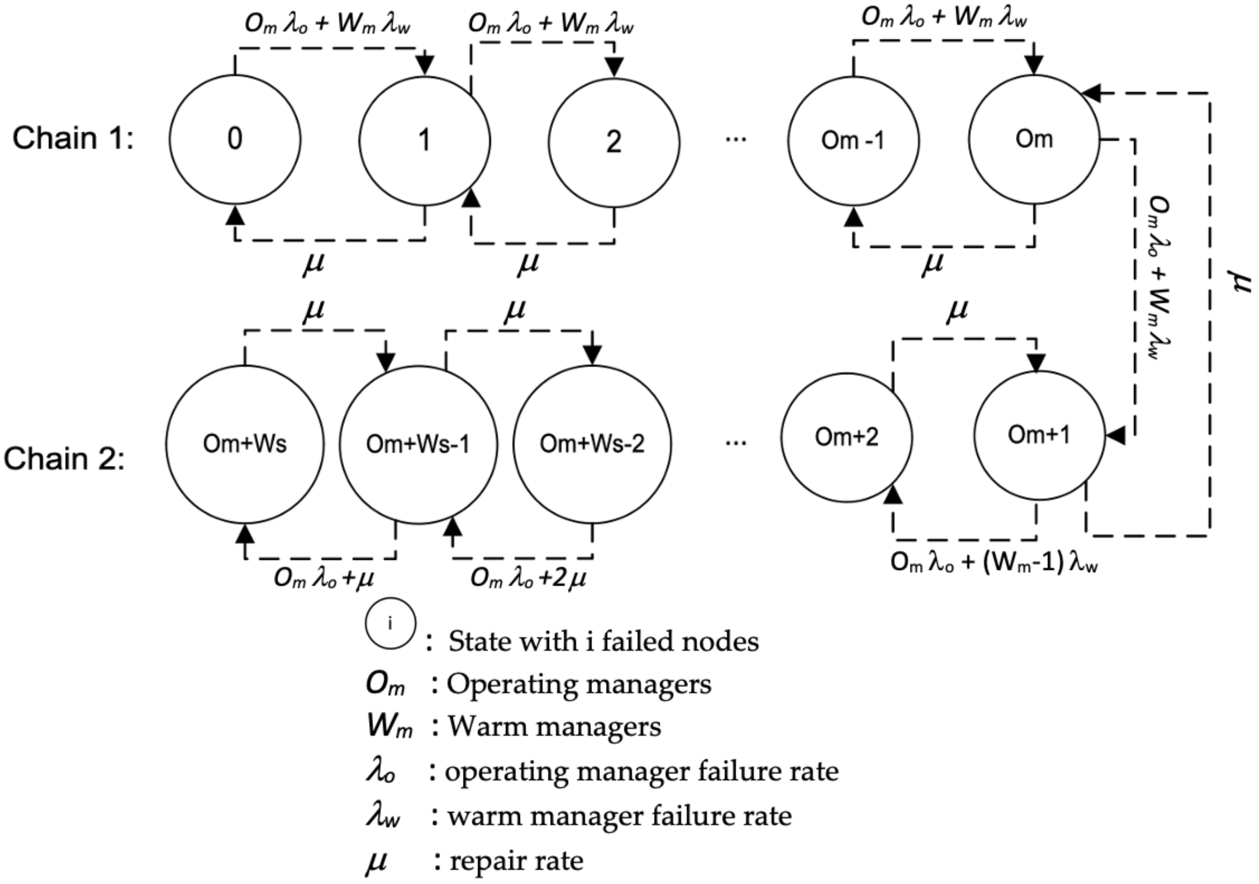

Chain 1:
0
1
2
···
Om -1
Om
$O_m \lambda_o + W_m \lambda_w$
$O_m \lambda_o + W_m \lambda_w$
$O_m \lambda_o + W_m \lambda_w$
μ
μ
μ
$O_m \lambda_o + W_m \lambda_w$
μ
Chain 2:
Om+Ws
Om+Ws-1
Om+Ws-2
···
Om+2
Om+1
μ
μ
μ
$O_m \lambda_o + \mu$
$O_m \lambda_o + 2\mu$
$O_m \lambda_o + (W_m\text{-}1) \lambda_w$
i : State with i failed nodes
O_m : Operating managers
W_m : Warm managers
λ_o : operating manager failure rate
λ_w : warm manager failure rate
μ : repair rate

Fig. 8. Markov chain of the proposed model.

Let T represent the total number of fog managers, and the states i, i$\in$ [0, T], represents the number of failed devices in a state. In the initial state P(0), there is a state with zero failed devices, and P(i) represents the probability of i failed devices. The states probability can be derived from the following equation:

Chain (1): $0 \le i \le O_m$

i = 0 (initial state): $O_m \lambda_o + W_m \lambda_w P(0) = \mu P(1)$

i = 1: $(O_m \lambda_o + W_m \lambda_w + \mu) P(1) = (O_m \lambda_o + W_m \lambda_w) P(0) + \mu P(2)$

i = 2: $(O_m \lambda_o + W_m \lambda_w + \mu) P(2) = (O_m \lambda_o + W_m \lambda_w) P(1) + \mu P(3)$

i = O_m -1 : $(O_m \lambda_o + W_m \lambda_w + \mu) P(O_m -1) = (O_m \lambda_o + W_m \lambda_w) P(O_m -2) + \mu P(O_m)$

i = O_m : $(O_m \lambda_o + W_m \lambda_w + \mu) P(O_m) = (O_m \lambda_o + W_m \lambda_w) P(O_m -1) + \mu P(O_m +1)$

A general expression for P(i) in chain 1 can be calculated by:

$$(O_m \lambda_o + W_m \lambda_w + \mu) P(i) = (O_m \lambda_o + W_m \lambda_w) P(i-1) + \mu P(i+1) \quad (1)$$

where i $\in$ [1, O_m]

Chain (2): $(O_m + 1) \le i \le (O_m + W_m)$

i = O_m + 1 : $[O_m \lambda_o + (W_m\text{-}1) \lambda_w + \mu] P(O_m + 1) = [O_m \lambda_o + (W_m\text{-}1) \lambda_w] P(O_m) + \mu P(O_m +2)$

i = O_m + 2 : $[O_m \lambda_o + (W_m\text{-}2) \lambda_w + \mu] P(O_m + 2) = [O_m \lambda_o + (W_m\text{-}2) \lambda_w] P(O_m+1) + \mu P(O_m +3)$

i = $(O_m + W_m - 1)$: $[O_m \lambda_o + \lambda_w + \mu] P(O_m + W_m - 1) =$

$$[O_m \lambda_o + 2\lambda_w] P(O_m + W_m - 2) + \mu P(O_m + W_m)$$

$$i = O_m + W_m : [O_m \lambda_o + \mu] P(O_m + W_m) = [O_m \lambda_o + \mu] P(O_m + W_m)$$

A general expression for P(i) in chain 1 can be calculated by:

$$i = O_m + W_m : [O_m \lambda_o + \mu] P(O_m + W_m) = [O_m \lambda_o + \mu] P(O_m W_m - 1) + \mu P(O_m + W_m + 1) \quad (2)$$

$$[O_m \lambda_o + (O_m + W_m - i) \lambda_w + \mu] P(i) = [O_m \lambda_o + ((O_m - W_m - i) +1) \lambda_w P(i-1) + \mu P(i+1)$$

$$\text{where } i \in [(O_m + 1) , (O_m + W_m)] \quad (3)$$

Given the fact that $O_m + W_m$ is equal to T and based on equation (2), the final state of the system is given by:

$$i = T \text{ (final state)} : [O_m \lambda_o + \mu] P(T) = [O_m \lambda_o + \mu] P(T - 1) + \mu P(T+1) \quad (4)$$

Consequently, the utilization of operating managers and warm managers can be denoted as ρ_o and ρ_w respectively. The utilization parameter can be calculated by dividing the failure rate by the repair rate. By solving Eqs. (1) to (4) P(i) can be calculated as follows:

$$\text{Chain 1} : P(i) (Om + Wm) i \; P(0) , 0 \leq i \leq Om) \quad (5)$$

$$\text{Chain 2} : P(i) (Om + Wm) om \; x \prod_{j=om}^{i-1} [Om \rho o + (T - j) \rho w] P(0) \; , Om \leq i \leq T \quad (6)$$

V. Availability Analysis for the Proposed Model

Since the proposed model is designed to enhance availability in fog computing, it is crucial to construct related metrics such as expected number of failed operating and warm fog managers, which is the focus of this paper. Generally, systems availability can be defined as the system probability of being in a functioning state at any time. Fog computing availability includes two parts: the availability of cloud data centers and the availability of fog nodes. Cloud computing availability has been receiving a decent amount of attention from scholars compared to fog computing. The following characteristics has been defined for the proposed model:

L = represents the expected number of failed managers

E[O] = the expected number of failed operating managers O_m

E[W] = the expected number of warm managers W_m

A = the availability of standby management layer

From Eq. (5) to (6), the following expressions can be driven:

$$L = \sum_{i=0}^{T} i \times P(i) \quad (7)$$

$$E[O] = \sum_{i=1}^{Om} (T - i) \times P(i) \quad (8)$$

$$E[W] = \sum_{i=(Om+1)}^{T} (T - i) \times P(i) \quad (9)$$

$$A = \frac{T - L}{T} = 1 - \frac{L}{T} \quad (10)$$

In order to illustrate the theoretical implication of the proposed model, an example with comprehensive calculation is provided. In this example, there are two operating managers with failure rate λ_o of 0.1, three warm managers with failure rate λ_w of 0.024, and one repair unit with repair rate μ of 0.8. Accordingly, ρ_o and ρ_w are equal to 0.125 and 0.03 respectively.

Based on Eq. (5) and the given values of:

$$O_m = 2, \rho_o = 0.125, W_m = 3, \rho_w = 0.03, \text{ and } \mu = 0.8$$

chain 1 yields to:

$$\text{Chain 1: } P(i) = (O_m \rho_o + W_m \rho_w)^i P(0), 0 \leq i \leq O_m \quad (5)$$

$$i = 1, P(1) = (O_m \rho_o + W_m \rho_w) P(0) = (0.25 + 0.09) P(0) = 0.34 P(0)$$

$$i = 2, P(2) = (O_m \rho_o + W_m \rho_w)^2 P(0) = (0.25 + 0.09)^2 = (0.34)^2 P(0) = 0.1P(0)$$

And chain 2 based on Eqs. 6 leads to

$$\text{Chain 2} : P(i) (O_m + W_m)^{om} x \prod_{j=om}^{i-1} [Om \rho o + (T - j) \rho w] P(0) \; , O_m \leq i \leq T \quad (6)$$

$$i = 3, P(3) = (0.34)^2 x \prod_2^2 [0.25 + (5 - j) 0.03] P(0)$$

$$\Rightarrow (0.34)^2 x [0.25 + (5-2) 0.03]$$

$$P(3) = 0.1 \times 0.34 P(0) = 0.034 P(0)$$

$$i = 4, P(4) = (0.34)^2 \times \prod_2^3 [0.25 + (5 - j) 0.03] P(0)$$

$$\Rightarrow (0.34)^2 x \{ [0.25 + (5-2) 0.03] x [0.25 + (5-3) 0.03]\}$$

$$P(4) = 0.1 \times 0.1 P(0) = 0.01P(0)$$

$$i = 5, P(5) = (0.7)^2 \times \prod_2^4 [0.25 + (5 - j) 0.03] P(0)$$

$$\Rightarrow (0.34)^2 x \{ [0.25 + (5-2) 0.03] x [0.25 + (5-3) 0.03] x [0.25 + (5-4) 0.03\}$$

$$P(5) = 0.1 \times 0.03 P(0) = 0.003 P(0)$$

The five state probabilities expressed in terms of P(0) are as follows:

$$[P(1), P(2), P(3), P(4), P(5)] P(0) = [0.34, 0.1, 0.034, 0.01 \; 0.003] P(0) \Rightarrow 0.49 P(0)$$

To calculate the value of P(0), the following normalization condition is used,

$$\sum_{i=0}^{T} P(i) = 1$$

$$\sum_{i=0}^{5} P(0) + P(1) + P(2) + P(3) + P(4) + P(5) = 1$$

$$\Rightarrow 0 .51 + P(0) = 1 \Rightarrow \therefore P(0) = 0.51$$

All the values above were calculated with respect to P(0). After calculating P(0), P(i) values are as follows:

$$P(1) = 0.17,\ P(2) = 0.05,\ P(3) = 0.017,\ P(4) = 0.005,\ \text{and}\ P(5) = 0.0015$$

Accordingly, solving Eq. (7) to (10) to calculate the proposed model metrics as follows:

$$L = \sum_{i=0}^{5} i \times P(i) = [P(1) + 2\ (P2) + 3\ (P3) + 4\ P(4) + 5\ P(5)] \Rightarrow$$

$$= (0.34 + 0.2 + 0.1 + + 0.04 + 0.015) = 0.7 \times 0.51 = 0.357$$

$$E[O] = \sum_{i=1}^{2}(5 - i) \times P(i) = 4\ P(1) + 3\ P(2) \Rightarrow (1.36 + 0.3) \times 0.51 \quad = 0.85$$

$$E[W] = \sum_{i=3}^{5}(5 - i) \times P(i) = 2\ P(3) + P(4) \Rightarrow (0.7 + 0.01) \times 0.51 \quad = 0.36$$

$$A = 1 - \frac{L}{T} = 1 - \frac{0.36}{5} \Rightarrow 93\%$$

Fig. 9 illustrates the probabilities of a fog management system of two operating managers and three warm managers with failure rates of 0.1 and 0.024 respectively and a repair rate of 0.8. The vertical axis represents the steady state probability, and the horizontal axis denotes the number of failed managers. The curve starts with the probability of losing one operating manager with a probability of 0.17. As the number of failed managers increases, the failure probability decreases until it reaches 0.0015. With the mentioned failure and repair rates, the model was able to reach 93% availability and the relevant measures such as E[O] and E[W] are 0.7 and 1.5 independently.

Additionally, Fig. 10 depicts the improvement in availability percentage of the proposed model using redundancy. The figure compares availability percentages of two models. First, the redundant model consists of two operating mangers, three warm mangers, and a repair unit. Second, the non-redundant model consists of only two operating managers. Failure rates are identical in both models. As the figure shows, the redundant model availability is 93% while the non-redundant model availability is 87%. Accordingly, the redundant model increases availability by 6% compared to a non-redundant model.

Further experiments were conducted to increase the proposed model availability over 98%. These experiments included changing the failure and repair rates. However, increasing the repair rate and decreasing failure rate can be very expensive. A different experiment was conducted which focused on the ratio of operating managers to warm managers.

As Fig. 11 represents, availability percentage increases as the number of warm managers increase. The figure depicts different configurations with correlations of operating managers O_m to warm managers W_m as follows:

A: $W_m = O_m$

(The number of W_m equals to the number of O_m e.g., O_m = 2 and W_m = 2).

B: $W_m = O_m^{\,2}$

(The number of W_m equals to the number of O_m to the power of 2 e.g., O_m = 2 and W_m = 4).

C: $W_m = O_m^{\,3}$

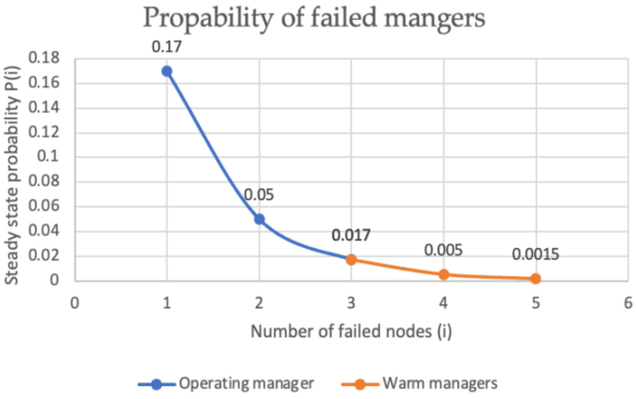


Fig. 9. Probabilities of failed managers in fog management system.

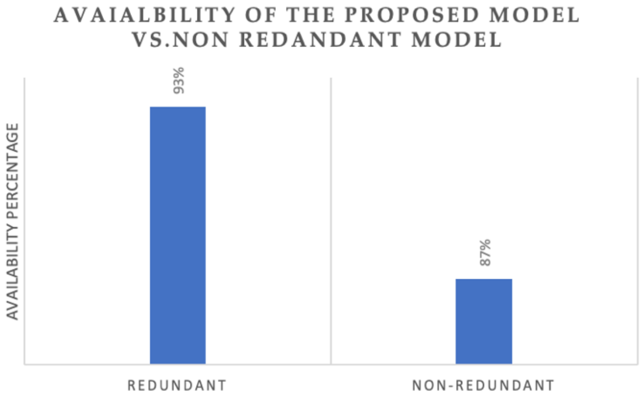


Fig. 10. Availability of the proposed model compared to a non-redundant model.

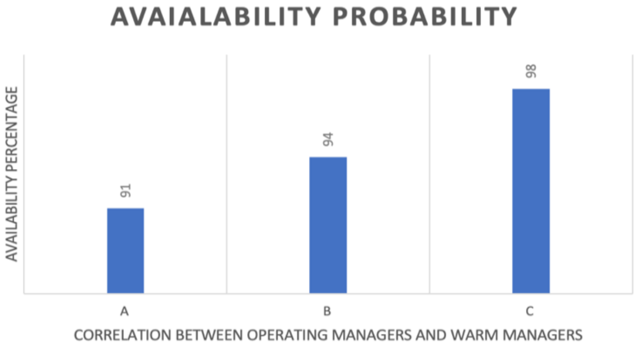


Fig. 11. Availability probability for different numbers of operating/warm nodes.

(The number of W_m equals to the number of O_m to the power of 3 e.g., O_m = 2 and W_m = 8).

Failure and repair rates are fixed for all models. As the figure shows, an availability rate of 98% can be reached when using configuration C, which is a high rate and defiantly it comes with expenses. In the case of configuration C, the extra cost is in the form of extra hardware which is more practical than unreasonable repair and failure rates.

VI. Comparison of the Proposed Model with other Models

The proposed model in this paper is a standby system consisting of operating nodes and warm nodes. The system is designed to reside in a separate layer between fog nodes and cloud data centers, which is called the management layer. This layer is discussed in detail in [7]. The mechanism used in this model is hardware redundancy. As mentioned in Section II, the main fault tolerance technique that have been incorporated into fog computing is data/application replication as in [18] and [19]. Availability probability was not calculated in these studies. The research in [18], did not include any evaluation metrics for the proposed software, On the other hand, authors in [19] presented a proactive scheme for service replication in IoT computing. The evaluation process of the scheme included service drop rate, response time, and service completion time. Even though this study covered aspect of availability, it was designed for qusai adhoc scenarios in general which may include fog computing. Also, it did not include hardware redundancy to be suitable for comparison with the presented model in this paper. Further research can be conducted to explore other models or techniques to improve availability in fog computing.

VII. Conclusion and Future Works

To enhance performance and availability in fog computing, researchers have been focusing on several aspects of fog computing such as resource management, load balancing algorithms, placement policies, and service replication. However, deployment in fault tolerance techniques has not received much attention. In this paper, a fog model is proposed with a standby management layer. The model is designed to tolerate losing fog managers at the management layer. Qualitative analysis of the proposed model is presented using a Markov chain. For further studies, we aim to study the limitations of the proposed model and enable periodic switching for the duplex system to avoid exhausting fog nodes. Furthermore, a cost effectiveness study along with extensive comparison of the proposed model with other models designed to improve availability in fog computing can be conducted.

References

[1] R. M. J. Z. S. A. Flavio Bonomi, "Fog computing and its role in the internet of things," in The first edition of the MCC workshop on Mobile cloud computing, 2012. (pp. 13-16). New York, NY, United States: Association for Computing Machinery., 2012.

[2] O. C. A. Working, "OpenFog Reference Architecture for Fog Computing," February 2017. [Online]. Available: https://site.ieee.org/denver-com/files/2017/06/OpenFog_Reference_ Architecture_2_09_17-FINAL-1.pdf. [Accessed June 2022].

[3] T. Welsh and E. Benkhelifa, "On Resilience in Cloud Computing: A Survey of Techniques across the Cloud Domain," ACM Computing Surveys, vol. 35, no. 3, pp. 1-36, 2021.

[4] O. H. M. Heberth F. Martine, H. A. Rubio and J. Marquez, "Computational and Communication Infrastructure Challenges for Resilient Cloud Services," Computers , vol. 11, no. 8, 2022.

[5] W. Wang, H. Chen and X. Chen, "An Availability-Aware Approach to Resource Placement of Dynamic Scaling in Clouds," in IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, USA, 2012.

[6] I. Koren and C. Krishna, Fault-Tolerant Systems, 2007.

[7] B. S. M. A. A. L. Sara Alraddady, "Fog Computing: Strategies for Optimal Performance and Cost Effectiveness," Electronics, vol. 11, no. 21, 2022.

[8] P. Maiti, H. K. Apat, B. Sahoo and A. K. Turuk, "An effective approach of latency-aware fog smart gateways deployment for IoT services," Internet of Things, vol. 8, 2019.

[9] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana and M. Parashar, "Mobility-Aware Application Scheduling in Fog Computing," IEEE Cloud Computing , vol. 4, no. 2, pp. 26 - 35, 2017.

[10] I. Lera, C. Guerrero and C. Juiz, "Availability-Aware Service Placement Policy in Fog Computing Based on Graph Partitions," IEEE Internet of Things Journal , vol. 6, no. 2, pp. 3641 - 3651, 2018.

[11] M. Aazam and E.-N. Huh, "Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT," in 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, Korea (South), 2015.

[12] T. Zhang, J. Jin, X. Zheng and Y. Yang, "Rate adaptive fog service platform for heterogeneous iot applications," IEEE Internet of Things Journal , vol. 7, no. 1, pp. 176 - 188, 2020.

[13] V. Karagiannis and S. Schulte, "Comparison of Alternative Architectures in Fog Computing," in 2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC), Melbourne, VIC, Australia, 2020.

[14] M. Aldossary and H. A. Alharbi, " Towards a Green Approach for Minimizing Carbon Emissions in Fog-Cloud Architecture," IEEE Access, no. 9, pp. 131720 - 131732, 2021.

[15] L. Benchikh and L. Louail, "Task scheduling approaches for fog computing," in 2021 30th Wireless and Optical Communications Conference (WOCC), Taipei, Taiwan, 2021.

[16] A. J. Kadhim and J. I. Naser, "Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN," China Communications, vol. 18, no. 2, pp. 271 - 289, 2021.

[17] J. Grover and R. M. Garimella, "Reliable and Fault-Tolerant IoT-Edge Architecture," in IEEE SENSORS, New Delhi, India, 2018.

[18] A. U. E. F. Asad Javed Department of Computer Science, K. Heljanko, A. Buda and K. Främling, "CEFIoT: A fault-tolerant IoT architecture for edge and cloud," in IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 2018.

[19] B. Choudhury, S. Choudhury and A. Dutta, "A Proactive Context-Aware Service Replication Scheme for Adhoc IoT Scenarios," IEEE Transactions on Network and Service Management , vol. 16, no. 4, pp. 1797 - 1811, 2019.

[20] T. Zhang, M. Xie and M. Horigome, "Availability and reliability ofk-out-of-(MCN):G warm standby systems," Reliability Engineering and System Safety, vol. 91, no. 4, p. 381–387, 2006.

[21] M. Naedele and J. W. Janneck, "Design patterns in Petri net system modeling," in Proceedings. Fourth IEEE International Conference on Engineering of Complex Computer Systems, Monterey, CA, USA, 1998.

[22] N. S. Noori and T. I. Waag, "Application of Hierarchical Colored Petri Nets for Real-Time Condition Monitoring of Internal Blowout Prevention (IBOP) in Top Drive Assembly System," in IEEE International Systems Conference (SysCon), Orlando, FL, USA, 2019.

Eye Contact as a New Modality for Man-machine Interface

Syusuke Kobayashi[1], Pitoyo Hartono[2]
Graduate School of Engineering, Chukyo University, Nagoya, Japan[1]
School of Engineering, Chukyo University, Nagoya, Japan[2]

***Abstract*—In daily life, people use many appliances, where different machines and tools should be operated with their specialized interfaces. These specialized interfaces are often not intuitive and thus require considerable time and effort to master. On the other hand, human communications are rich in modalities and mostly intuitive. One of them is eye contact. This study proposes eye contact for enriching modalities for human-machine interface. The proposed interface modality, based on a neural network for object detection, allows humans to initiate machine operations by looking at them. In this paper, the hardware framework for building this interface is elaborated and the results of usability assessment through users' experiments are reported.**

Keywords—Eye contact; man-machine interface; non-verbal communication; object detection; neural network

I. Introduction

Machines and tools have been invented for centuries to support humans to efficiently work and to make life comfortable. However, most machines, for example, home appliances, require specialized interfaces that are not necessarily intuitive for humans. The situation is more serious in industrial settings where many types of machinery need to be operated under specialized rules. Sometimes it takes considerable time for humans to be proficient in the interface while misoperation may cause serious accidents. Therefore, it is desirable to develop interfaces that minimize human errors and allows users to instantly understand and be proficient in using them. The intuitiveness of an interface is influenced by various factors since the user's subjective senses are important. For example, in psychology, there is a concept of Organization of Memory [1, 2]. Since there is a limit to the capacity of human memory, it is important to represent information and skills in simple forms that can be easily remembered or executable by humans. One of the important aspects of the simplicity of the representation is familiarity. From these considerations, interfaces will be more intuitive if they are based on familiar experiences [3]. Hence, enclosing familiarity is a good strategy for building an intuitive human-machine interface.

While most machines need fixed and specialized interfaces, humans flexibly utilize various modalities for communicating with each other. In human interactions, verbal communication is the most frequently used interface. However, humans also utilize rich nonverbal modalities for communication, such as gestures, facial expressions, gaze, and eye contact.

Recently, interfaces based on voice and natural language recognition have been widely used in households. They enable humans to verbally interact with machines. While verbal interfaces are effective in household settings, they are not necessarily useful in other settings, such as factories, busy streets, public spaces, and public transportation. In such situations, humans complement verbal communication with nonverbal modalities [4] to seamlessly interact with each other. Hence, it is also beneficial for human-machine interfaces to complement verbal-based modality with nonverbal modalities, for example, eye contact.

In recent years, the advance in technology allows the proposal for many nonverbal interfaces between humans and machines. For example, gaze-based interactions between humans and computers have been developed to assist people who are unable to perform some physical movements due to spinal cord injury or other causes, but also for helping healthy users efficiently operate computers. For example, some studies have been conducted on the use of eye gaze for cursor manipulation in Graphical User Interface (GUI) [5, 6]. When operating a GUI using a mouse or touch screen, the user's gaze is directed to a button on the screen before making a selection. While it is natural and reasonable to use eye movement as a pointer, there is a so-called Midas Touch Problem [7], in which the system cannot determine the intention of a user, in that it is difficult to distinguish whether the user is looking at the screen or has an intention for clicking a button on the screen. Furthermore, it has been reported that clicking by staring or blinking has some latencies compared to clicking a mouse [8]. A system has also been developed to move a wheelchair in the direction of the user's gaze [9]. Here, the discomfort of having to look down the road when operating the wheelchair has been reported in the experiment in simply linking gaze input to a particular movement of a machine is not natural and not intuitive. Additionally, eye gaze is information that can be used to extract human unconscious interests and attention. One of the gaze-based systems [10] generates e-commerce recommendations based on gaze information. While conventional recommendation systems require past shopping characteristics of a user, to determine what to recommend, this system can estimate the user's preferences with high accuracy based on his/her gaze movements. In addition, there is also a study that detects drivers' distractions using gaze tracking as Advanced Drive Assistance System (ADAS) [11]. By dealing with gaze, a human can interact with machines intuitively, naturally, and efficiently. The intuitiveness of gaze information motivates this study.

This study attempts to propose a means for adding a new modality for nonverbal interaction between humans and machines. Here, the basic concept is to allow eye contact between humans and machines. Eye contact has four roles in human communication [12-14]. The one that is highly relevant to this study is the cognitive role of displaying attention to other people and conveying an intention for starting to communicate. In this study, eye contact is expanded to establish intuitive interactions between humans and machines.

There were existing interfaces that attempt to utilize eye contact. For example, the smart speaker "Tama" [15] can be activated using mutual gaze for starting an interaction. It is reported that the usability and the sense of dialogue improved. Other systems include the construction of an IoT system that combines eye gazing and gestures for human appliances [16]. It realized intuitive interaction between humans with home appliances through gaze and gesture via a so-called "Watch module". Our study shares some similarities with these past studies, in that they realize natural and intuitive interaction by using eyes [15-17]. However, the proposed study differs in that it offers direct interactions with the objects without requiring any other intermediate media and hence increases the naturalness and intuitiveness of the interaction. The proposed system also established a flexible relationship that was not limited to smart speakers and home appliances, but also any type or number of objects.

In the past, a basic framework was developed for this system [18]. This paper reports on the preliminary experiments' results on the performance of the proposed systems and the user's assessments. It is important to mention that it is not our intention to compete with the existing systems' efficiency. Here our objective is to investigate the usability of the proposed eye-contact system and to assess its potential for enriching the user interface modalities. In this paper, the proposed system's characteristics are assessed through statistical tests on users' experiment data.

The rest of the paper is organized as follows. Section II explains the hardware and software configurations of the proposed Gaze Switch. Section III explains the experiments, while the final section explains the conclusions and future work for this study.

II. Outline of Gaze Switch

Gaze Switch developed in this research is an interface that enables humans to activate or deactivate a machine by looking at it. This interaction is comparable to eye contact. Fig. 1 illustrates the process of establishing inter-human and human-machine through eye contact.

Eye contact between two humans starts when they gaze at each other and in the process, each party needs to perceive the gaze. In this study, for human machine-interaction, it is assumed that the machines are always gazing at humans, and so when a human visually perceives the machines, eye contact is established. Here, a neural network is utilized for determining the target object. Fig. 2 shows an overview of the Gaze Switch system developed in this research.

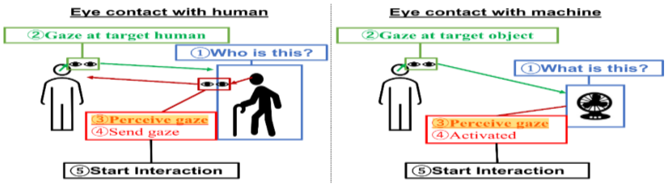


Fig. 1. Eye contact.

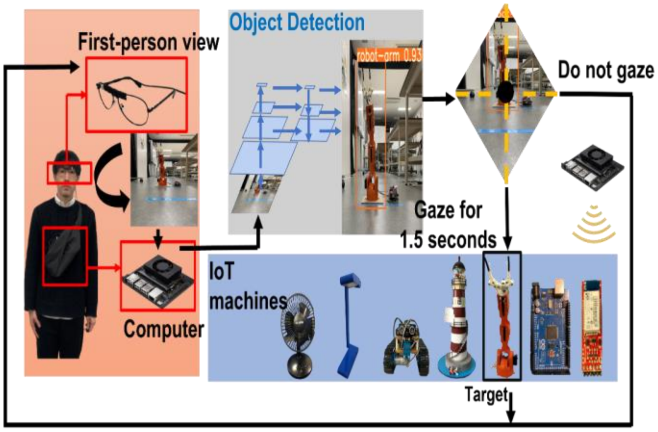


Fig. 2. Outline of the proposed system.

Here, a small camera is attached to the eyeglasses worn by a user. This camera captures and sends the image to a computer to be further processed by a neural network running on the computer for object detection. Here, the objects to be detected must be pre-specified for training the neural network, although the type and number of the objects are not constrained. YOLOv5 [19, 20] is utilized for the neural network's easy implementation and fast response. Fig. 3 shows the five machines as targets in this study.

The input to YOLOv5 is the image perceived by a human through the attached camera, and the output is bounding boxes, the center normalized coordinate of objects in the images, their heights and widths, and their IDs, as shown in Fig. 4.

To train the neural network, 3427 images for training data and 773 images for test data consisting of the five machines in various postures and distances were generated and labeled. The learning results are presented in the next section.

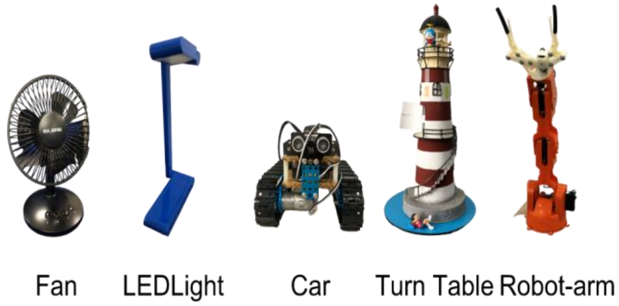


Fig. 3. Appearances of five machines.

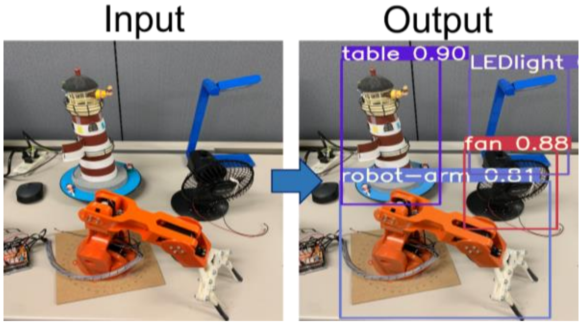


Fig. 4. The output of YOLOv5.

After detecting an object with YOLOv5, the system checks whether the human gaze is focused on that object. Here, in establishing eye contact, it is assumed that the human always put the intended object at the center of his/her field of view. Thus, the center of the coordinate of the obtained images is treated as the focus of the gaze. The system checks that the eye contact object is within the gaze focus for 1.5 seconds. Subsequently, the system refers to the object ID and sends a signal to the target machine. Here, each target machine is connected to a control PC wirelessly via Bluetooth.

By executing the above process in real time, the proposed Gaze Switch system is realized.

III. Experiments

The viability of the proposed Gaze Switch is assessed through user experiments. Before the user experiments, some preliminary experiments were run for verifying the basic operability of the proposed system.

The preliminary experiments were run on Windows 10 Pro Intel® Core™ i7-9700k CPU @3.60GHz to 4.90GHz 16.00GB and Nvidia® GTX750Ti GPU @1020MHz to 1085MHz, while the user experiments were run on Jetson Xavier NX [21], operated by Jetson Pack 4.5.1 for improving the system's compactness and processing speed. The neural networks for both experiments were the same.

A. Preliminary Experiments

In the preliminary experiment, the detection range of the neural network was assessed. The necessity of this assessment is due to the existence of a natural range of eye contact and whether the proposed system adheres to this natural range [22]. In particular, the mean Average Precision (mAP) against validation data was evaluated as shown in Fig. 5.

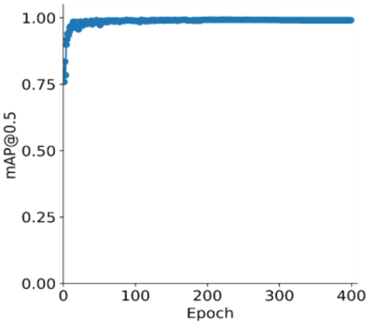


Fig. 5. Learning precision.

From Fig. 5, it can be observed that the mAP exceeds 0.9 after the training process, indicating that the neural network can learn the object detection task.

Next, validation data were created for evaluating the neural network's detection accuracy. Here, 700 labeled images of the seven objects at various distances are generated and checked for their detection accuracies. Fig. 6 shows the results of the accuracy test regarding the distance of the object (with the example of the robotic arm).

This figure shows that the accuracy does not significantly decrease until 5 meters. This indicates that the operating range of the system is around 5 meters which is similar to the human's natural range for eye contact.

The preliminary experiments indicate that the proposed system is viable for establishing eye contact intuitively and naturally.

After assessing the neural network's learning and detection capabilities, two experiments with nine human subjects were conducted. The main purpose was to verify the basic usability and operability of the Gaze Switch. Before the experiments, the subjects were explained the objective of the proposed interface.

After that, the subjects practiced using the interface for about five minutes. In the experiment, a monitor in front of the subject randomly showed an object that the subject must operate. Here, the subject should try to operate the specified machine by looking at it. The interface was evaluated in whether the human subject could operate the object within a specific time range. In the experiment, the subjects were instructed ten times in random order to operate each target twice.

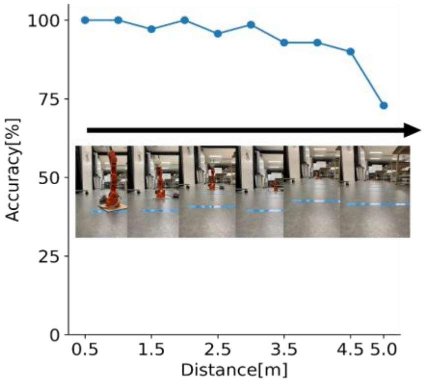


Fig. 6. Accuracy across distances.

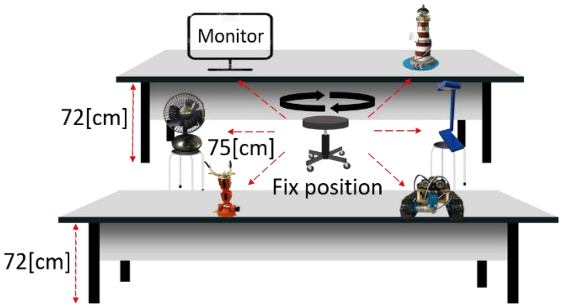


Fig. 7. Settings for preliminary experiment 1.

In this experiment, the target objects were randomly positioned but fixed to their respective position. During the experiment, the subjects were instructed to sit in a fixed position and use a swivel chair to turn their bodies to establish eye contact with the objects (see Fig. 7).

The human subjects were instructed to operate an object that randomly appears on the monitor within 10 seconds, and then return his/her gaze to the monitor. If the subject fails to operate the object within 10 seconds, the experiment continues, but the task is considered a failure. Fig. 8 shows the average accuracy for operating the instructed object.

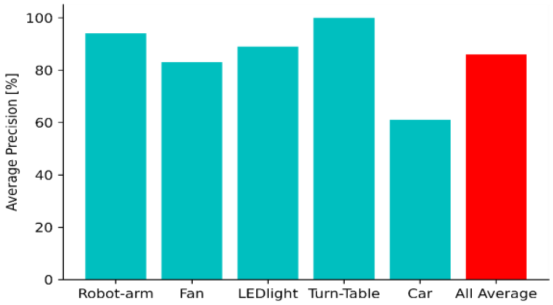


Fig. 8. The result of preliminary experiment 1.

The overall average accuracy was 88% which indicates the subjects were able to operate the specified targets. However, the accuracy rate for the object "Car" is low. This low accuracy is due to "Car" moving away from its fixed position. This indicates that it may be difficult for a human to operate moving objects from a fixed position using this interface.

Next, Preliminary Experiment 2 was conducted with the setup shown in Fig. 9.

The same task was applied to this experiment as in Preliminary Experiment 1. However, in this experiment the subject was allowed to freely move rather than operate from a fixed position. This experiment aims to evaluate the operability of this system in an environment that is more similar to a general living space. Fig. 10 shows the average accuracy in Preliminary Experiment 2.

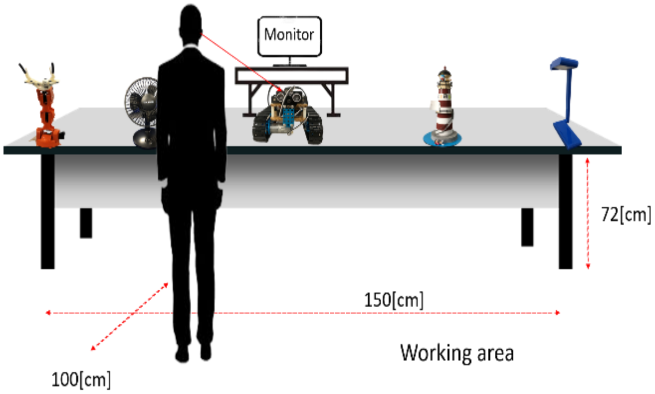


Fig. 9. Settings for preliminary experiment 2.

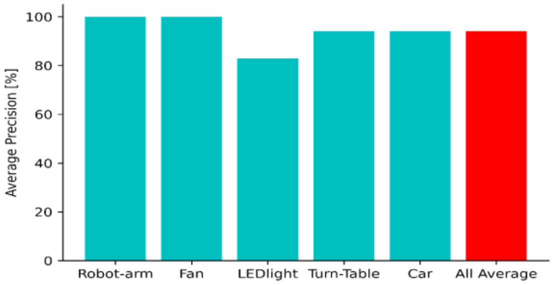


Fig. 10. The result of preliminary experiment 2.

The results in Preliminary Experiment 2 are better than those in Preliminary Experiment 1. In particular, the average accuracy for the object Car is significantly improved. This accuracy is because each subject can move his/her body in a way that makes it easier to follow the object when necessary.

From these experiments, the operability range and usage of the proposed system can be learned. The insights gained from the preliminary experiments are then utilized for users' assessment tests.

B. User Assessment Experiment

User assessment experiments were conducted to investigate the usability of the proposed eye-contact system and to assess its potential for its usage of interface modality. As an evaluation index, usability defined in ISO 9240-11 [23] was used. This criterion encompasses efficiency, effectiveness, and user satisfaction when a user executes a specific task to achieve a goal. This section explains the experiment method and the analytical results. A short demo movie for this experiment can be accessed from https://youtu.be/rKBbP2aLcxY.

A conventional remote-control switch was utilized as a benchmark against the proposed Gaze Switch. The outline of a remote-control switch is shown in Fig. 11.

Here, a target machine can be activated/deactivated by pushing a correlated button in the remote controller (RC) like selecting a TV channel.

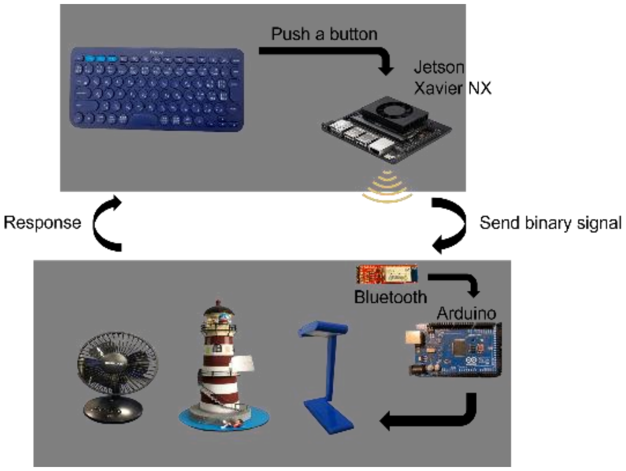


Fig. 11. Outline of the remote-control switch.

Naturally, as many subjects have already been familiar with RC for a long time, it is easy to predict that RC yields higher usability measures. Therefore, the objective of this experiment is not to directly compare the usability measure of the proposed Gaze Switch with RC but to compare the improvement of the usability over some repeated experiments. Here, it is enough to argue that if the users experience improvements in their familiarity and operability of the Gaze Switch over some repeated experiments, then the validity of the Gaze Switch as a new modality can be confirmed. It is also important to show that the improvements' signatures are similar to those of RC. Here, the number of repeated experiments was four with an interval of one week before the next experiment.

In the experiment, the task is based on a scenario where a user activates a household appliance while reading a book at home. This scenario is applied based on the results of the preliminary experiments where the fixed position of users yields worse results so that the rigorousness of the test is guaranteed. Fig. 12 illustrates the experimental environment.

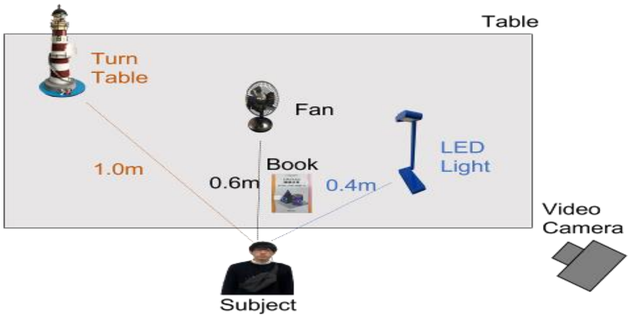


Fig. 12. The setting for the user assessment experiments.

The procedure was as follows.

Step 1. The subject sits and reads a book.

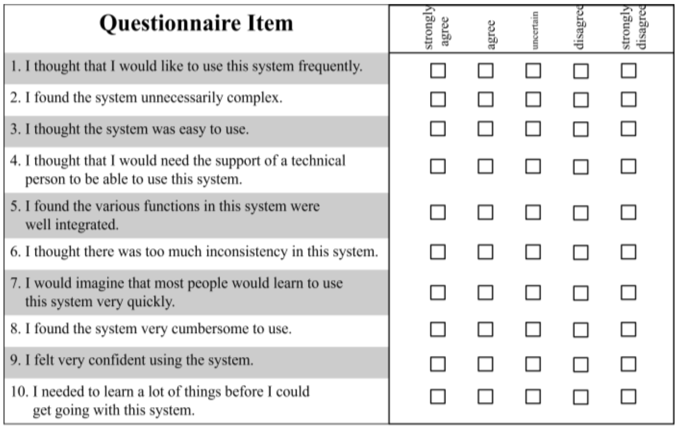
Step 2. A control PC instructs the subject to activate or deactivate one of three designated machines: Fan, LED Light, or Turn Table, in a random manner.

Step 3. The subject stops reading and activates/deactivates the machine with an interface at hand.

Step 4. The subject resumes reading upon confirming the activation or deactivation.

Go to Step 1 until the terminal condition is met.

During the experiment, the reaction time from the instruction to the activation/deactivation of the specified machine and the accuracy of the interaction were evaluated. Additionally, the users were asked to complete the questionnaire for measuring their satisfaction based on System Usability Scale (SUS) [24] on a 5-point Likert scale as shown in Fig. 13. The result of SUS is standardized in the range of 0 to 100. A null hypothesis test regarding the significant difference between the interfaces' characteristics and the familiarity factor due to the repeated usage of these usability-quantitative data was run. Here, the significance of the difference is assessed through a p-value with a 0.05 threshold. Here, the null hypothesis is that the evaluated factors are identical regarding the usage of RC and Gaze Switch.



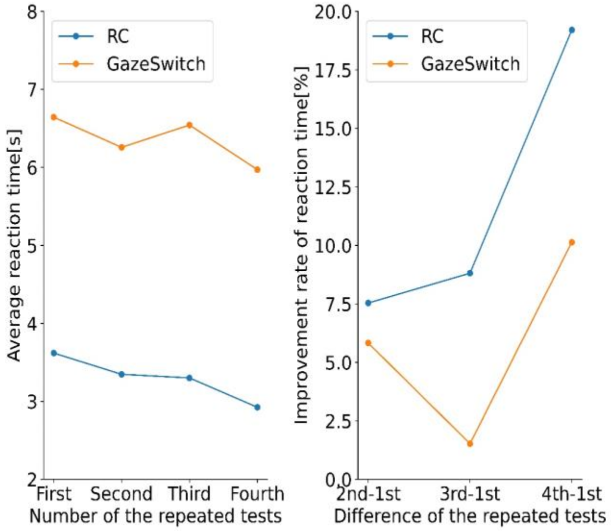
Questionnaire Item	strongly agree	agree	uncertain	disagree	strongly disagree
1. I thought that I would like to use this system frequently.	□	□	□	□	□
2. I found the system unnecessarily complex.	□	□	□	□	□
3. I thought the system was easy to use.	□	□	□	□	□
4. I thought that I would need the support of a technical person to be able to use this system.	□	□	□	□	□
5. I found the various functions in this system were well integrated.	□	□	□	□	□
6. I thought there was too much inconsistency in this system.	□	□	□	□	□
7. I would imagine that most people would learn to use this system very quickly.	□	□	□	□	□
8. I found the system very cumbersome to use.	□	□	□	□	□
9. I felt very confident using the system.	□	□	□	□	□
10. I needed to learn a lot of things before I could get going with this system.	□	□	□	□	□

Fig. 13. Question items of the system usability scale.

Eq. (1) shows the improvement rate, $IR_{i,j}$ of the j-th factor in the i-th repeated test, where $R_{i,j}$ is the score of factor j in the i-th experiment where $j \in \{\text{Reaction time, Accuracy, SUS score}\}$ and $i \in \{\text{Second, Third, Fourth}\}$. This experiment was conducted with 12 subjects.

$$IR_{i,j} = \frac{R_{i,j} - R_{First,j}}{R_{First,j}} \times 100\# \qquad (1)$$

First, the average reaction time needed for operating the specified object with RC and Gaze Switch for the respective test is shown in Fig. 14(a) and the IRs are shown in Fig. 14(b).



(a) Average. (b) Improvement rates.

Fig. 14. Reaction time.

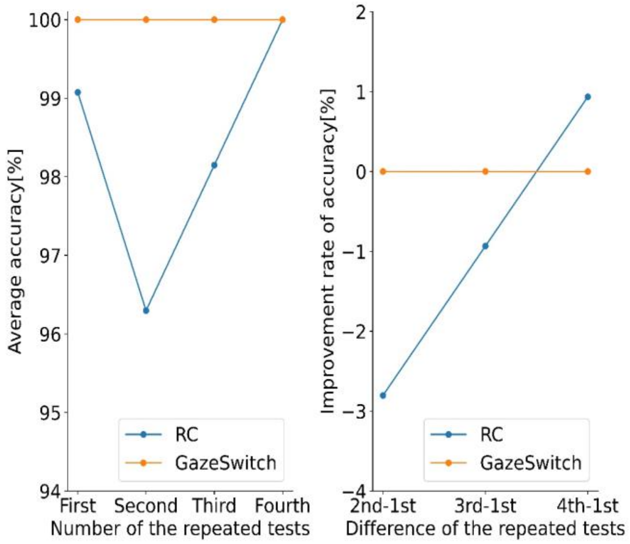
RC is superior to Gaze Switch with regard to the reaction time as can be learned from Fig. 14(a). To check whether the difference is significant, a Wilcoxon signed rank test was conducted that showed a significant difference (p<0.001). In addition, a Friedman test on the familiarity factor of RC and Gaze Switch was conducted. The result showed significant differences between the repeated tests on RC (p=0.002) but no significant differences on Gaze Switch (p=0.376). Fig. 14(b) shows that the gap in the reaction time between RC and Gaze

Switch does not decrease with the number of tests, and this trend can be predicted to stay true. Hence, it can be argued that the difference regarding the reaction time does not depend on familiarity due to the repeated usage of the two interfaces but depends on the basic characteristics of the two interfaces. Table I shows the average reaction time of each machine in the experiment on the respective test.

During the experiments, Gaze Switch allows the subjects to interact with the machines by perceiving them for 1.5 seconds but the latency compared to RC's reaction time was about 3.0 seconds. This latency is due to a discrepancy between the human's field of view and that of the camera, but not the system's operating range.

Next, the average accuracies (the correctness of activating/deactivating the instructed object) using RC and Gaze Switch for the respective test are shown in Fig. 15(a) and IRs are shown in Fig. 15(b).

It is obvious from Fig. 15(a) that the subjects did not need a long time to get familiar to use the Gaze Switch, indicating its good intuitiveness. The intuitiveness of the Gaze Switch is further emphasized in Fig. 15(b). Regarding the difference, the result of a Wilcoxon signed rank test on the interface factor showed a significant difference (p=0.008) between the accuracy of Gaze Switch and RC. In addition, a Friedman test on the familiarity factor in RC was conducted with no significant differences in RC (p=0.137). The significance test results show that RC is not necessarily stable in its intuitiveness. This is because the users occasionally misoperate the machines with RC due to the failure to memorize the relation between the buttons in the RC and the machines. By contrast, with the proposed Gaze Switch, the user can operate an intended machine by looking at it, so it does not need any memorization. From this experiment, it can be argued that the intuitiveness of the proposed interface contributes to its accuracy. Table II shows the accuracy of the respective machines.



(a) Average (b) Improvement rates

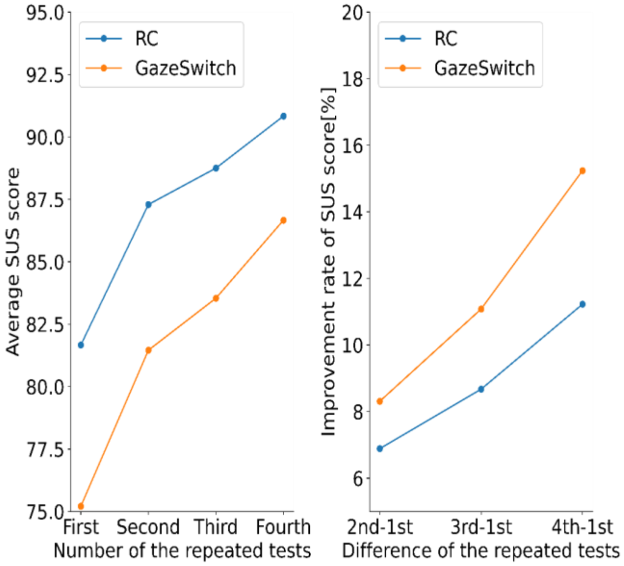
Fig. 15. Accuracy.

TABLE I. AVERAGE REACTION TIME

	RC Average Reaction Time (s)			**Gaze Switch Average Reaction Time (s)**		
Trials	***Fan***	***TurnTable***	***LEDlight***	***Fan***	***TurnTable***	***LEDlight***
First	3.81	3.54	3.51	6.39	6.93	6.61
Second	3.44	3.27	3.33	5.73	6.70	6.34
Third	3.33	3.54	3.03	6.44	6.89	6.30
Fourth	2.93	2.81	3.03	6.04	5.94	5.93
All	3.38	3.29	3.29	6.15	6.62	6.30

TABLE II. AVERAGE ACCURACY

	RC Average Accuracy (%)			**Gaze Switch Average Accuracy (%)**		
Trials	***Fan***	***TurnTable***	***LEDlight***	***Fan***	***TurnTable***	***LEDlight***
First	100	97.2	100	100	100	100
Second	97.2	94.4	97.2	100	100	100
Third	97.2	97.2	100	100	100	100
Fourth	100	100	100	100	100	100
All	98.6	97.2	99.3	100	100	100



(a) Average (b) Improvement rates

Fig. 16. SUS score.

Finally, the average SUS scores of RC and Gaze Switch are shown in Fig. 16(a) and the IRs are shown in Fig. 16(b).

Regarding the SUS score, RC is inferior to Gaze Switch as indicated in Fig. 16(a). The value for Cronbach’s Alpha (0≦a ≦1) to measure whether each item of the questionnaire is reliably able to measure the identical concept (here, satisfaction) by confirming the average covariance between pairs of the items, and the variance of the total score, was a = 0.83. These results indicate the result of the questionnaire is reliable because the value is 0.8 or more. Meanwhile, a Wilcoxon signed rank test to the result on the interface factor showed a significant difference (p=0.014). Likewise, A Friedman test on the familiarity factor in each RC and Gaze Switch showed significant differences in RC (p=0.027) and Gaze Switch (p<0.001). A Wilcoxon signed-rank test on the familiarity factor in RC and Gaze Switch was also conducted. The result showed a significant difference between the First and Forth of the repeated tests in Gaze Switch (p=0.004 after Bonferroni correction for the multiple comparisons problems). Fig. 16(b) shows that the difference in questionnaire scores and the number of repeated tests is getting wider.

It is restressed here that the objective of the experiments is not to directly compare the performance of the RC against the proposed Gaze Switch. The primary objective is to investigate the characteristics of the Gaze Switch in its usage as an interface modality using RC as a baseline. The experiments indicate that the Gaze Switch shows good user intuitiveness. Regarding the reaction time and the user SUS, the Gaze Switch shows good familiarity growth, meaning repeated usage will yield better experiences. The growth trends are also similar to that of a more established interface of RC, which shows the appropriateness of the proposed Gaze Switch as a new modality in the human-machine interface.

IV. Conclusion

In this study, we developed a hardware framework for expanding an intuitive and familiar communication modality, eye contact, for human-machine interaction. The proposed system allows humans to intuitively operate machines through eye contact. Unlike the existing gaze interfaces that often depend on specialized tools, the system allows direct interaction with various machines, thus offering better flexibility and intuitiveness. The users’ assessment tests in this study demonstrate that familiarity with eye contact in human daily communications translates into intuitiveness and robustness of the system. Through this study, it can be argued that eye contact is a reasonable modality in the human-machine interface.

The authors are aware of some technical drawbacks of the proposed system. For example, the relatively long reaction time decreases the usability of the proposed system. This is due to the discrepancy between the human field of view and the field of view captured by the camera. In the near future, this problem can be alleviated by better calibration of the camera or using a multi-camera system to align the view better.

Like the rich modalities in human interactions, in the future, the proposed Gaze Switch is not intended for single usage but in combination with other modalities, for example, verbal and nonverbal interfaces. The combinations of various interfaces will improve the precision of human-machine interactions and remove the difference between inter-human interactions and human-machine interactions. The seamless integration of machines into human interactions in daily life is one of the most important aspects in the coming era of AI technology, Metaverse, and XR, and hence the proposed eye-contact system has good potential for enriching the existing modalities for human-machine interfaces.

References

[1] G. Mandler, “Organization and Memory,” Psychology of Learning and Motivation, vol. 1, pp. 327-372, 1967.

[2] S. M. Polyn, K. A. Norman, and M. J. Kahana, “Task context and organization in free recall,” Neuropsychologia, vol. 47, no. 11, pp. 2158-2163, 2009.

[3] M. Hirokawa, K. Inoue, T. Iwaki, and T. Kashima, “Fundamental Study on an Intuitive Interface Design Theory -Approach from Viewpoint of Organization and Familiarity,” Transaction of Japan Society of Kansei Engineering, vol. 13, no. 5, pp.543-554, 2014.

[4] A. Mehrabian, “Communication without words,” Psychological Today, vol. 2, pp. 53-55, 1968.

[5] M. Choi, D. Sakamoto, T. Ono, “Bubble Gaze Cursor + Bubble Gaze Lens: Applying Are Cursor Technique to Eye-Gaze Interface,” in Proceedings of the 2020 ACM Symposium on Eye Tracking Research and Applications Full Papers, no. 11, pp. 1-10, 2020.

[6] M. Choe, Y. Choi, J. Park, “Comparison of Gaze Cursor Input Methods for Virtual Reality Devices,” International Journal of Human-Computer Interaction, vol. 35, issue. 7, pp. 620-629, 2018.

[7] R. J. K. Jacob, “The use of eye movements in human-computer interaction what you get,” ACM Transactions on Information Systems, vol. 9, no. 2, pp.152-169, 1991.

[8] C. Lutteroth, M. Penkar, G. Weber, “Gaze vs. Mouse: A Fast and Accurate Gaze-Only Click Alternative,” in Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology pp. 385-394, 2015.

[9] J. M. Araujo, G. Zhang, J. P. P. Hansen, S. Puthursserypaddy, “Exploring Eye-Gaze Wheelchair Control,” ACM Symposium on Eye Tracking Research and Applications, no. 16, pp. 1-8, 2020.

[10] S. Jaiswal, S. Virmani, V. Sethi, K. De, P. P. Roy, “An intelligent recommendation system using gaze and emotion detection,” Multimedia Tools and Applications, vol. 78, pp.14231-14250, 2018.

[11] A. Ledezma, V. Zamora, O. Sipele, M. P. Sesmero, A. Sanchis, “Implementing a Gaze Tracking Algorithm for Improving Advanced Driver Assistance Systems,” Electronics, vol. 10, no. 12, 1480, 2021.

[12] A. Kendon, “Some functions of gaze-direction in social interaction,” Acta Psychologica, 26:22-23, 1967.

[13] A. Senju, and M.H. Johnson, “The eye contact effect: mechanisms and development,” Trends in Cognitive Sciences, vol. 13, no. 3, pp. 127-134, 2009.

[14] M. S. Cary, “The Role of Gaze in the Initiation of Conversation,” Social Psychology, vol. 41, no. 3, pp. 269-271, 1978.

[15] I. Kawaguchi, H.Kuzuoka, and D. McMillan, “The Effect of Interaction Using Gaze Input/Output on Smart Speaker,” The Transactions of Human Interface Society, vol. 21, no. 3, 2019.

[16] J. Kim, S. Choi, and J. Jeong, “Watch & Do: A smart IoT inter-action system with object detection and gaze estimation,” IEEE Transactions on Consumer Electronics”, vol. 65, no. 2, pp. 195-204, 2019.

[17] F. Putze, D. Weiß, L. -M. Vortmann, T. Schultz, “Augmented Reality Interface for Smart Home Control using SSVEP-BCI and Eye Gaze,” 2019 IEEE International Conference on Systems Man and Cybernetics (SMC), pp.2812-2817, 2019.

[18] K. Syusuke, and P. Hartono, “Development of Eye Gaze Switch,” Proceedings of the Forum on Information Technology 2021, J-007, Vol. 3, pp. 255-257, 2021.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.779-788, 2016.

[20] G. Jocher, YOLOv5 by Ultralytics (Version 7.0). https://doi.org/10.5281/zenodo.3908559, 2020.

[21] Nvidia, Inc.: Jetson Xavier NX; https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/.

[22] S. Daniel, “Eye contact perception at distances up to six meters,” Portland State University Dissertations and Theses, Paper 3693, 1985.

[23] International Organization for Standardization, “ISO9241-210:2020 Ergonomics of human system interaction Part 210: Human-centred design for interactive systems,” 2010.

[24] J. Brooke, “SUS-A quick and dirty usability scale,” in Usability evaluation in industry, pp. 189-194, 1996.

Pancreatic Cancer Segmentation and Classification in CT Imaging using Antlion Optimization and Deep Learning Mechanism

Radhia Khdhir[1], Aymen Belghith[2], Salwa Othmen[3]
Department of Computer Science-College of Science and Arts in Qurayyat, Jouf University, Saudi Arabia[1]
Computer Science Department-College of Informatics and Computing, Saudi Electronic University, Saudi Arabia[2]
Computers and Information Technology Department, College of Science and Arts, Saudi Arabia[3]

Abstract—**Pancreatic cancer, a fatal type of cancer, has a very poor prognosis. To monitor, forecast, and categorise cancer presence, automated pancreatic cancer segmentation and classification utilising Computer-Aided Diagnostic (CAD) model can be used. Furthermore, deep learning algorithms can provide in-depth diagnostic knowledge and precise image analysis for therapeutic usage. In this context, our study aims to develop an Antlion Optimization-Convolutional Neural Network-Gated Recurrent Unit (ALO-CNN-GRU) model for pancreatic tumor segmentation and classification based on deep learning and CT scans. The ALO-CNN-GRU technique's objective is to segment and categorize the presence of cancer tissues. This technique consists of pre-processing, segmentation and feature extraction and classification phases. The images go through a pre-processing stage to reduce noise from the dataset that was obtained. A hybrid Gaussian and median filter is applied for the pre-processing phase. To identify the pancreatic area that is affected, the segmentation is processed utilizing the Antlion optimization method. Then, the categorization of pancreatic cancer as benign or malignant is done by employing the classifiers of the Convolutional neural network and Gated Recurrent Unit networks. The suggested model offers improved precision and a better rate of pancreatic cancer diagnosis with an accuracy of 99.92%.**

Keywords—Pancreatic cancer; Antlion optimization; deep learning; convolutional neural network

I. Introduction

The pancreas is a vital organ in the human body that secretes both internally and externally and is prone to several illnesses. Deeper within the belly, the pancreas is a glandular organ [1]. It is made up of the endocrine and exocrine tissues, which, separately, are essential to digestion and glucose metabolism. Endocrine and exocrine pancreatic illnesses are divided into categories based on the tissue most frequently impacted by pathologies. Chronic pancreatitis, pancreatic cancer, and acute pancreatitis are exocrine pancreas diseases. There is an increasing realization that these three conditions exist on some kind of continuous sequence [2]. Pancreatic endocrine malfunction, a key factor connecting the various kinds of diabetes mellitus, characterizes disorders of the endocrine pancreas. Diabetes of the exocrine pancreas, type 1 DM and type 2 DM, make up the bulk of the spectrum in terms of prevalence in adults [3]. There are other forms of DM as well. Consequently, disorders of the endocrine and exocrine pancreas place enormous socioeconomic and health costs on people across the world.

Pancreatic cancer has a relatively high mortality rate that closely resembles its incidence rates. The fourth most frequent cause of cancer mortality and the twelfth most frequent cause of cancer overall is pancreatic cancer [4]. 62,210 new cases of pancreatic cancer are anticipated to be discovered in the USA in 2022 with a percentage of 3.2% of all cancer cases. Pancreatic cancer patients typically have vague symptoms, such as loss of weight and abdominal discomfort, which can delay detection. Usually, pancreatic cancer patients experience no symptoms up to an established stage of the illness. A standardized approach for monitoring individuals who are at risk of pancreatic cancer does not exist, yet [5]. Pancreatic intraepithelial neoplasias, which are tiny, non-invasive epidermal proliferations inside the pancreatic ducts, are the primary causes of most pancreatic malignancies. The exceedingly dismal outcome for pancreatic cancer is underlined by the strong correlation between disease occurrence and death. In USA, the five-year survival rate for those with pancreatic cancer is still as low as 7%. The terminal stage at which the majority individuals are identified is among the causes of the lower survival rate and is arguably the most significant one. Prior to the illness progressing to an advanced phase, the majority of pancreatic cancer patients have no symptoms [6].

Numerous techniques based on deep learning and machine learning have recently been proposed to identify pancreatic cancer because of the invention of Computer-Aided Diagnosis (CAD). Accurate segmentation of the pancreatic and reliable categorization are often necessary for an automated and elevated pancreas cancer diagnostic model [7]. In everyday practice, pancreatic disorders are diagnosed and monitored using cutting-edge cross-sectional imaging techniques such as computed tomography, magnetic resonance cholangiopancreatography, magnetic resonance imaging, and endoscopic ultrasonography (EUS). Through these techniques, the complete pancreas may be seen in connection to the adjacent anatomy [8]. The most frequently employed imaging technique for the first assessment of suspicious pancreatic cancer is computed tomography. It is chosen over Magnetic Resonance Imaging (MRI) as the initial line modalities because it is less expensive and more widely accessible than MRI. A more accurate assessment of the pancreatic parenchyma and calcifications can be made using computed tomography. Structural information may be obtained from Computerized Tomography (CT) scans. The preferred imaging technique for diagnosing pancreatic illness nowadays is contrast-enhanced CT, which may also have the best standard specifications rates for pancreatic cancer. In various fields, the segmentation of the pancreas in CT images can help clinical processes, such as pancreatic cancer detection, treatment, and surgical support [9]. Therefore, it is worthwhile to investigate a reliable, accurate, and automated segmentation approach for the pancreas.

In abdominal computed tomography images, segmenting organs including the liver, spleen, and pancreas is essential for computer-aided diagnosis, quantitative and qualitative studies, and surgical support. In CAD systems that do quantitative image analysis of people with diabetes or pancreatic diagnostic tools, pancreas segmentation in particular is a crucial component. Because of its size, shape, and position in the abdomen on computed tomography scans, the pancreas is extremely difficult to segment [10]. In the area of organs segmentation, the network architectures of UNet, SegNet and FCN are especially well-liked, and all these techniques have been successfully used in the segmentation of big organs, such the heart, kidneys, and lungs. The segmentation of minor organs might still use different approach. Particularly challenging in terms of form learning are the extremely intricate anatomical components of the pancreatic. Second, there is a dearth of annotated medical imaging data [11]. As a result, the segmentation is unable to achieve a high level of precision. Additionally, the pancreas' location in the abdominal wall differs from patient to patient, and the border contrast is influenced by the amount of visceral fat that surrounds the pancreas. These additional elements are combined to make pancreatic segmentation difficult and susceptible to both over and under-segmentation. To address these issues, we aim to develop a powerful pancreatic segmentation approach that will improve the precision and robustness of pancreatic segmentation.

Because the pancreas is more varied in size and shape than other organs and is challenging to categorize with neighboring organs, segmenting it is a challenging process. The manual segmentation of the pancreas by radiologists requires a lot of time and effort. Additionally, there are variations in the manual results produced by radiologists. Even though advanced machine learning technology, techniques were put forth to separate the pancreas after extracting specific properties utilizing prior information [12]. Numerous deep learning-based studies have been recently completed as hardware efficiency and deep learning technologies have advanced. The results have outperformed earlier approaches in a variety of applications. The convolutional neural network, one of several deep learning methods, performs well for the segmentation and classification challenge and may be efficiently used to segment the pancreas [13].

The research’s main contributions are as follows:

- A significant number of patients' CT images are first gathered, and the CT datasets are analysed in the system.
- Furthermore, the recovered CT pancreatic images contain unwanted noises, which are filtered utilizing an advanced Gaussian and median filter.
- An upgraded Antlion Optimization model has been used for the segmentation process.
- Gray Level Co-occurrence Matrix was used for feature extraction.
- The ALO-CNN-GRU classifies the affected regions as benign and malignant.
- The efficacy of the suggested technique is demonstrated by validating its performance and comparing it to current methodologies.

The paper is structured as follows. Several related works are covered in Section II. The suggested ALO-CNN-GRU architectures are explored in depth in Section III. In Section IV, experiment findings are shown, evaluated, and a thorough assessment of the suggested strategy in comparison to current best practices is made. In Section V, the final portion of the document, the work is concluded.

II. Related Works

In order to improve the segmentation of pancreatic cancer, Jun Li et al. suggested a dual meta-learning approach based on idle data [14]. For clinical treatment and detection of pancreatic cancer, automated segmentation is essential. The segmentation performance is however constrained by the tiny size and hardly perceptible borders. Researchers gather pancreatic cancer unused multi-parametric Magnetic resonance imaging from numerous researches to build a significantly larger dataset for improving the Computed tomography pancreatic cancer segmentation in order to resolve the issue brought on by the small-scale dataset. So, for pancreatic cancer, researchers suggest a deep learning segmentation technique with a dual meta-learning structure. Elevated characteristics become more discriminatory as a result of its ability to combine salient information from CT images. In order to flawlessly satisfy the gaps in visual appeal as well as provide rich intermediate depictions for the ensuing meta-learning scheme, the arbitrary transitional methods between CT and MRIs are generated. Researchers then use prototype meta-learning based on transitional modes to identify and transmit similarities. The distraction caused by internal variations is finally reduced by using a meta-optimizer to flexibly understand the relevant characteristics inside CT data. The suggested approach is a robust pancreatic cancer segmentation approach that is simple to implement into existing segmentation networks and has promise as a viable model for addressing the problems of data scarcity with idle data but its accuracy is very low when compared to the other existing models.

A Deep Learning-Enabled Automated Medical Decision-Making System for Pancreatic Tumor Diagnosis on Computed Tomography was proposed by Thavavel Vaiyapuri et al. [15]. The computer-aided diagnostic model can automatically identify and categorize pancreatic cancers. Deep learning and Machine learning models that have lately been created can be utilized to automatically and quickly diagnose pancreatic cancer. The study introduces an innovative deep-learning-enabled medical decision-making system for classifying pancreatic tumors from CT scans. The IDLDMS-PTC technique's primary goal is to assess the CT images for the presence of pancreatic tumors. An emperor's penguin optimizer with multilayer thresholding approach is derived using the IDLDMS-PTC model for segmenting pancreatic tumors. Furthermore, the MobileNet algorithm is used as an optimum feature extractor for classifying pancreatic tumors. The multileader optimization approach is employed to alter the bias and weight parameters of the AE approach in the best possible way. The innovative features are demonstrated by the EPO algorithm's design for selecting the best threshold and by the MLO algorithm's design for parameter adjustment. Numerous simulations were run on data sets, and the results showed that the IDLDMS-PTC model performed well compared to other techniques but the model still needs improvements.

The Anatomy-Aware Transformers were addressed by Yingda Xia et al. for the efficient pancreatic cancer screening [16]. The most fatal malignancy, pancreatic cancer is quite infrequent. It is not advised to screen the entire asymptomatic populace because to the possibility that a sizable proportion of false positive people may have unneeded imaging tests, considerably increasing health care expenditures with no discernible patient benefits. In the study, researchers explore the viability of detecting respectable pancreatic masses utilizing a single-phase non-contrast CT scan and classifying the detected masses as pancreatic ductal adenocarcinoma, other anomalies, or healthy pancreas. The majority of the time, regular radiologist or perhaps even pancreatic experts perform the duty ineffectively. Researchers suggest a novel deep classification method with an anatomy transformer employing pathology verified mass kinds and transfer of knowledge from contrast-enhanced Computed tomography to non-contrast Computerized Tomography as supervision. The research identifies a possible new instrument with considerably higher accuracy and reduced computational risk and cost for widespread pancreatic screening and treatment. However, the proposed instrument is not able to analyze vast amounts of data.

Xiaoyu Yang et al. used Interpolation neural network with local linear embedding for cancer progression segmentation [17]. The thick Computed tomography with its low resolution and wide spacing increases the risk of misinterpretation and makes it extremely difficult to automatically separate organs and tumors. Because of the technology limitations, particularly in automated pancreatic tumor segmentation, there is poor consistency among segments of the 3D CT image containing few tumors per image. In addition, tumor characteristics, such as size, form, localization, and severity, vary greatly between instances. The segmentation process is very unclear due to the hazy borders of tiny tumors. Researchers integrate the LLE-based interpolation neural network into the pancreatic tumor segmentation challenge in an effort to address these issues, which primarily incorporates the following enhancements. To represent the connection between nearby segments and the interpolated segment, researchers use local linear embedding. It adjusts the organ's geometric transition between segments. The LLE module and neural network work together to greatly improve image quality, resulting in more consistent and sharper images being produced for each sequence. The system utilizes a multiscale cascading technique to lessen the impact of sudden changes in tumor size on segmentation outcomes. Furthermore, to improve accuracy, the mechanism must be improved.

Vahid Asadpour et al. analyzed pancreatic cancer tumors in Computed tomography images using a patch-based multi-resolution convolutional neural network. In the article, researchers suggested a cascaded framework for extracting the volumetric form of the tumor and pancreas in adenocarcinoma patients. The method combines an elastic atlas that can fit on 3D volumetric shape retrieved from Computed tomography slices, a convolutional network with multiple forward pathways to recognize the fragments of images with particles to good resolutions using a multi-resolution structure. Geometrical characteristics that were altered both globally and per organ were employed to value the atlas organs. A multiresolution CNN was utilized for the categorization of image patches. By using an edge detection technique, the pancreas and tumor were finally segmented. The completely cascaded technique that has been developed outperformed all other methods in comparison. However, it takes a lengthy time to analyze data compared to other approaches [18].

The Preoperative Estimation of Pancreatic Survival Rates and Operational Margins utilizing Contrast-Enhanced CT Imaging were described by Jiawen Yao et al. [19]. One of the deadliest fatal malignancies, pancreatic ductal adenocarcinoma, has a poor prognosis. For individuals who are qualified for first treatment of PDAC, surgery continues to provide the highest chance of a possible cure. Nevertheless, even among resected individuals who were at the similar stage and got identical therapies, results might differ dramatically. In the study, researchers introduce a unique deep neural network called 3D CE-ConvLSTM, which can extract the tumor regression characteristics or features from CE-CT imaging techniques, for the survival prediction of PDAC victims. Researchers describe a multi-task CNN that can predict margins and outcomes, and which gains from understanding variables associated with tumor resection margins to enhance survival prediction. Comparing the suggested framework to current cutting-edge techniques to survival analysis, prediction performances must be improved.

An Effective Deep Learning-Based Pancreatic Cancer and Non cancer Categorization Model Utilizing CT scans were proposed by Maha M. Althobaiti et al. [20]. The prognosis for pancreatic tumors, which are a fatal kind of tumor, is quite dismal. To monitor, anticipate, and categorize the presence of pancreatic tumors, an automated pancreatic tumor categorization employing a computer-aided diagnostic model is required. Artificial intelligence can provide in-depth diagnostic knowledge and precise picture interpretations during treatment. The work develops an ODL-PTNTC model for optimum deep learning-based pancreatic tumor and nontumor categorization utilizing CT images. The ODL-PTNTC method's objective is to identify and categorize pancreatic tumors and non-tumors. Adaptive window filtering is a method included in the suggested ODL-PTNTC approach to eliminate noise. Additionally, the image segmentation procedure uses the Sailfish Optimizer based Kapur's Thresholding approach. A collection of feature vectors is also produced through feature extraction utilizing the Capsule Network. For categorization reasons, Political Optimizer with CFNN is also used. The performance of the classifier of the ODL-PTNTC approach can be enhanced by DL-based segmentation methods.

III. Proposed Methodology

In the beginning, pancreatic cancer images from Computerized Tomography (CT) are gathered. After that, the images are employed for testing and training. Pre-processing of pancreas-dependent CT images involves the removal of extraneous noise using a combined Gaussian and median filter. In this study, the nodules associated with pancreatic cancer are found and segmented using the Antlion optimization method. The Grey Level Co-occurrence Matrix is then used to extract the features. The suggested ALO-CNN-GRU method is utilized to categorize the aggressiveness of pancreatic cancer nodules. Fig. 1 represents the proposed method's framework.

A. Dataset Description

From the General Hospital of the Shenyang Military Area Command, 80 patients' CT data were analyzed. A total of 1700 images in Digital form are included in every pair of CT imaging sequence. Each CT picture is 512 x 512 pixels in resolution. There are 40 patients who have pancreatic cancer, while the others are well. Both malignant and benign pancreatic cancer images can also be found in the CT imaging. Three nuclear medicine specialists choose the descriptions for the images. Researchers perform segmentation tests using the NIH pancreatic segmentation dataset, which comprises 82 abdominal contrast-enhanced 3-Dimensional CT images, to fairly compare our segmentation approach to others. The surface pancreatic segmentation masks have indeed been submitted by a med student and have been reviewed by a radiologist. 1782 CT data are present, of which 863 serve as the training datasets and 919 serve as the testing datasets.

B. Pre-Processing

The initial stage in the identification of pancreatic cancer is pre-processing. It is employed to remove extraneous data and fill in dataset shortages. The evaluation of the sample images is slowed down by unrelated and unexpected sounds that affect the computed tomography images. The speckle noises that are brought on by internal and external causes mostly impact the CT images. Therefore, a hybrid filter using a mixture of Gaussian and median is applied in this research to minimize noise in CT pancreatic images. The Gaussian filter is applied to minimize the noise in the CT scans as well as the residual variations of geographical intensities. The Gaussian filter is utilized to replace the noisy pixels in the images with the Gaussian-distributed average value of the surrounding pixels. The median filter, on the other hand, may effectively remove spiky noises while maintaining the image's crisp edges and will also restore each pixel's grey level. In order to segment the images of pancreatic cancer, the noise-reduced images are employed in the ALO-CNN-GRU model.

The equation for the Gaussian filter is:

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{r^2}{2\pi\sigma^2}}) \tag{1}$$

Where σ is the standard deviation of the distribution. The distribution's mean is considered to be zero.

The equation for the median filter is:

$$\hat{g}(m,n) = median_{(s,t)\in T_{mn}}\{f(s,t)\} \tag{2}$$

Here $f(s,t)$ residual pixel following removal.

The improved equation for the pre-processing of CT images is:

$$\hat{g}(m,n) = median_{(s,t)\in T_{mn}}\{G(r)\} \tag{3}$$

Here *r* is the pixel dimension of every image. Every image is denoted by *m*.

C. Segmentation

The segmentation process is primarily employed to segment the affected region in the visualizations of CT images. The performance of image processing depends on how well the segmentation process works. Image segmentation is widely used to pinpoint an image's influenced areas as well as the constraints imposed by its curves and lines. The process of image segmentation divides each collection of pixels in the images into its own group of labels. In medical image processing, image segmentation's primary goal is to locate the cancer-affected regions and provide sufficient information for further identification. The identification of morphological characters and the detection of pancreatic cancer typically involve pancreas segmentation. Nevertheless, due to the significant anatomical heterogeneity of the pancreas, precise segmentation remains difficult. For this purpose, we segment the pancreas utilizing Antlion Optimization (ALO).

1) Antlion optimization: The Ant Lion (called Antlion) optimization algorithm imitates an antlion's natural hunting strategy. While antlions primarily hunt as larvae, the adult period is used for reproducing. A larval antlion moves in a circle while ejecting sand with its muscular jaws to create a cone-shaped opening in the sand. Following the construction of the traps, the larvae conceal underneath the base of the cone and waits for insect species ants—to enter the hole and become entangled. When the antlion discovers that there is prey in the trap, it tries to catch its victim. Although they try to get away from the trap, insects are often not immediately caught. In this case, antlions cunningly throw grains along the edge of the opening to assist the prey in slipping towards the bottom of the pit. A prey that is captured in the jaws is dragged under the ground and eaten. Antlions discard the remaining prey outside of the hole after eating it and prepare the pit for the subsequent hunt. Another noteworthy aspect of antlions' way of life is the relationship between the dimensions of the traps, one's personal level of hunger, and the form of the moon.

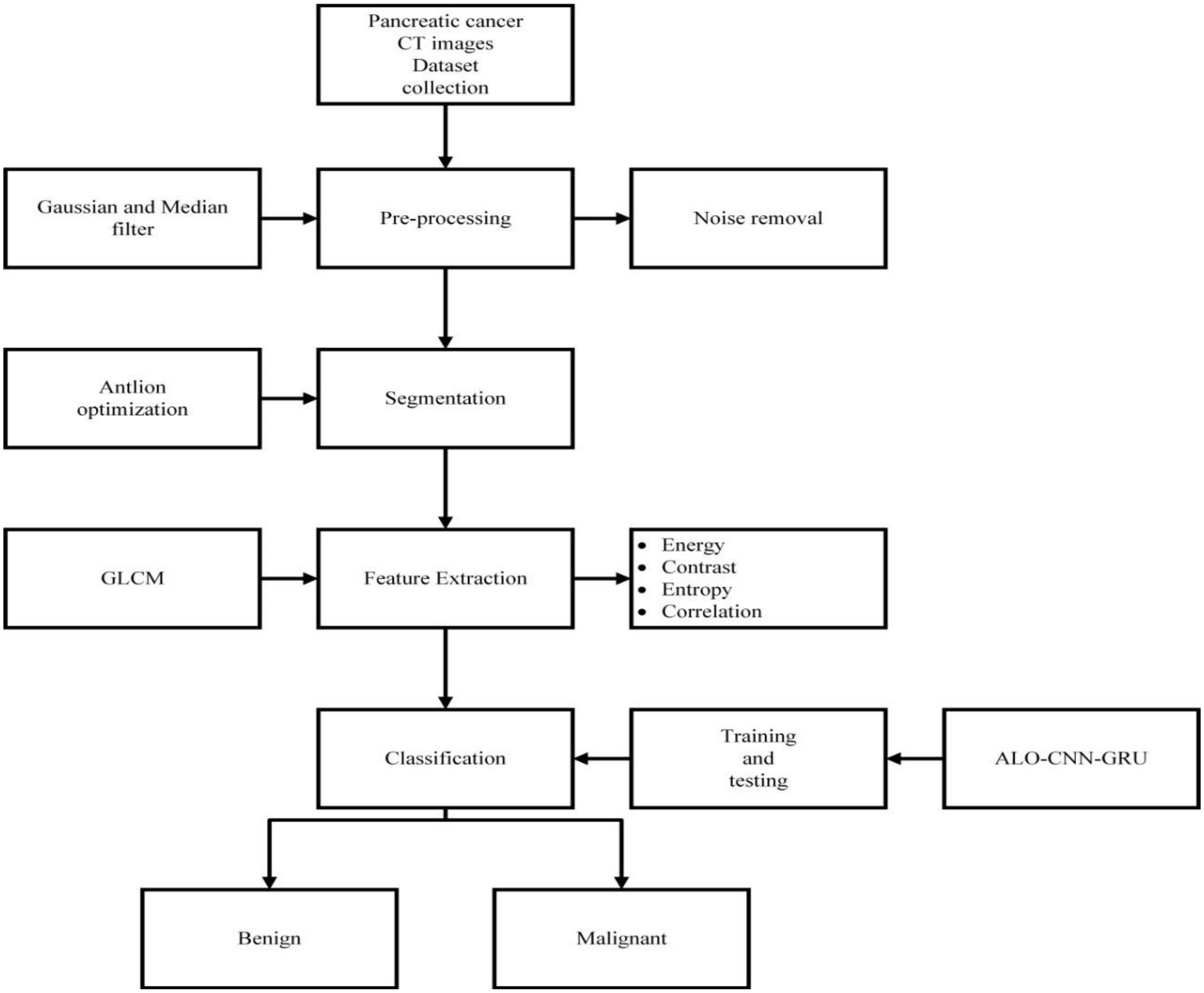


Fig. 1. Proposed method’s framework.

2) ALO algorithm operators: The ALO algorithm simulates ant behaviour in nature, including how they create trap and goes on the hunts. Ants taking lengthy walks at random. The ant moves arbitrarily throughout the search area in accordance with equation (4).

$$X_i^t = \frac{(X_i^t - b_i)\times(h_i - G_i^t)}{(h_i - b_i)} + G_i \tag{4}$$

where b_i is the random walk's i-th variable's minimum value, G_i^t is the i-th variable's minimal value at the t-th iteration, and h_i reflects the highest value for variable *i* at iteration *t*. The ant random walk is normalized to the maximum or minimum values of the search area if it occurs outside the search area.

3) Trapping in antlions pits: The antlion traps have an impact on the ants' random travels. It is demonstrated that a variety of variables govern the ants' random travels (g, h). Where g is the lowest consistent factor for i-th ant and h is the highest consistent factor for i-th ant.

$$g_i^t = Antlion_j^t + G^t$$

$$h_i^t = Antlion_j^t + h^t \tag{5}$$

4) Building trap: The ability of an antlion to hunt was modelled using a roulette wheel. The person operating the roulette wheel chooses the antlion based on its fitness because only one antlion is allowed to capture every ant. The likelihood of catching ants increases with an antlion's fitness.

5) Ants moving toward the antlion: The antlion begins to spray sand outward from the center of the trap once it understands there's an ant in the conic hole. Flexible reduction of the ant's random journey range is calculated as follows:

$$g^t = \frac{g^t}{I}$$

$$h^t = \frac{h^t}{I} \tag{6}$$

where g^t the smallest of all the variable at the t-th iterations, and h^t shows the vectors that contains the most variable at the t-th iterations.

6) Prey capture and trap construction: Subsequently, the antlion captures the ant and consumes its body when the ant grows stronger than it. To improve its chances of a new hunt, the antlion changes its location to match that of the hunted ant. The suggested equation is shown below.

$$Antlion_j^t = Ant_i^t, \ if \ f(Ant_i^t) > f(Antlion_j^t) \qquad (7)$$

The antlion known as the Elite-Antlion is the one that consistently provides the best answer across all iteration and searches domains. Every iteration compares the elite-antlion to the antlion only with best fitness. By using the ant's randomness algorithm throughout the roulette wheel (R_B^t) and the unplanned excursion of Elite-Antlion (R_P^t), the elite-antlion strategy affects how the ants wander randomly around the antlion.

$$Ant_i^t = \frac{R_B^t + R_P^t}{2} \qquad (8)$$

The complete flow chart of the working process is given below in Fig. 2.

```
Algorithm 1: ALO-CNN-GRU mechanism
Input: Computerized Tomography Images
Output: Benign, malignant
Import input CT image data
Let I be the input data that is taken for analysis
$I = \{I_1, I_2, I_3 \ldots.\}$
Pre-processing of images          //Gaussian and median filter
Segmentation of images            //Antlion Optimization
Create a population of n ants and n antlions at random starting sites
Determine each ant's and antlion's fitness
Locate the elite antlion;
t=0
While (t ≤ T)
    Foreach $ant_i$ do
        Use the roulette wheel to choose an antlion
        Entice ants to approach the antlion
        Construct a random walk to $ant_i$ and normalize
    End
    Compute each ant's level of fitness
    To become fitter, swap an antlion for the equivalent ant
    If an antlion gets more fit than the elite, upgrade the elite
    end if
    Feature extraction            //GLCM
    Classification                //CNN-GRU classifier
    Classifying as benign, malignant
    end if
end while
```

D. Feature Extraction

The feature extraction process entails transforming unstructured data into numerical qualities that may be used to protect the data included in the original set of information. These characteristics, which are obtained from every one of the CT images that were taken, represent how each patient interprets CT scans individually. During testing, the images' dimensionality is enhanced. However, in order to identify pancreatic cancer, the dimensionality of the images must be decreased. To fix this problem, the feature extraction process was being used.

In the feature extraction, the Gray Level Co-occurrence Matrix (GLCM) is utilized. By calculating the pairings of pixels with specific values, it may calculate the structure of the images. The GLCM analyses the image's grayscale to determine the pixels' brightness. Energy, Correlation, homogeneity, entropy, contrast, and other properties of the second-order representations are evaluated for the purpose of deleting the statistical texture characteristic.

1) *Energy:* Energy is described as the insertion of a square in images when the concentration levels are unequal, and the grey levels are often greater. Eq. (9) is used to determine the energy of the input data.

$$E = \sum_j \sum_k \{V(j,k)\}^2 \qquad (9)$$

where an image is indicated as V and grey level squares are denoted as (j, k).

2) *Contrast:* When an image's local contrast is evaluated, attributes are used, and when the concentration is even, a low value is calculated. The original images contrast and total number of grey levels are projected in eqn. (10)

$$C = \sum_{m-0}^{\mathrm{Pq}} \mathrm{m} \ \{ \sum_{j-1}^{\mathrm{Pq}} \sum_{k-1}^{\mathrm{Pq}} V\ (j,k)\} \qquad (10)$$

where P denote the grey level of the images, V denotes the images, and (j, k) denotes the grey level square of the images.

3) *Correlation:* The correlation characteristics could be utilized to account for both the linear dependency of the grey levels on the pixels and the numerical correlations between the variables. In eq. (11), the qualities are revealed.

$$\mathrm{C} = \frac{\sum_j \sum_k (j,k) V\ (j,k) - \mu\mathrm{m}\ \mu\mathrm{n}}{\sigma_m\ \sigma_n} \qquad (11)$$

The mean and standard deviation values, in the images, are μ_{m}, μ_{n} , $\sigma_{m,}\ \sigma_n$ which are defined as row and column, respectively.

4) *Entropy:* The expected significant amount of the unpredictability of the distribution of the grey level, which is indicated in eq. (12), is referred to as entropy.

$$En = -\sum_j \sum_k V(j,k) \log(V(j,k)) \qquad (12)$$

E. Classification using CNN and GRU

1) *Convolutional neural network:* Utilizing many CNN front layers, which are used to uncover patterns in images, convolutional neural networks are able to recognize lines and corners.

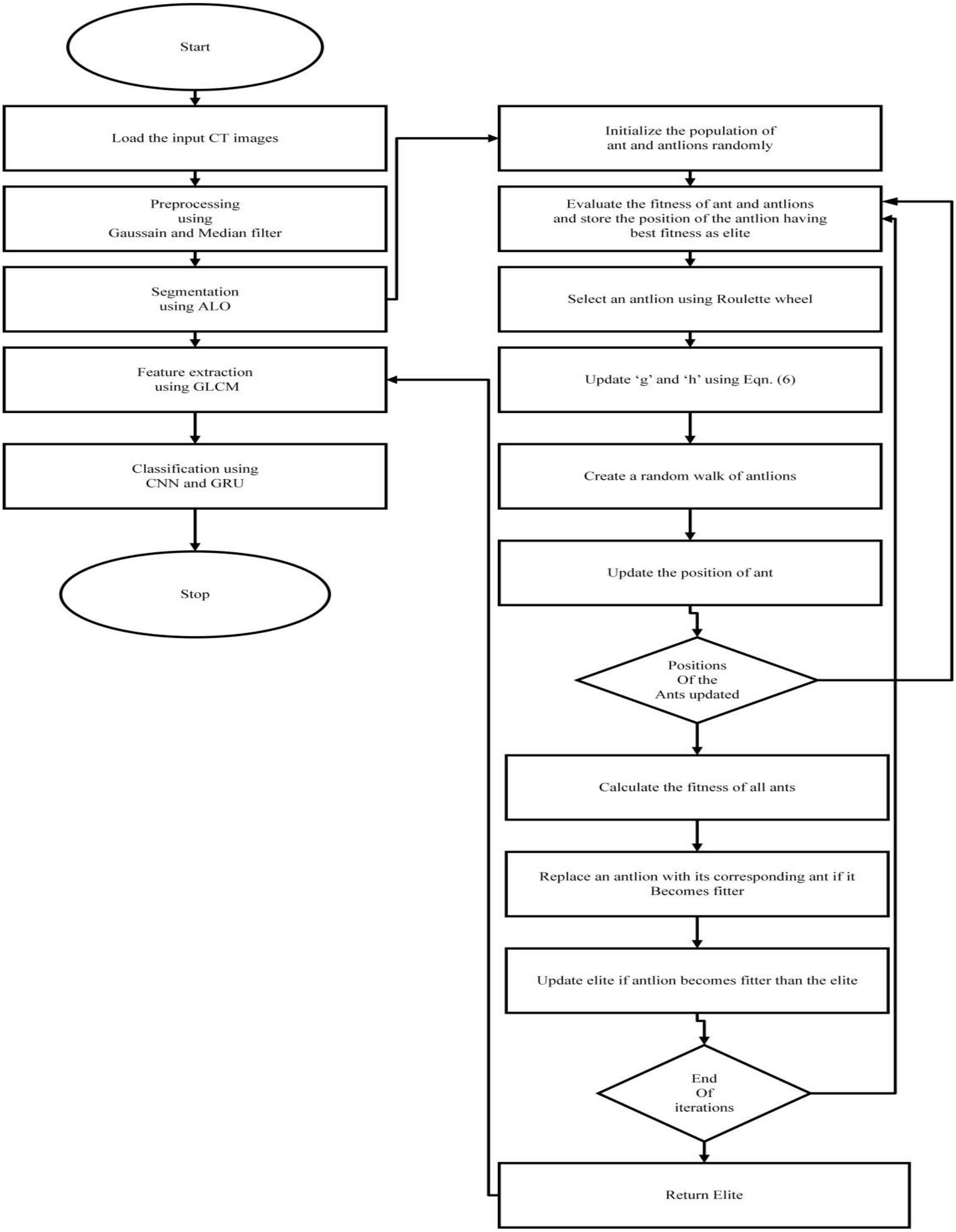

Start
Load the input CT images
Preprocessing using Gaussain and Median filter
Segmentation using ALO
Feature extraction using GLCM
Classification using CNN and GRU
Stop
Initialize the population of ant and antlions randomly
Evaluate the fitness of ant and antlions and store the position of the antlion having best fitness as elite
Select an antlion using Roulette wheel
Update 'g' and 'h' using Eqn. (6)
Create a random walk of antlions
Update the position of ant
Positions Of the Ants updated
Calculate the fitness of all ants
Replace an antlion with its corresponding ant if it Becomes fitter
Update elite if antlion becomes fitter than the elite
End Of iterations
Return Elite

Fig. 2. ALO-CNN-GRU technique flow chart.

However, when they get farther into the neural network, they may transfer these patterns there and attempt to find more distinguishing properties. For extracting image features, the CNN model is incredibly effective. Furthermore, the suggested CNNs model, according to the researchers, successfully distinguishes pancreatic CT scans. The pooling, convolutional layer (CLs), and fully connected layers make up the CNN's three major layers (FCs). Calculating the results of neurons connected to local sites is the responsibility of the CLs. By taking into account the region's and weights' dot product, it is decided. The common filters in the instance of the input images are made up of small area pixels. These filters can examine the images by swiping a window over them while automatically regulating any recurring patterns that show up in any image location. The stride is the separation in a series of filters. In the event that the initial parameter set is less than any one of the filter dimensions, they expand the convolution to incorporate screens that overlap. Each image's equation from the training set is given in Eq. (13)

$$p(c,q)=\frac{o(c,q)-\mu}{\sigma} \qquad (13)$$

a) Convolutional layer: The convolutional layer uses every layer to examine every image's complexity after collecting a range of input images. It is connected directly to the characteristics needed in the given image. It is expressed in Eq. (14)

$$f_i^n = x(\sum_{j\in N_i} f_j^{n-1} * p_{ji}^n + t_i^n) \qquad (14)$$

N_i – It stands for an input choice. The output is a bias that is additive. The kernel used for the map i, which if the map t and map s both sums over map i.

b) Max pooling layer: This layer is included in the down sampling layer to minimize fitting and the quantity of the neurons utilized. In addition to controlling overfitting, the pooling layer minimizes the size of the feature map, number of parameters, pace of computation, and the training time. It is calculated by using eqn. (15)

$$x_{mab} = max_{(c,d)\in} f_{ncd} \qquad (15)$$

Map, f_{ncd} is the element as (c, d) within the pooling region pts which represents a local neighbourhood around the place (t, s).

c) Fully connected layer: Fully Connected Layer has been employed for the representation classification context. Convolutional layers are put first, then FC layers. The output and input illustrations are mapped using the Fully Connected layer. The last layers of the network are completely coupled layers. The output of the max pooling layer serves as the input for the fully connected layer.

d) Softmax layer: The Softmax layer converts the values into a standardized proportion distribution. The classifier receives the output as an input. The Softmax classifier is a standard contribution classifier that applies the Softmax layer's structure to pancreatic cancer nodules. It is shown in Eq. (16)

$$\sigma(\vec{X})_a = \frac{e^{x_a}}{\sum_{i=1}^{n} e^{x_i}} \qquad (16)$$

2) Gated recurrent unit network: In order to solve the gradient vanishing problem, the GRU model was most frequently used in recurrent neural networks (RNN). GRU is more effective than LSTM since it has an internal cell state and three primary gates. The data is kept in a secure location within the GRU. The update gate offers both past and forward information, whereas the reset gate offers previous knowledge. The needed data from the former condition of the system is preserved and kept by the present memory gate using the reset gate. The input modulation gate allows for the introduction of nonlinearity while simultaneously giving the input zero-mean properties. The mathematical formulation of the basic GRU of rest and updated gates is, as defined by the following,

$$A_t = \sigma\,(Y_t.Z_{ya} + F_{t-1}.Z_{ha} + d_a) \qquad (17)$$

$$B_t = \sigma\,(Y_t.Z_{yb} + F_{t-1}.Z_{hb} + d_b) \qquad (18)$$

where Z_{ya} and Z_{yb} present weight parameters, while the d_a , $d_b$are biased. Fig. 3 represents the fundamental design of the GRU model.

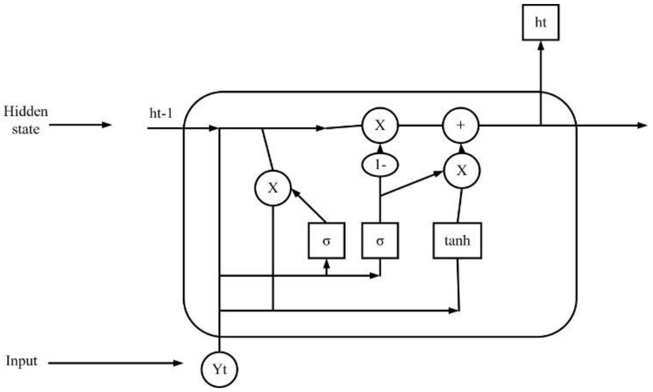


Fig. 3. The fundamental design of the GRU model.

3) Combined CNN-GRU: Four convolution layer (CL) layers, three max-pooling layers, and three fully linked layers made up the CNN-GRU model (FC). The activation function was included since it might not stimulate every neuron at once, which improves performance and speeds up learning. The raw image data were initially provided with the dimensions in CLs. Features must be extracted for the CNN-GRU model by going through Convolutional Layers. To reduce the nonlinearity dimension, ReLUs were combined with Convolutional Layers. The parameter for the training dataset is likewise decreased by the pooling layer. After the pooling layer, the training variables were transferred over from the hidden layer to avoid overfitting issues in the system. Fig. 4 represents the overall structure of the CNN-GRU model.

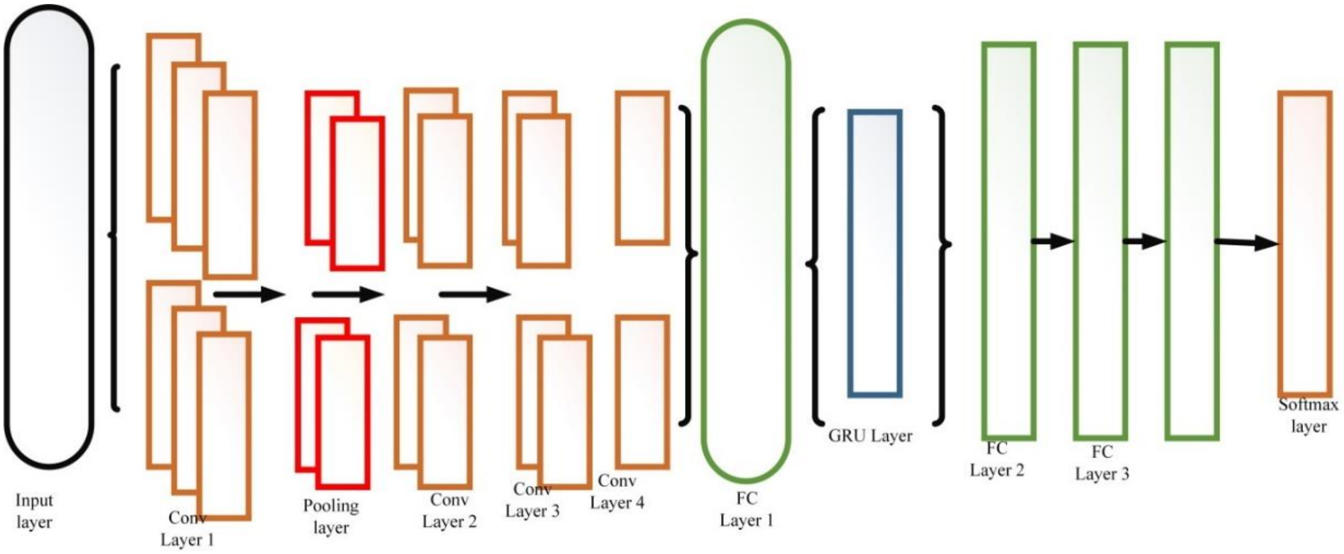


Fig. 4. Overall structure of the CNN-GRU model.

IV. Results and Discussion

A dataset of images from computerized tomography is used to evaluate the suggested strategy. The 1782 CT images were pre-processed utilizing a hybrid Gaussian and median filter. CT image pre-processing is essential for enhancing the image's visual appeal before additional processing. The images in the collection often contain noise; the noise is removed by pre-processing. After that, the segmentation process is completed using the Antlion optimization method. The areas of pancreatic cancer are identified and separated throughout this procedure. The Gray Level Co-occurrence Matrix then does the feature extraction. The combined CNN-GRU performs the classification after that. The effectiveness of the offered technique is assessed using performance indicators such as Recall, Accuracy, F-measure, and Precision.

A. Accuracy

The model of the system accuracy is a measure of how precisely it functions across all classes. Generally, it is the statement that all observations are accurately expected observations. Accuracy is expressed in Eq. (19),

$$Accuracy = \frac{Tpos+Tneg}{Tpos+Tneg+Fpos+Fneg} \quad (19)$$

TABLE I. Comparison of Accuracy

Methods	Accuracy (%)
IDLDMS	99.35
Multi resolution CNN	89.67
CE-ConvLSTM	73.6
ODL-PTNTC	98.82
Proposed ALO-CNN-GRU	99.92

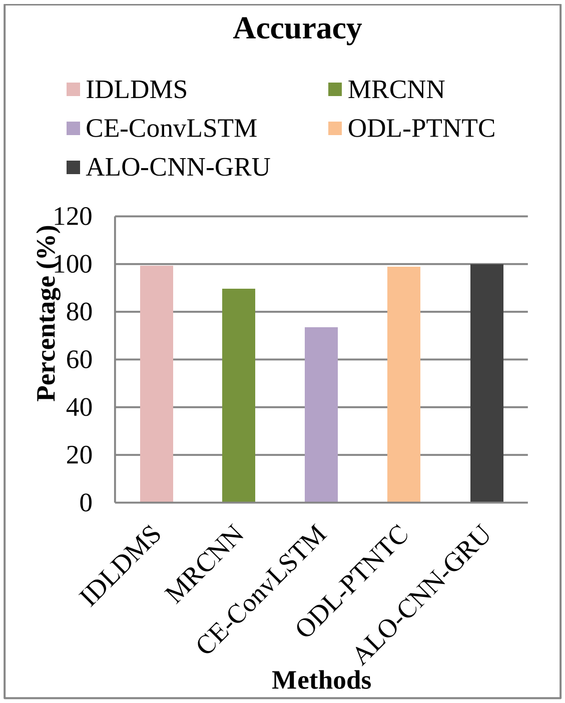


Fig. 5. Comparison of accuracy.

When compared to the current Pancreatic cancer segmentation and classification techniques like IDLDMS, MRCNN, CE-ConvLSTM, ODL-PTNTC, which are shown in Table I, the projected methodology ALO-CNN-GRU obtains superior accuracy. The accuracy of ALO-CNN-GRU and other approaches is compared in Fig. 5.

B. Precision

Precision is determined by counting the precise positive ratings that differ from the total positive evaluations. By using eqn. (20), it is possible to determine the accurate identification of cancer nodules in the afflicted area.

$$P = \frac{Tpos}{Tpos+Fpos} \qquad (20)$$

TABLE II. COMPARISON OF PRECISION

Methods	Precision (%)
IDLDMS	99.35
Multi resolution CNN	91.37
CE-ConvLSTM	81.3
ODL-PTNTC	98.73
Proposed ALO-CNN-GRU	99.64
AAT	95.2
LLE	96.55

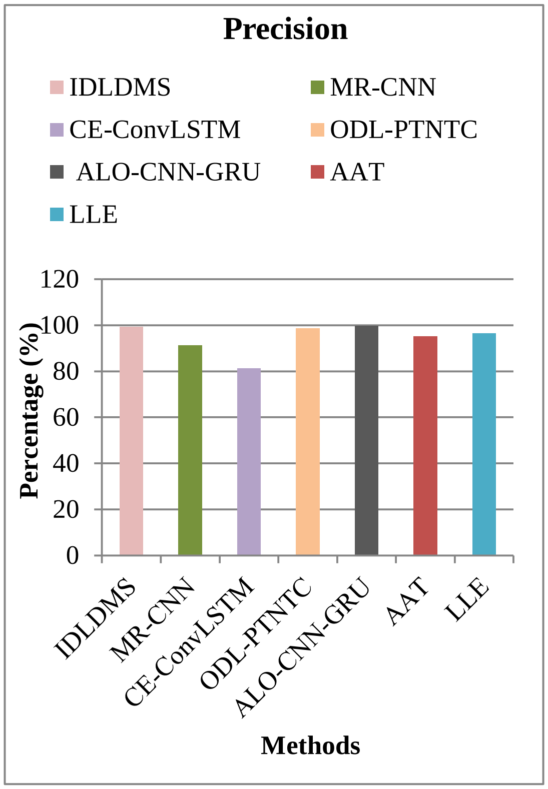


Fig. 6. Comparison of precision.

The precision comparison between ALO-CNN-GRU and other methods is shown in Fig. 6. Table II shows that the proposed Antlion Optimization-Convolutional Neural Network-Gated Recurrent Unit strategy outperforms previous pancreatic cancer segmentation and classification methods including IDLDMS, MR-CNN, CE-ConvLSTM, ODL-PTNTC, AAT, and LLE with a greater precision of 99.64%.

C. Recall

The recall is the ratio of the total amount of positive sample to the number of actual positives that were accurately classified as positives. It displays the percentage of prediction about the identification of tumor nodules indicated by Eq. (21) were correct.

$$R = \frac{Tpos}{Tpos+Fneg} \qquad (21)$$

TABLE III. COMPARISON OF RECALL

Methods	Recall (%)
IDLDMS	98.84
Multi resolution CNN	93.63
CE-ConvLSTM	65.9
ODL-PTNTC	98.73
Proposed ALO-CNN-GRU	99.53
AAT	95.8

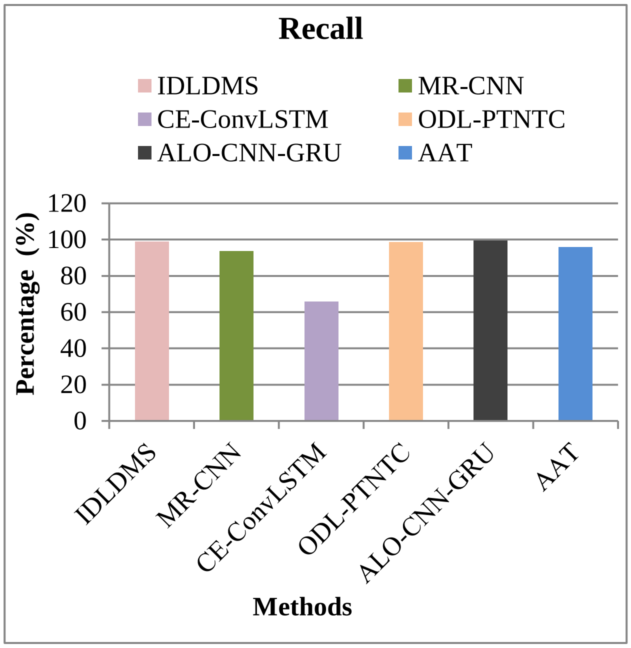


Fig. 7. Comparison of recall.

The suggested methodology ALO-CNN-GRU greatly improves recall comparing to the current Pancreatic cancer segmentation and classification techniques such IDLDMS, MR-CNN, CE-ConvLSTM, ODL-PTNTC, and AAT, as shown in Table III. Fig. 7 displays the recall comparison between ALO-CNN-GRU and other methods.

D. F1-Score

Recall and precision are combined in the F1-Score computation. The F1-Score is computed using recall and precision that is represented in Eq. (22),

$$F1 - score = \frac{2\times precision \times recall}{precision\times recall} \qquad (22)$$

TABLE IV. COMPARISON OF F1-SCORE

Methods	F1-Score (%)
IDLDMS	99.48
Multi resolution CNN	84.97
CE-ConvLSTM	70.5
ODL-PTNTC	98.82
Proposed ALO-CNN-GRU	99.72

Fig. 8 illustrates a comparison of the ALO-CNN-GRU and other methodologies' F1-Scores. Table IV demonstrates that the proposed methodology, Antlion Optimization-Convolutional Neural Network-Gated Recurrent Unit technique, achieves a superior F1-Score in comparison to the current pancreatic cancer segmentation and classification approaches, such as IDLDMS, MR-CNN, CE-ConvLSTM, and ODL-PTNTC.

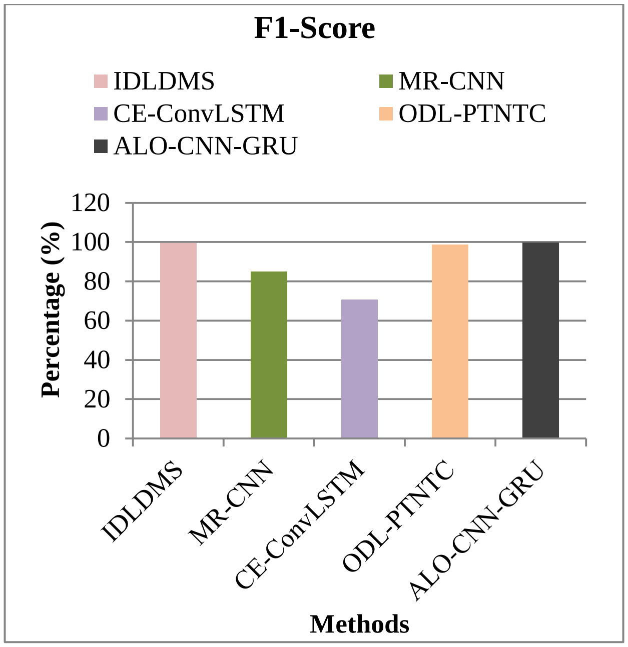


Fig. 8. Comparison of F1-score.

V. CONCLUSION

An essential part of the human body, the pancreas performs both internal and exterior secretion duties and is prone to a number of illnesses. Pancreatic malignancies, which are deadly in nature, currently have very poor prognoses. Computerized pancreatic cancer segmentation and classification utilizing a computer-aided diagnostic model is needed to monitor, identify, and categorize the incidence of cancer. Accurate image interpretation can be provided throughout deep learning algorithms for therapeutic application. To achieve this goal, the study created an ALO-CNN-GRU model for CT images and deep learning-based pancreatic tumor segmentation and classification. To remove noise from the acquired dataset, the images go through pre-processing. For the pre-processing, a hybrid Gaussian and median filter is being used. The segmentation is performed using the Antlion optimization algorithm to determine the impacted pancreatic region. Next, the classifiers of the Convolutional neural network and Gated Recurrent Unit networks are used to categorize pancreatic cancer as benign and malignant. We examined recall, accuracy, precision, and F1-score as performance indicators. Better performance metrics are produced by the suggested ALO-CNN-GRU technique. The experimental results back up the claim that the suggested strategy works better than approaches already in use. In order to accurately detect and categorize the pancreatic cancer locations, future versions of the methods will employ LSTM with additional deep learning mechanisms and combine sophisticated optimization with classification algorithms.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Saudi Electronic University for logistic support for this work through the 2nd Interdisciplinary Scientific Research Hackathon, project no. (SRH005T1).

REFERENCES

[1] S.-L. Liu et al., “Establishment and application of an artificial intelligence diagnosis system for pancreatic cancer with a faster region-based convolutional neural network,” Chin. Med. J. (Engl.), vol. 132, no. 23, pp. 2795–2803, Dec. 2019, doi: 10.1097/CM9.0000000000000544.

[2] M. Reyngold, P. Parikh, and C. H. Crane, “Ablative radiation therapy for locally advanced pancreatic cancer: techniques and results,” Radiat. Oncol., vol. 14, no. 1, p. 95, Dec. 2019, doi: 10.1186/s13014-019-1309-x.

[3] K.-L. Liu et al., “Deep learning to distinguish pancreatic cancer tissue from non-cancerous pancreatic tissue: a retrospective study with cross-racial external validation,” Lancet Digit. Health, vol. 2, no. 6, pp. e303–e313, Jun. 2020, doi: 10.1016/S2589-7500(20)30078-9.

[4] I. Joo et al., “Preoperative CT Classification of the Resectability of Pancreatic Cancer: Interobserver Agreement,” Radiology, vol. 293, no. 2, pp. 343–349, Nov. 2019, doi: 10.1148/radiol.2019190422.

[5] Y. Iwatate et al., “Radiogenomics for predicting p53 status, PD-L1 expression, and prognosis with machine learning in pancreatic cancer,” Br. J. Cancer, vol. 123, no. 8, pp. 1253–1261, Oct. 2020, doi: 10.1038/s41416-020-0997-1.

[6] H. Nasief et al., “A machine learning based delta-radiomics process for early prediction of treatment response of pancreatic cancer,” Npj Precis. Oncol., vol. 3, no. 1, p. 25, Dec. 2019, doi: 10.1038/s41698-019-0096-z.

[7] Y. Toyama, M. Hotta, F. Motoi, K. Takanami, R. Minamimoto, and K. Takase, “Prognostic value of FDG-PET radiomics with machine learning in pancreatic cancer,” Sci. Rep., vol. 10, no. 1, p. 17024, Oct. 2020, doi: 10.1038/s41598-020-73237-3.

[8] O. M. Griffin, S. N. Duggan, R. Ryan, R. McDermott, J. Geoghegan, and K. C. Conlon, “Characterising the impact of body composition change during neoadjuvant chemotherapy for pancreatic cancer,” Pancreatology, vol. 19, no. 6, pp. 850–857, Sep. 2019, doi: 10.1016/j.pan.2019.07.039.

[9] S. Srisajjakul, P. Prapaisilp, and S. Bangchokdee, “CT and MR features that can help to differentiate between focal chronic pancreatitis and pancreatic cancer,” Radiol. Med. (Torino), vol. 125, no. 4, pp. 356–364, Apr. 2020, doi: 10.1007/s11547-019-01132-7.

[10] A. D. Singhi, E. J. Koay, S. T. Chari, and A. Maitra, “Early Detection of Pancreatic Cancer: Opportunities and Challenges,” Gastroenterology, vol. 156, no. 7, pp. 2024–2040, May 2019, doi: 10.1053/j.gastro.2019.01.259.

[11] L. Boldrini, D. Cusumano, F. Cellini, L. Azario, G. C. Mattiucci, and V. Valentini, “Online adaptive magnetic resonance guided radiotherapy for pancreatic cancer: state of the art, pearls and pitfalls,” Radiat. Oncol., vol. 14, no. 1, p. 71, Dec. 2019, doi: 10.1186/s13014-019-1275-3.

[12] S. Rudra et al., "Using adaptive magnetic resonance image-guided radiation therapy for treatment of inoperable pancreatic cancer," Cancer Med., vol. 8, no. 5, pp. 2123–2132, 2019, doi: 10.1002/cam4.2100.

[13] S.-L. Liu et al., "Establishment and application of an artificial intelligence diagnosis system for pancreatic cancer with a faster region-based convolutional neural network," Chin. Med. J. (Engl.), vol. 132, no. 23, pp. 2795–2803, Dec. 2019, doi: 10.1097/CM9.0000000000000544.

[14] J. Li, L. Qi, Q. Chen, Y.-D. Zhang, and X. Qian, "A dual meta-learning framework based on idle data for enhancing segmentation of pancreatic cancer," Med. Image Anal., vol. 78, p. 102342, May 2022, doi: 10.1016/j.media.2021.102342.

[15] T. Vaiyapuri et al., "Intelligent Deep-Learning-Enabled Decision-Making Medical System for Pancreatic Tumor Classification on CT Images," Healthcare, vol. 10, no. 4, Art. No. 4, Apr. 2022, doi: 10.3390/healthcare10040677.

[16] Y. Xia et al., "Effective Pancreatic Cancer Screening on Non-contrast CT Scans via Anatomy-Aware Transformers," in Medical Image Computing and Computer Assisted Intervention – MICCAI 2021, vol. 12905, M. de Bruijne, P. C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, and C. Essert, Eds. Cham: Springer International Publishing, 2021, pp. 259–269. doi: 10.1007/978-3-030-87240-3_25.

[17] X. Yang, Y. Chen, X. Yue, C. Ma, and P. Yang, "Local linear embedding based interpolation neural network in pancreatic tumor segmentation," Appl. Intell., vol. 52, no. 8, pp. 8746–8756, Jun. 2022, doi: 10.1007/s10489-021-02847-9.

[18] V. Asadpour, R. A. Parker, P. R. Mayock, S. E. Sampson, W. Chen, and B. Wu, "Pancreatic cancer tumor analysis in CT images using patch-based multi-resolution convolutional neural network," Biomed. Signal Process. Control, vol. 68, p. 102652, Jul. 2021, doi: 10.1016/j.bspc.2021.102652.

[19] J. Yao, Y. Shi, L. Lu, J. Xiao, and L. Zhang, "DeepPrognosis: Preoperative Prediction of Pancreatic Cancer Survival and Surgical Margin via Contrast-Enhanced CT Imaging." arXiv, Aug. 26, 2020. Accessed: Dec. 13, 2022. [Online]. Available: http://arxiv.org/abs/2008.11853

[20] M. M. Althobaiti, A. Almulihi, A. A. Ashour, R. F. Mansour, and D. Gupta, "Design of Optimal Deep Learning-Based Pancreatic Tumor and Nontumor Classification Model Using Computed Tomography Scans," J. Healthc. Eng., vol. 2022, pp. 1–15, Jan. 2022, doi: 10.1155/2022/2872461.

A New Task Scheduling Framework for Internet of Things based on Agile VNFs On-demand Service Model and Deep Reinforcement Learning Method

Li YANG
Department of Electronic Information Engineering, Leshan Vocational and Technical College,
Leshan 614099, Sichuan, China

***Abstract*—Recent innovations in the Internet of Things (IoT) have given rise to IoT applications that require quick response times and low latency. Fog computing has proven to be an effective platform for handling IoT applications. It is a significant challenge to deploy fog computing resources effectively because of the heterogeneity of IoT tasks and their delay sensitivity. To take advantage of idle resources in IoT devices, this paper presents an edge computing concept that offloads edge tasks to nearby IoT devices. The IoT-assisted edge computing should meet two conditions, edge services should exploit the computing resources of IoT devices effectively and edge tasks offloaded to IoT devices do not interfere with local IoT tasks. Two main phases are included in the proposed method: virtualization of edge nodes, and task scheduling based on deep reinforcement learning. The first phase offers a layered edge framework. In the second phase, we applied deep reinforcement learning (DRL) to schedule tasks taking into account the diversity of tasks and the heterogeneity of available resources. According to simulation results, our proposed task scheduling method achieves higher levels of task satisfaction and success than existing methods.**

Keywords*—*Internet of things; task scheduling; edge computing; resource allocation

I. Introduction

The recent rapid development of artificial intelligence [1, 2], machine learning [3], optical networks [4, 5], smart grids [6], cloud computing [7, 8], 5G connectivity [9], Blockchain, and Internet of Things (IoT) [10, 11] is leading to an exponential growth in data usage across a wide range of engineering and commerce disciplines. Over the last decade, the IoT has been recognized for its remarkable potential in computer science. It brings out an environment in which many intelligent objects with limited resources can interact with each other through different technologies [12]. The IoT tries to link the physical and virtual worlds by equipping physical devices with processing, networking, detection, and identification functions. The IoT objects are employed in different applications, such as vehicle networks, energy management, traffic control, medical treatment, and healthcare, aiming to gather information about the physical world [13, 14]. In this regard, to obtain valuable information and fulfil the assigned tasks, a massive amount of data is produced that brings challenging problems for efficient information processing, especially for the specific scenarios requiring real-time data handling. Considering edge computing capability in real-time processing, the computing IoT tasks can be offloaded to edge devices for implementation [15, 16].

Moreover, since the IoT services and applications are increasing daily, a practical approach to serving the growing needs in different application domains becomes vital [17, 18]. To address the mentioned problems, achieve high resource utilization, reduce communication costs, and improve the lifetime of IoT networks, task scheduling methods can have significant impacts [19, 20]. These methods aim to schedule the existing tasks in a suitable sequence to accomplish tasks under problem-specific constraints, such as communication costs among IoT objects, resource utilization, and the operational lifetime of sensor nodes [21].

Nowadays, Network Function Virtualization (NFV) is known as a superior technology in telecommunication networks. It refers to centralizing and virtualizing network functions that can be run in data centres on standard and commercial off-the-shelf (COTS) hardware instead of distributed and proprietary hardware [22]. Using NFV, network device purchases and related maintenance costs can be reduced effectively. As a logical consequence of NFV, Virtual Network Functions (VNFs) take advantage of middleware to virtualize network functions. The deployment of VNFs on commodity hardware reduces the need for dedicated hardware devices to perform individual network functions. Our main objective is to improve edge computing operational efficiency by implementing the agile VNF on-demand model and the deep reinforcement learning-based task scheduling method. The proposed method includes two main phases, virtualization of edge nodes and task scheduling based on the deep reinforcement learning method. In the first step, a layered edge framework is presented. In the second step, we employed deep reinforcement learning (DRL) to solve task scheduling problems considering the tasks' diversity and the heterogeneity of available resources. This article makes the following major contribution.

- We analyze the optimization of time scheduling and the assignment of virtual machines in edge computing using model-free DRL-based task scheduling. VM availability, task characteristics, and queue dynamics are considered in the formulation of the problem as an MDP problem.

- In this case, the action is represented as a pair of VMs and tasks, whose dimension may be extremely large. In the MDP formulation, a new mechanism is designed to decouple the scheduling time step from the real-time step. With this mechanism, the action space remains linear with the product of the number of VMs and the queue size, and multiple tasks can be scheduled simultaneously.

The rest of the paper in organized as follows. Related works are reviewed in Section II. The proposed task scheduling method is described in Section III. Section IV reports the simulation results. Finally, the conclusion is presented in Section V.

II. Related Work

Al-Habob, Dobre [23] considers offloading multiple tasks concurrently to several mobile edge computing servers. Offloading latency and failure probability are minimized by scheduling interdependent subtasks for servers. Models based on conflict graphs and genetic algorithms are used to solve scheduling problems. Experimental findings demonstrate that these algorithms approach optimal solutions found by exhaustive searches. In addition, even though parallel offloading employs orthogonal channels, sequential offloading is more likely to fail than parallel offloading. In contrast, parallel offloading provides lower latency. Latency gaps between parallel and sequential schemes decrease as the dependency between sub-tasks increases.

Chen, Guo [24] examine the business processes of edge cloud task scheduling. The authors propose an algorithm to optimize resource-constrained task scheduling using the construction of profit matrices, classification, and clustering preprocessing. Tasks with similar characteristics are grouped and categorized using clustering preprocessing. On the basis of the constructed profit matrix, a resource-constrained task scheduling strategy is derived. In the second step, edge cloud components such as virtual machines, user requests, tasks, and resources are constructed using Petri nets. As a final step, the suggested mechanism is evaluated through several experiments. According to simulations, the algorithm maximizes profit while managing tasks efficiently, reliably, and with a high load balance.

A mathematical formulation of the fog node task scheduling problem is presented by Azizi, Shojafar [25] to reduce fog node energy consumption while maintaining IoT QoS requirements. The goal of their model was to reduce deadline violation time. The authors proposed two partially greedy algorithms, semi-greedy with priority awareness and multi-start semi-greedy with priority awareness, to map IoT tasks to fog nodes efficiently. The proposed approaches were evaluated in terms of system lifespan, latency, energy, and the IoT task completion rate. In comparison with existing algorithms, tests indicate that the introduced algorithms improve task deadline compliance and reduce the total time spent violating deadlines.

Kanbar and Faraj [26] developed the Region Aware Dynamic Scheduling (RADISH) model, which consists of five consecutive processes. To reduce latency in scheduling, they implemented a bi-class neural network based on task nature to classify incoming tasks based on their nature, taking into account login credentials, emails, passwords, types of services, and quality of service parameters. In the second process, the improved moth flame optimization is used to schedule classified tasks considering workload, deadline, priority, and energy. Third, load balancing is accomplished by clustering potential fields. With the aim of balancing server load and improving efficiency, three repository systems have been implemented. The final step involves introducing the Hopcroft-Karp algorithm that takes into account the VM state and minimizes allocation times and enhances the quality of service.

Hybrid Flamingo Search with a Genetic Algorithm (HFSGA) is implemented by Hussain and Begh [27] to optimize task scheduling for cost minimization. HFSGA and other well-known optimization algorithms are compared on seven essential benchmark optimization functions. Furthermore, Friedman Rank Tests are conducted to ascertain the results' significance. Implementing the model produces better results regarding task completion percentages, makespan, and costs. This work shows better results than existing algorithms such as round-robin, genetic, PSO, and ACO.

In order to secure the allocation of tasks on cloud and fog nodes, Najafizadeh, Salajegheh [28] proposed a multi-objective simulated annealing algorithm. A compromise solution is found by applying the goal-programming approach. In addition, a new goal called client-driven access level and schedule is created in regard to distributing tasks among fog and cloud nodes. The proposed algorithm was found to be 50% more efficient regarding deadlines, 88% more efficient regarding control levels, and 10% more efficient regarding service delays in comparison to moth-flame optimization, tabu search, and PSO algorithms.

Task scheduling in edge computing requires the consideration of two special problems: time scheduling and resource allocation. The task execution order is determined by time scheduling, while resource allocation is responsible for allocating tasks to suitable virtual machines (VMs) for execution. In the field of edge computing, a number of scheduling issues have been explored [29-33]. However, the majority of existing works focus on resource allocation, while time scheduling has received little attention. Tan, Han [34] proposed a general model to minimize task response times when tasks are offloaded to edge servers. Zhang, Du [35] proposed a scheduling algorithm based on Lyapunov optimization in order to minimize the communication delay and the computing delay. Chen, Thomas [36] developed a dual-scheduling framework to accommodate the unstable capacity of servers and task arrival rates in heterogeneous vehicular edge computing. In [37], a mixed integer nonlinear programming (MINLP) algorithm was employed for data-parallel offloading and scheduling of computationally-intensive data-parallel tasks in order to minimize the average completion time. According to [38], tasks with the lowest delay are scheduled first using the shortest-job-first (SJF) scheduling method.

Alameddine, Sharafeddine [39] explored the use of device-to-device collaboration for task offloading by taking into consideration the mobility of humans in order to optimize the task assignment and power management. The issue of energy-efficient task scheduling for IoT edge computing has been addressed in [20] by a heuristic algorithm. In these methods, ideal mathematical models are used and optimization is achieved through mixed-integer nonlinear programming (MINLP) or heuristic algorithms. In spite of the fact that these model-oriented algorithms can produce excellent results, they are not well suited to dynamic environments in which task arrival rates and popularity are unknown in advance. It is also important to note that the model-based task scheduling algorithms are largely concerned with optimizing one-step instead of pursuing long-term objectives. In these algorithms, the availability of resources is assumed to be fixed during the scheduling period.

III. Proposed Task Scheduling Method

In our proposed mechanism, each gateway node serves as a connection point for several IoT nodes connected to the gateway node based on their shortest distance. Our main aim is to propose an edge computing service model based on gateways to maximize IoT resource utilization, accelerate the processing of users' service requests, and enhance edge computing efficiency. The proposed model involves user requests passing through the edge gateway, which determines whether to process tasks. To decrease data processing time, the controller forwards service requests to the cloud if the edge gateway is unable to process them. The edge computing service model contains three main components, lightweight VNF configuration, scheduler, and resource estimation. According to the proposed model, when a request for a service is received, it is checked whether or not an adequate amount of computing resources is available to fulfil such demand. In this regard, one of the following events may occur. Depending on the outcome of the task scheduling algorithm, the edge gateway processes the requests through the scheduler and queues them in the system. On the other hand, requests are transferred directly to the cloud if the edge gateway lacks the resources.

Improving edge gateway operational efficiency is the main aim of task scheduling methods. Since the service requests are different from each other, a task scheduling algorithm determines how to meet the demands of each request. A scheduling method reduces time spent on tackling high-demand tasks as a primary objective. Our main objective is implementing on-demand models for agile VNFs and deep reinforcement learning-based task scheduling methods to enhance edge computing efficiency. The proposed method includes two main phases, virtualization of edge nodes and task scheduling based on the deep reinforcement learning method.

In order to simplify the task scheduling process, we will only focus on computational resources. In order to make scheduling decisions, the scheduler monitors the status information of incoming tasks and virtual machines (VMs), including the task sizes, the expected completion time, the computing speed (in million instructions per second (MIPS)), and the waiting time. The scheduler determines when to schedule (e.g., the scheduling order and the start time for each task) and where to schedule (e.g., which VM is assigned to each task) based on the observations. In order to schedule the tasks, they are divided into two sets: a waiting set and a backlog queue. A task in the waiting set occupies a waiting slot that can be observed fully, whereas the scheduler can only observe the number of tasks in the backlog queue. In each scheduling time step, the scheduler selects at most one task from the waiting slot for scheduling. In this study, we examine the scheduling of tasks in edge computing when only one edge server is deployed. It is the objective to maximize the long-term task satisfaction of all tasks, which is achieved by:

$$\boldsymbol{max}\sum_{\boldsymbol{t=1}}^{\boldsymbol{T}}\sum_{\boldsymbol{i\in J,j\in V}}\boldsymbol{g_{i,j}} \qquad (1)$$

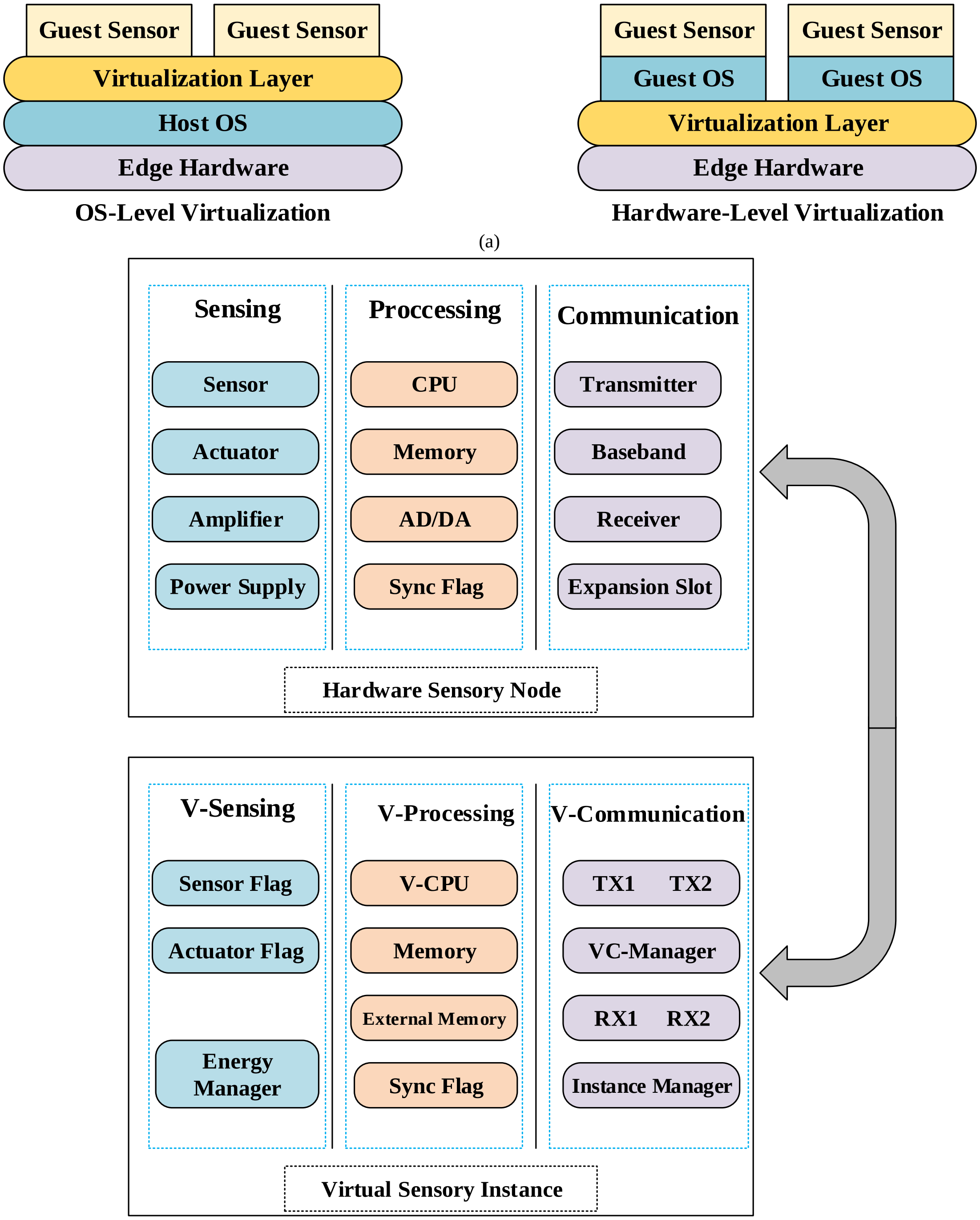
where $g_{i,j}$ is the task satisfaction of the task i scheduled to VM j.

A. *First Phase: Virtualization of Edge Nodes*

This step proposes a virtualized edge framework to support Cloud-to-Things applications at all layers. The virtualization process involves three stages: virtualizing objects, network functions, and services. The implementation of object virtualization allows physical sensors to gain IP capability without compromising their unique functionalities. In order to represent heterogeneous physical entities, we need to create a unified software Virtual Object (VO) on edge nodes. An abstract VO is capable of interacting over the Internet with various hosts.

Additionally, it can act as a close neighbour to physical objects through wireless or wired connections available at the edge. VOs provide semantic descriptions of actual objects. Although physical objects have heterogeneous functions, they generally have limited memory and components such as communication modules, sensing modules, and power supply units. These components can be expanded and installed on edge devices for real physical objects as virtual software instances.

As illustrated in Fig. 1(a), two critical streams of object virtualization are taken into account to illustrate VO-hosting solutions: hardware-level and OS-level virtualization. VOs are compatible with established operating systems. Edge players, such as providers, developers, and end users, can develop dedicated hosting platforms with advanced features, such as memory, hardware interface, and CPU, in order to achieve better performance. Fig. 1(b) provides more details about the sensor virtualization framework. Both virtual sensor instructions and physical sensor data are stored in the "Sync Flag". Version numbers are used to synchronize actual and virtual objects. "Energy Manager" indicates the battery life of physical sensors and turns them on. Compared to physical sensors, virtual sensors are composed of V-communication, V-processing, and V-sensing components. In the left column, "Actuator Flag" contains the instructions given to the physical sensor, and "Sensor Flag" holds the sensor's collected data. V-processing is the equivalent of physical sensor processing. A predefined local or network storage device can be used as an external memory.



(a)

(b)

Fig. 1. Object virtualization scheme.

To manage virtual and physical nodes as well as VNFs in the proposed method, an Object Virtualization Manager (OVM) is required. As depicted in Fig. 2, the OVM manages and orchestrates physical and virtual entities. Initially, the OVM deploys and programs the corresponding VOs, monitors and coordinates their operation, discovers and registers them, and creates and terminates them. In this regard, the OVM stores the configuration files of VOs editable by remote or local users. In order to perform automatic self-configuration for service deployment, the OVM downloads the configuration profiles of registered VOs. A VO will be migrated to a more resourceful container if its container is overloaded. VNFs, as software instances, contain several portions of VMs that run network functions on standard hardware. In order to decompose the VNF into reusable components, which can be designed as executable microservices and optimized, upgraded, and configured independently, network functions typically involve several approaches, including verification, computing, media access control, coding, and signaling. These components combine to form VNFs in micro containers.

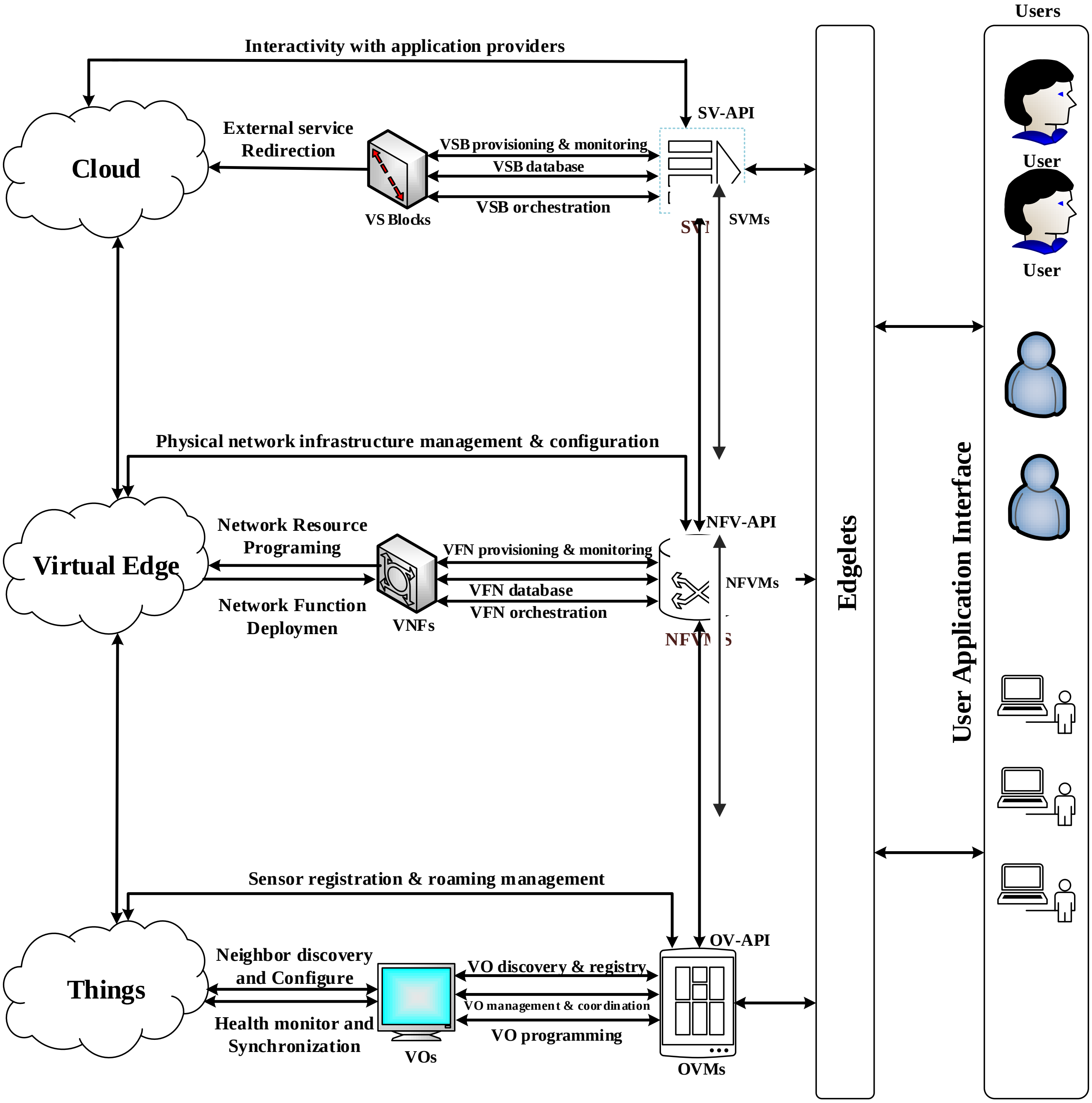

Users
Interactivity with application providers
SV-API
Cloud
External service Redirection
VSB provisioning & monitoring
VSB database
VSB orchestration
VS Blocks
SVMs
SVI
User
User
Physical network infrastructure management & configuration
NFV-API
Virtual Edge
Network Resource Programing
Network Function Deploymen
VFN provisioning & monitoring
VFN database
VFN orchestration
VNFs
NFVMs
NFVMS
Edgelets
User Application Interface
Sensor registration & roaming management
OV-API
Things
Neighbor discovery and Configure
Health monitor and Synchronization
VO discovery & registry
VO management & coordination
VO programming
VOs
OVMs

Fig. 2. Virtualization of edge computing framework for IoT environment.

B. *Second Phase: Task Scheduling based on Deep Reinforcement Learning*

Scheduling tasks in edge computing involves two main concerns, allocation of resources and schedule. Resource allocation involves assigning tasks to the appropriate virtual machines, and time scheduling determines the order of task execution. To maximize the quality of experience, our task scheduling strategy considers the expected delay requirement for heterogeneous virtual machine resources. The level of task satisfaction acts as a reward in the deep reinforcement learning algorithm. Network edge servers handle computationally-intensive tasks owing to the difficulty of performing them on local devices. The edge servers are configured with various virtual machines that differ in terms of computational capacity and execution time. The scheduler is responsible for monitoring the status of virtual machines and incoming tasks. The waiting time, the speed of computing, the expected completion time, and task size are significant factors in scheduling decision-making. Observations confirm that the scheduler decides the scheduling order and the start time of each task, determining suitable virtual machines. Fig. 3 shows the general layout of the proposed task-scheduling framework.

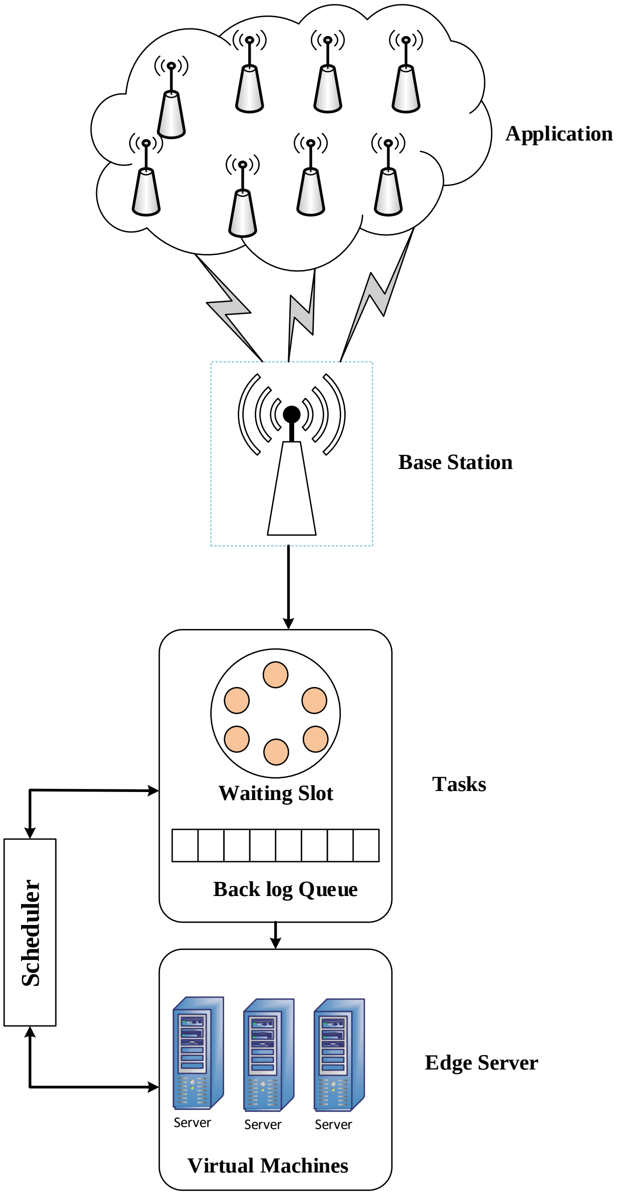


Fig. 3. Proposed task scheduling framework.

IV. Experimental Results

A numerical evaluation of the proposed task scheduling mechanism is presented in this section. Pytorch running on Python 3 produced all simulation results. A comparison was made between the proposed mechanism and two baselines. We implemented the task scheduling policy using a four-layer DNN structure. Each hidden layer encompasses 64 neurons with a rectified linear unit (ReLU) as an activation function. The output layer consists of (M+1)×O neurons, where M refers to the number of VMs and O denotes the maximum tasks in the waiting slot. In training, we set the discount factor to 0.99, which indicates that the current decision is affected by future steps. The learning rate is set to 10−4, and gradient descent is performed using the Adam optimizer. The paper focuses on scheduling tasks in the edge system by considering only computational resources. VM resources and task characteristics are the only environment parameters considered in calculating residual computation delay. IoT devices generate tasks and send them to the base station for transmission. The edge server receives these tasks periodically. Latency is expected to range from 5 to 10 seconds, with transmission delays ranging from 1 to 5 seconds. Task sizes range from 500 to 4000 MI. Virtual machines have a processing capacity ranging from 1000 to 2000 MIPS. Waiting slots are set to O=5 and backlog queues to |b|=5.

The effects of task popularity skewness, task arrival rate, and virtual machine count on task satisfaction and success ratio were studied. Fig. 4 and Fig. 5 illustrate the excremental outcomes. As illustrated in Fig. 4, the number of virtual machines and the task arrival rate are associated with the cumulative degree of task satisfaction. With the popularity skewness set at 0.3, the task arrival rate ranges from 3 to 7, and the number of virtual machines increases from 3 to 5. According to Fig. 5, the cumulative task satisfaction degree decreases as the task arrival rate increases. A higher arrival rate means that more tasks are waiting for scheduling in the edge system simultaneously, which lengthens their waiting time. As the number of VMs increases, the average task satisfaction degree also increases. Tasks can be scheduled across multiple virtual machines, resulting in shorter waiting times. Fig. 5 shows the skewness of task popularity versus task satisfaction degree. Task popularity skewness ranges from 0.1 to 0.9 with three virtual machines. A higher cumulative task satisfaction is associated with greater skewness in task popularity. Each type of task has a different popularity skewness. As task popularity skewness increases, smaller tasks become more popular while larger tasks become less popular, thus reducing the overall waiting time.

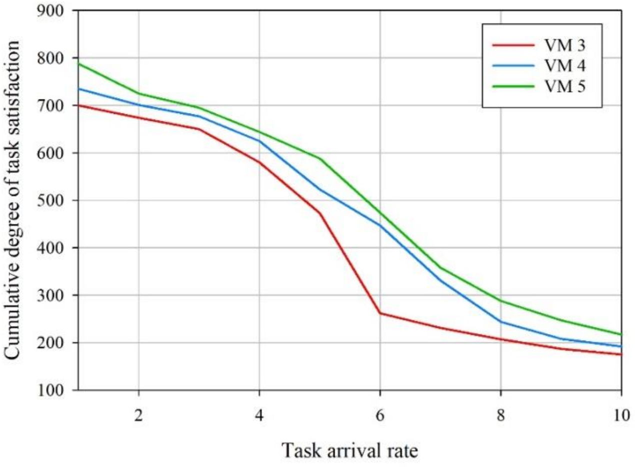


Fig. 4. Task satisfaction degree vs. task arrival rate.

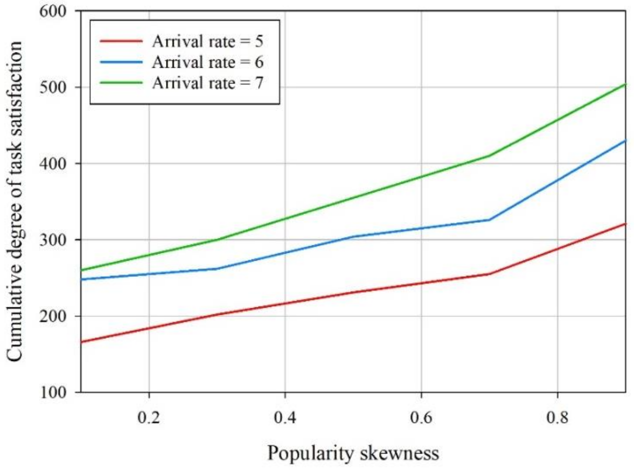


Fig. 5. Task satisfaction degree vs. popularity skewness.

Shortest Job First (SJF) [38] and First Come First Service (FCFS) [37] algorithms are chosen as benchmarks for evaluating the performance of the proposed task scheduling mechanism. SJF and FCFS assign the scheduled task to virtual machines with the highest instant reward. This leads to the scheduling of tasks in a greedy manner. These benchmarks can therefore be categorized into greedy-FCFS and greedy-SFJ. A comparison was made between our proposed method and these benchmarks in terms of task success rate and task satisfaction rate. Total task satisfaction is affected by the average task satisfaction degree, which makes it possible to evaluate the algorithm's overall quality. The task is considered complete when the response time is less than the expected delay. According to Eq. (1), the task success ratio can be obtained by dividing the number of satisfied tasks by the number of tasks in total.

$$\boldsymbol{\varepsilon_s} = \frac{\boldsymbol{N_s}}{\sum_{j\epsilon J} \boldsymbol{N_T}}. \tag{1}$$

Fig. 6 and Fig. 7 show how performance is affected by varying task arrival rates. In all algorithms, the average task success ratio and level of task satisfaction decrease as task arrival rates rise. Our method is significantly more efficient than the greedy-FCFS and greedy-SJF scheduling algorithms. In particular, the suggested method can increase average task satisfaction degrees by around 50% and 25% over greedy-FCFS and greedy-SJF. As FCFS schedules earlier-arriving tasks first, subsequent tasks may be delayed when the earlier-arriving tasks demand a lot of CPU power. For long tasks, greedy SFJ prioritizes shorter tasks over longer ones. The expected delay demand is not taken into account by greedy-FCFS or greedy-SJF algorithms.

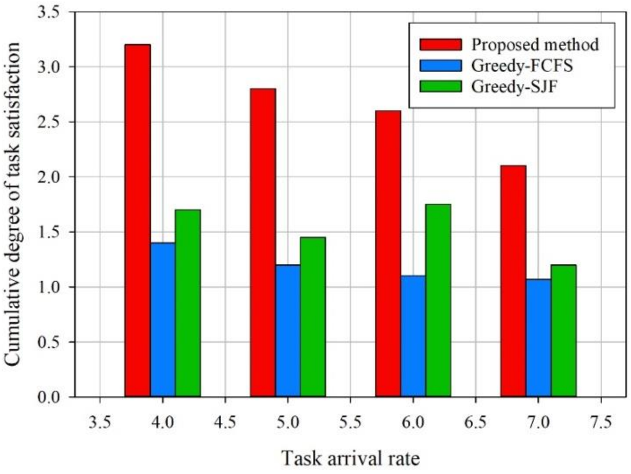


Fig. 6. Task satisfaction degree vs. task arrival rate.

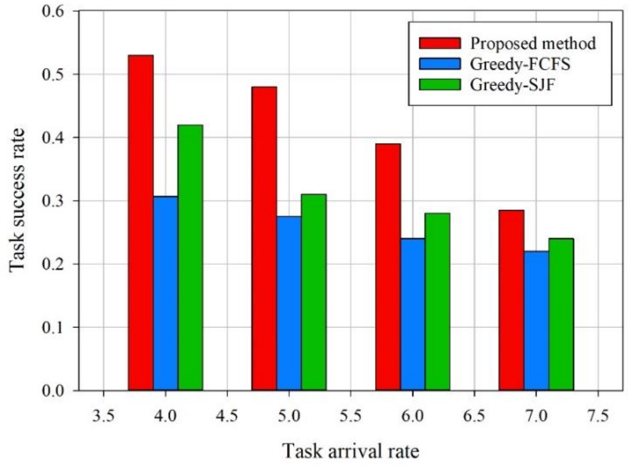


Fig. 7. Task success rate vs. task arrival rate.

Fig. 8 and Fig. 9 compare our task scheduling mechanism with the baselines in terms of task popularity. A higher skewness factor results in a higher degree of task satisfaction and a higher success ratio. The proposed algorithm can significantly improve task satisfaction, as shown in Fig. 8. The gap widens with increasing popularity factor values compared to greedy-SJF and greedy-FCFS algorithms. The greedy-SJF algorithm suffers from performance degradation due to a larger proportion of small tasks being assigned as popularity increases. The higher skewness factor allows for more accurate predictions of task lengths, leading to more efficient scheduling. This leads to better results for popularity-based algorithms compared to greedy-SJF and greedy-FCFS algorithms. The improved task satisfaction and success ratio observed in Fig. 8 are a direct result of this improved scheduling efficiency.

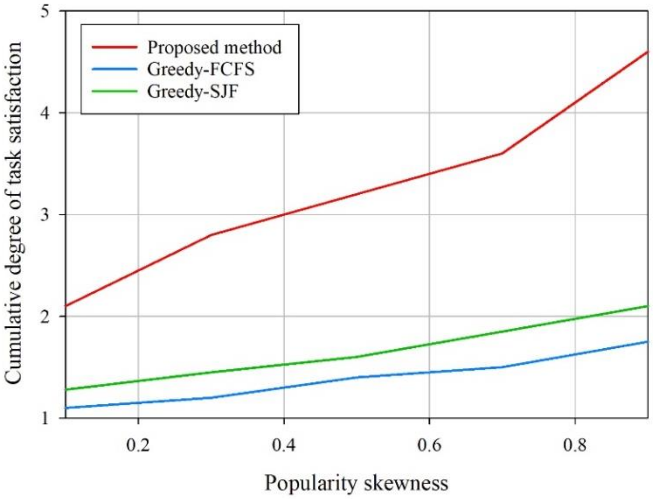


Fig. 8. Task satisfaction degree vs. populariry skewness.

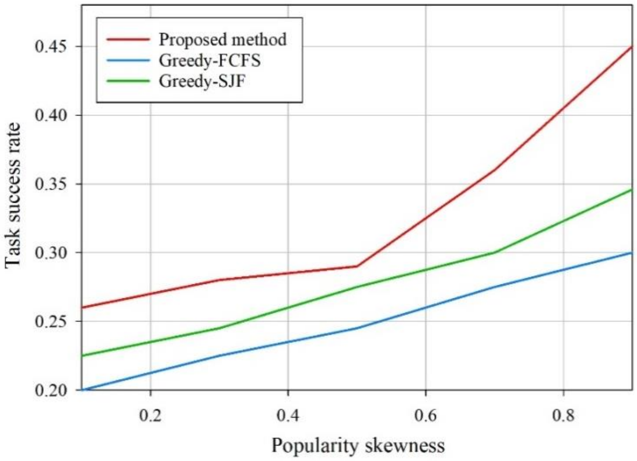


Fig. 9. Task success rate vs. popularity skewness.

V. Conclusion

With the rapid growth of IoT applications, edge and IoT devices have expanded their services in recent years. Fog computing offers a latency sensitivity advantage over cloud computing for IoT-enabled smart applications. Task scheduling effectively reduces application computation time and latency while improving quality of service. This paper's proposed task scheduling framework comprises two main phases: virtualization of edge nodes and task scheduling based on deep reinforcement learning. The first phase offers a layered edge framework. In the second phase, we applied DRL to schedule tasks taking into account the diversity of tasks and the heterogeneity of available resources. Simulations indicate that our proposed task scheduling method leads to greater levels of task satisfaction and success than existing approaches.

References

[1] Vahedifard, F., et al., Artificial intelligence for radiomics; diagnostic biomarkers for neuro-oncology. World Journal of Advanced Research and Reviews, 2022. 14(3): p. 304-310.

[2] Saeidi, S.A., et al. A novel neuromorphic processors realization of spiking deep reinforcement learning for portfolio management. in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022. IEEE.

[3] Akhavan, J. and S. Manoochehri. Sensory data fusion using machine learning methods for in-situ defect registration in additive manufacturing: a review. in 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). 2022. IEEE.

[4] Khosravi, F., et al. Implementation of an Elastic Reconfigurable Optical Add/Drop Multiplexer based on Subcarriers for Application in Optical Multichannel Networks. in 2022 International Conference on Electronics, Information, and Communication (ICEIC). 2022. IEEE.

[5] Khosravi, F., et al., Improving the performance of three level code division multiplexing using the optimization of signal level spacing. Optik, 2014. 125(18): p. 5037-5040.

[6] Haghshenas, S.H., M.A. Hasnat, and M. Naeini, A Temporal Graph Neural Network for Cyber Attack Detection and Localization in Smart Grids. arXiv preprint arXiv:2212.03390, 2022.

[7] Taami, T., S. Krug, and M. O'Nils. Experimental characterization of latency in distributed iot systems with cloud fog offloading. in 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS). 2019. IEEE.

[8] Pourghebleh, B., et al., The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments. Cluster Computing, 2021: p. 1-24.

[9] He, P., et al., Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review. Sustainable Computing: Informatics and Systems, 2022. 36: p. 100822.

[10] Pourghebleh, B., et al., A roadmap towards energy ‐ efficient data fusion methods in the Internet of Things. Concurrency and Computation: Practice and Experience, 2022: p. e6959.

[11] Kumar, A., et al., Smart power consumption management and alert system using IoT on big data. Sustainable Energy Technologies and Assessments, 2022: p. 102555.

[12] Stoyanova, M., et al., A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues. IEEE Communications Surveys & Tutorials, 2020. 22(2): p. 1191-1221.

[13] Khan, W.Z., et al., Industrial internet of things: Recent advances, enabling technologies and open challenges. Computers & Electrical Engineering, 2020. 81: p. 106522.

[14] Mohseni, M., F. Amirghafouri, and B. Pourghebleh, CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic. Peer-to-Peer Networking and Applications, 2022: p. 1-21.

[15] Cui, Y.-y., et al., A novel offloading scheduling method for mobile application in mobile edge computing. Wireless Networks, 2022. 28(6): p. 2345-2363.

[16] Alqarni, M.M., A. Cherif, and E. Alkayal, A Survey of Computational Offloading in Cloud/Edge-based Architectures: Strategies, Optimization Models and Challenges. KSII Transactions on Internet and Information Systems (TIIS), 2021. 15(3): p. 952-973.

[17] Zhang, D., et al., Task offloading method of edge computing in internet of vehicles based on deep reinforcement learning. Cluster Computing, 2022. 25(2): p. 1175-1187.

[18] Kamalov, F., et al., Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective. Sustainability, 2023. 15(4): p. 3317.

[19] Hasan, M.Z. and H. Al‐Rizzo, Task scheduling in Internet of Things cloud environment using a robust particle swarm optimization. Concurrency and Computation: Practice and Experience, 2020. 32(2): p. e5442.

[20] Abdel-Basset, M., et al., Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications. IEEE Transactions on Industrial Informatics, 2020.

[21] Shadroo, S., A.M. Rahmani, and A. Rezaee, The two-phase scheduling based on deep learning in the Internet of Things. Computer Networks, 2021. 185: p. 107684.

[22] Guizani, N. and A. Ghafoor, A network function virtualization system for detecting malware in large IoT based networks. IEEE Journal on Selected Areas in Communications, 2020. 38(6): p. 1218-1228.

[23] Al-Habob, A.A., et al., Task scheduling for mobile edge computing using genetic algorithm and conflict graphs. IEEE Transactions on Vehicular Technology, 2020. 69(8): p. 8805-8819.

[24] Chen, L., et al., Resource constrained profit optimization method for task scheduling in edge cloud. IEEE Access, 2020. 8: p. 118638-118652.

[25] Azizi, S., et al., Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. Journal of network and computer applications, 2022. 201: p. 103333.

[26] Kanbar, A.B. and K.H.A. Faraj, Region aware dynamic task scheduling and resource virtualization for load balancing in IoT-fog multi-cloud environment. Future Generation Computer Systems, 2022.

[27] Hussain, S.M. and G.R. Begh, Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog–cloud environment. Journal of Computational Science, 2022. 64: p. 101828.

[28] Najafizadeh, A., et al., Multi-objective Task Scheduling in cloud-fog computing using goal programming approach. Cluster Computing, 2022. 25(1): p. 141-165.

[29] Li, Z., et al., Credit-based payments for fast computing resource trading in edge-assisted Internet of Things. IEEE Internet of Things Journal, 2019. 6(4): p. 6606-6617.

[30] Wang, P., et al., Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems. IEEE Internet of Things Journal, 2018. 6(2): p. 2872-2884.

[31] Li, S., et al., Joint admission control and resource allocation in edge computing for internet of things. IEEE Network, 2018. 32(1): p. 72-79.

[32] Zhang, X., et al., Resource allocation for a UAV-enabled mobile-edge computing system: Computation efficiency maximization. IEEE Access, 2019. 7: p. 113345-113354.

[33] Ataie, I., et al. D 2 FO: Distributed Dynamic Offloading Mechanism for Time-Sensitive Tasks in Fog-Cloud IoT-based Systems. in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC). 2022. IEEE.

[34] Tan, H., et al. Online job dispatching and scheduling in edge-clouds. in IEEE INFOCOM 2017-IEEE Conference on Computer Communications. 2017. IEEE.

[35] Zhang, Y., et al., Resource scheduling for delay minimization in multi-server cellular edge computing systems. IEEE Access, 2019. 7: p. 86265-86273.

[36] Chen, X., et al., A hybrid task scheduling scheme for heterogeneous vehicular edge systems. IEEE Access, 2019. 7: p. 117088-117099.

[37] Chiang, Y.-H., T. Zhang, and Y. Ji, Joint cotask-aware offloading and scheduling in mobile edge computing systems. IEEE Access, 2019. 7: p. 105008-105018.

[38] Li, C., et al., Collaborative cache allocation and task scheduling for data-intensive applications in edge computing environment. Future Generation Computer Systems, 2019. 95: p. 249-264.

[39] Alameddine, H.A., et al., Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. IEEE Journal on Selected Areas in Communications, 2019. 37(3): p. 668-682.

A Blockchain-based Three-factor Mutual Authentication System for IoT using PUFs and Group Signatures

Meriam Fariss, Ahmed Toumanari
Laboratory of Applied Mathematics and Intelligent Systems Engineering (MAISI)
National School of Applied Sciences (ENSA), Agadir, Morocco[1]

Abstract—**The widespread adoption of Internet of Things has brought many benefits to society, such as increased efficiency and convenience in various aspects of daily life. However, this has also led to a rise in security threats. Moreover, resource-constrained feature of IoT devices makes them vulnerable to various attacks that compromise the user's privacy and sensitive information confidentiality. It is therefore essential to address the security concerns of IoT devices to ensure their reliable and secure operation. This paper proposes a blockchain-based three-factor mutual authentication system for IoT using Elliptic Curve Cryptography, physical unclonable functions and group signatures. The main purpose is to achieve a secure mutual authentication among different involved entities while providing anonymous group member authentication and reliable auditing. The AVISPA tool is utilized in the paper to formally prove that the proposed system satisfies the security and privacy requirements.**

Keywords—Internet of Things; blockchain; mutual authentication; physical unclonable functions; biometrics; group signatures; elliptic curve cryptography

I. Introduction

Internet of Things (IoT) is a rapidly developing technology that has gained significant traction in various fields such as healthcare, military, smart cities and houses [1]–[4]. It involves smart devices that collect thousands of gigabytes of data and use this collected data to make instant decisions that are immediately shared with remote users and servers. Nonetheless, the absence of inherent security measures renders the IoT-based architectures susceptible to many security breaches and privacy violations [5]. Many surveys and researches have been conducted to show the security challenges in IoT [6]–[8].

Authors in [9] indicate that there were 50 billion connected devices by the end of 2020 and this number is expected to increase to 14.7 billion by 2023. As the number of connected devices in IoT continues to increase, there are numerous challenges and issues that arise, particularly in regards to security and privacy. To overcome these challenges, new and emerging technologies such as fog computing and blockchain are integrated with IoT.

Blockchain has gained significant attention from researchers due to its ability to protect IoT devices and security-critical data [10]–[12]. By incorporating blockchain technology into IoT devices, it can provide an effective solution to the security and privacy challenges facing IoT devices. Blockchain can ensure the integrity and authenticity of data, and provide a secure platform for sharing data between devices. Additionally, blockchain can help to create a decentralized and trustless network, which is essential for secure communication and transactions between IoT devices. Blockchain's security stems from its use of cryptographic techniques, such as hash functions, digital signatures, and encryption, to ensure the integrity and confidentiality of data stored on the blockchain.

Besides blockchain, fog computing is an emerging technology that brings important enhancement to the security of IoT devices [13]. The limited resources of these latter leave them vulnerable to security threats. To address this issue, fog computing can enhance their capabilities by offering localized compute, storage, and networking for a cluster of IoT devices. By performing processing and storage tasks closer to IoT devices at the fog node instead of moving the data to a cloud server, fog computing reduces latency and increases network efficiency due to its high-quality services and quick response time. This can help address security concerns by reducing the amount of data that needs to be transmitted to the cloud, which in turn reduces the attack surface for cyber criminals. Additionally, fog computing can provide an additional layer of security by enabling real-time threat detection and response. This can help detect and mitigate security threats more quickly, reducing the potential damage that can be caused by such attacks.

A. Our Contribution

To address the aforementioned security threats while taking into consideration the resource constrained feature in IoT environment, we propose in this paper a blockchain-based secure mutual authentication system for IoT using Physical Unclonable Functions (PUFs) and group signatures providing the following advantages:

1) Permissioned blockchain: to achieve more control, privacy and high transparency over the network, we use a permissioned blockchain where only selected nodes are allowed to participate in consensus.

2) Group signature scheme with two authorities: In our proposed scheme, we distinguish between the group manager and the opening manager roles. The former is in charge of assigning private signing keys to group members, while the latter can open signatures. This enhanced security, as both have their own secret key, mitigates the risk of untrustworthy authorities.

3) Three-factor authentication: the proposed authentication protocol requires the user to provide three types of credentials: the first factor is something the user knows (password), the second factor is something the user has (hardware token) and the third factor is something the user is (biometric characteristic like a fingerprint, iris scan or facial recognition). Hence, the security of the system is strengthened, as it is much more difficult for an attacker to obtain all three types of credentials.

4) PUFs: they are used to generate a unique private key for every token, which can be used for cryptographic operations such as signing and encryption. The private key is generated by applying a one-way function to the PUF's response, which ensures that the private key cannot be reverse-engineered from the response. The private key is securely stored in the hardware token and can only be accessed by authorized users with the appropriate credentials.

5) Fog computing: provides a trusted entity with more computing and storage resources that supervises a group of IoT devices, controls access and manages communication between devices and remote users. This improves security by providing a local point of control, reducing data transmission, and improving network efficiency, reliability and scalability.

B. Organization

The remainder of this paper is organized as follows. In section II, we present an overview of the related work. Section III is dedicated to the basic concepts of blockchain namely smart contracts and the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. The cryptographic primitives adopted in this paper are presented in Section IV. The description of our proposed blockchain-based protocol is presented in Section V. We dedicate Section VI to the informal security analysis of our proposed protocol and the formal security analysis using the widely used AVISPA Tool. Finally, we draw our conclusions and present our future work in Section VII.

II. Related Work

Every year, numerous studies are conducted to secure exchanged data over the unattended IoT area. These studies suggested reliable architectures and frameworks to overcome the challenges and security threats in order to achieve secure mutual authentication between all involved parties in the IoT environment. Among these security solutions, blockchain technology brings many solutions for tackling security and privacy concerns in the context of IoT. In 2022, researchers in [14] developed a secure and efficient authentication mechanism for fog computing using blockchain technology. The proposed approach aims to overcome the limitations of traditional authentication methods while maintaining high levels of security and performance. Another secure IoT system was proposed in 2022 [15]. This paper presented a new approach to managing device identities in IoT systems based on blockchain technology. The approach enhanced data security through two methods: a lightweight time-based identification protocol that validates data using hub identification, and a blockchain application that provides secure data storage and sharing among multiple parties with easy access and immutability. In [16], researchers proposed a blockchain-based scheme where certificateless cryptography, Elliptic Curve Cryptography (ECC), and pseudonym-based cryptography (PBC) are employed. The goal is to achieve users' privacy and to hide the true identity of IoT devices using pseudonym-based cryptography. In 2020, Huang et al. [17] presented in their paper a blockchain-based authentication framework for IoT networks to achieve fast decentralized authentication while preserving privacy. The framework satisfied various security requirements such as strong key protection, identity anonymity, single registry, and traceability. In 2017, Cha et al. [18] suggested a blockchain-connected gateway design that is claimed to ensure security and user privacy. However, in 2020, Yavari et al. [19] revealed that [18] is vulnerable to various attacks, including secret disclosure, replay, traceability, and token reuse attacks. They proposed an improved blockchain-based authentication protocol that provides secure access management and anonymity. The paper in [20] proposed a solution to address security risks in IoT, particularly in decentralized authentication by providing a secure framework using blockchain technology which supported certificate issuance, update, revocation, and audit functions through the use of a smart contract. Authors in [21] presented a multi-layer security model for IoT devices operating in multi-hop cellular networks that utilized blockchain's distributed technology. The proposed model offered a viable approach to deploying decentralized blockchain technology for securing cellular-enabled IoT networks. After analyzing the limitations of traditional IoT authentication and security mechanisms, authors in [22] proposed a blockchain-based model to address these issues namely the single-point-failure issue.

III. Blockchain

Blockchain is a secure and transparent digital ledger that records transactions across a decentralized network. Nodes in the network must reach a consensus before new blocks of transactions can be added. The ledger includes various types of transactions, and each block contains a header with the previous block hash, timestamp, version, nonce, difficulty target, and Merkle root [23]. Blockchain can be divided into three types: public, permissioned, and private. Public blockchains like Bitcoin and Ethereum are open to anyone, transactions are validated by consensus mechanisms, and no central trusted authority is required. In contrast, only trusted participants are allowed in private and permissioned blockchains, but there are significant differences between them. In private blockchains, a single private entity controls the network, while in permissioned blockchains, a consortium of organizations adds an access control layer and allows multiple organizations to validate transactions, making it more decentralized than private blockchains.

A. PBFT

Introduced in the late 90s, Practical Byzantine Fault Tolerance (PBFT) [24] is a consensus algorithm used in permissioned blockchain networks to ensure secure and consistent agreement on the network's state even in the presence of malicious nodes. The network security is ensured by the PBFT algorithm as long as the number of faulty nodes is under a predefined threshold *$f=(n-1)/3$* where *n* is the total number of nodes. In the PBFT consensus, eligible nodes can switch from primary or leader nodes (during a period of time called view) to secondary nodes to reach a consensus on the state of the system. When the leader node is non faulty, the PBFT consensus works as follows:

- Client requests: The client initiates a request and sends it to the network.
- Pre-prepare message: The leader verifies the request message and broadcasts the pre-prepare message to all other replica nodes, containing the client's request and a sequence number.
- Prepare messages: Upon receiving the pre-prepare message, each replica node verifies its legitimacy and broadcasts a Prepare message.
- Commit messages: When no less than two-thirds of the total consensus nodes have sent Pepare messages, the leader node broadcasts a commit message to all other nodes (ie. a consensus has been reached on the client's request).
- Reply message: When at least two third of the received commit messages are valid, nodes return a reply message to the client containing the result of the request transaction.
- State Update: the ledger state is then updated in every consensus node.

If the system encounters a verification failure or network interruption case, then the View change protocol is executed to select another primary node in the network, that is responsible for carrying out the consensus process from the prepare phase through the following steps:

- View_Change_Request: after detecting an exception message, every node in the network broadcasts a view change request to all other participating nodes.
- New_View_Prep: after verifying the View_Change_Request and broadcasting an acknowledgment (no less than ⅔ of the total nodes), the nodes collaborate to prepare a new chosen primary node to substitute the previous one.
- New_View: through a voting process, nodes must reach a consensus to select the new primary node. This latter will take the responsibility of processing requests and generating new blocks.

B. *Smart Contract*

To address the trust problem in a decentralized environment, smart contracts are programs where required conditions are implemented using a Turing complete language (like go language in Hyperledger Fabric, Solidity in Ethereum). The smart contract byte code is stored in the blockchain platform with a unique address and automatically executed when predefined conditions are verified. It is replicated across all the blockchain consensus nodes. Hence, no third trusted party is needed to make decisions. The main benefits of smart contracts are speed, efficiency, trust and transparency. It also benefits from the security immutability features offered by the blockchain.

IV. Cryptographic Primitives

A. *Group Signatures*

Group signature is a type of digital signature scheme that was first proposed by Chaum and Van Heyst [25], and then many other contributions on group signature schemes were made in order to allow group members to anonymously sign messages while being traceable by a designated authority. In 2015, [26] proposed a novel short group signature scheme along with two group membership revocation methods that only disclose revocation information to verifiers. This section briefly presents the digital signature scheme by [26] which is adopted in our proposed protocol.

- System Setup phase

In this phase, the system parameters are initialized. The input is a security parameter λ and the output is (*PP*, *sk*, *gpk*, *trace*). Public parameters *$PP=(q,G_1,G_2,G_T,e,P_1,P_2,h())$*, where G_1, G_2 and G_T are three cyclic groups of λ-bit prime order q and $e{:}G_1 x G_2 \rightarrow G_T$ is a bilinear map. P_1 and P_2 are the generator points of G_1 and G_2 respectively, and $h{:}\{0,1\}^* \rightarrow Z_q$ is a secure hash function.

The Group Manager chooses randomly two secret parameters *d* and *s* in Z_q^* where *(d,s)* represents its private key *sk*. The group public key is *$gpk = (D, S, U)$* where *$D = d.P_1$*, *$S = s.P_2$* and *$U = u.P_1$*. The secret parameter *u* represents the private tracing key *$trace=u$* only known by the openening manager that uses it in *GTrace* algorithm to find the member's real identity.

- Enroll

In this phase group members are enrolled by the group manager. The input is *(PP, sk)* and the output is a private key *$gsk_i=(x_i , Z_i)$* is generated for each group member GM_i by the group manager who chooses randomly a distinct x_i for each member and sets $Z_i = z_i.P_1$ where $z_i = (d- x_i) (sx_i)^{-1} \in Z_q^*$. After that, the group manager computes *$tag_i = x_i.Z_i$* and maps it with the relevant group member's identity in a members table. This table also contains $status_i$ that shows if the member is allowed to access the network or is revoked.

- GSign

During this phase, a group member can sign his/her messages. The input is *(PP, gpk, gsk_i, message)* and the output is the signature *$\sigma = (C_1, C_2, c, w)$*. Each group member can sign his/her messages using his/her private key *gsk_i* as follows: choose $k \in Z_q^*$ randomly, computes *$C_1=k.P_1$,$C_2=x_i.Z_i+k.U$* and *$Q=e(U,S)^k$*, computes *$c=h(message,C_1,C_2,Q)$* and *$w=kc+x_i$*.

- GVerify

This is the signature verification phase. The input is *(message, σ, gpk)* and the output is the verification result if the message has been signed by a group member or not. The verifier computes:

$$Q' = \frac{e(C_2,S).e(P_1,P_2)^w}{e(cC_1+D,P_2)} \quad (1)$$

and checks if *c = h(message, C_1, C_2, Q')* to confirm or not the validity of the signature.

- MRev

This is the group membership revocation algorithm. The group manager publishes a Revocation List (RL) that contains $tag_i = x_i.Z_i$ for revoked members. The revocation algorithm MRev operated by the verifier takes as input the signature *σ=(C_1, C_2, c, w)* for each member in the RL and the output is: for each member in RL, verifiers can test whether the value of $tag_i = x_i.Z_i$ belongs to a revoked member as follows: compute $e(C_2 - tag_i , S)$ and compare it to Q'. If the equality holds, then the signature σ belongs to a revoked member. Hence, the signature is rejected.

- GTrace

This is the tracing algorithm that takes as input *(trace, message, σ)* and outputs the signer identity using the tracing key *trace = u*. The group opener computes $tag_i = x_i.Z_i = C_2 - u.C_1$, then it searches in the table mapping each tag_i with the corresponding member identity.

B. Fuzzy Commitment Scheme

Fuzzy commitment scheme F is a cryptographic primitive that allows a party to commit to a message without revealing the message itself. It was first introduced in 1999 by Juels and Wattenberg [27]. It is performed in two phases. The commitment phase where the committer creates a commitment by applying a one-way function to a random secret value and the message, which prevents the receiver from determining the message from the commitment. The opening phase where the committer discloses the secret value and committed message. The receiver can verify the commitment by applying the same one-way function to both values.

Fuzzy commitment enhances privacy and security in biometric based authentication systems, by generating a commitment value based on biometric data, such as a fingerprint or face scan, and a secret key.

C. Physical Unclonable Functions

A Physical Unclonable Function (PUF) [28] is a hardware security primitive that is designed to generate a unique signature for a physical device, based on its manufacturing process and the physical variations that occur during the manufacturing process. The mathematical model of a PUF is based on a challenge-response mechanism, where a unique response is generated for every challenge. A PUF can be represented as: R = P(C). The challenge-response pair (CRP) is unique to each PUF. The PUF takes a challenge C as input and produces a response, which is a unique digital fingerprint that can be used for device identification. The response R is generated based on the physical characteristics of the device, such as the pattern of its silicon crystal lattice or the precise positions of transistors in the circuit.

V. The Proposed BC-Auth Scheme

In this section, we discuss the details of our proposed system which is based on blockchain technology, group signatures, Elliptic Curve Cryptography, Physical Unclonable Functions, the Message Authentication Code (MAC) and biometrics. In this BC-Auth, we adopt the permissioned blockchain type, where only legitimate nodes are allowed to access the blockchain, in order to achieve high privacy protection. This blockchain is managed and maintained by permissioned nodes on the basis of the Practical Byzantine Fault Tolerance consensus (PBFT) (see Section III A). These consensus nodes have enough computational, communication and storage resources to operate hundreds of thousands transactions within seconds.

Similar to the Bitcoin block structure, the block in our design is composed of the previous block hash, block version, timestamp, block size, the Merkle root, transaction counter and the recorded transactions in this block.

A. BC-Auth Model

We describe, in this section, the BC-Auth scheme model. The participants involved in this model are:

- Group manager: a trusted entity that enrolls new legitimate group members and generates their private keys and the group public key. It also revokes malicious users and maintains the Revocation List.
- Opening manager: the group manager cannot trace group members. The entity responsible for members' tracing is the opening manager using a secret parameter *trace*.
- Group members: the remote users that are allowed to access the IoT devices remotely.
- Consensus nodes: in our model, we work with the permissioned blockchain where the participating nodes are chosen and authorized to maintain the blockchain under the PBFT consensus mechanism.
- Fog node: a trusted entity with additional computing and storage resources that oversees a group of IoT devices and manages access to them. It also ensures the communication between these devices and the remote users.
- Devices: IoT devices are resource-constrained devices. Each IoT device corresponds to a single fog node. They collect data from the physical world to control and manage the industrial processes.

B. Smart Contract and PBFT in BC-Auth

In our proposed design, each logged in user interacts with the smart contract to broadcast his/her request transactions where *status=0*. These pending transactions are firstly verified by consensus nodes via GVerify_{gpk} to confirm that they're signed by a group member. If the verification fails, the transaction is discarded. Secondly, the opening manager monitors the smart contract to retrieve new verified group members transactions and sends allowed transactions (*status=1*) to the blockchain after verifying that the signer does not belong to the RL. These latter are verified by consensus nodes via $\text{Verify}_{\text{pko}}$. Finally, the fog node monitors the smart contract to find new valid transactions. When a sufficient number of consensus nodes agree that the transaction is valid (i.e. successfully verified by $\text{GVerify}_{\text{gpk}}$ and $\text{Verify}_{\text{pko}}$), it can then be chained in the blockchain.

A new block is added to the PBFT-based blockchain by a designed primary node PN_i of the current consensus round as follows:

- PN_i collects all the valid transactions of the current round and appends them to a new candidate block $Block_i$;
- PN_i broadcasts $Block_i$ to all consensus nodes;
- Upon receiving $Block_i$, each node verifies its validity based on many parameters such as the Block header, the block generator digital signature, the list of transactions contained in the block,
- If $Block_i$ passes this verification successfully, then each node broadcasts a prepare message along with hash ($Block_i$) to all other participating nodes.
- If the number of received messages is no less than two third of the total nodes, then each node adds the candidate block to its local copy of the ledger.

Hence, the consensus is reached and $Block_i$ is added to the blockchain. If the consensus is not reached, then View-Change protocol is executed (see Section III A).

C. *BC-Auth Protocol*

Our proposed protocol consists of four phases: initialization phase, member enrollment, login and mutual authentication, and member revocation phase. In this section, we describe these steps in detail.

1) System initialization phase: The initialization phase must be performed before the execution of the protocol. This phase takes as input a security parameter λ and outputs the public parameters $PP=(q,G_1,G_2,G_T,e,P_1,P_2,h())$. Using the Elliptic Curve Integrated Encryption Scheme (ECIES) [29], the group manager chooses randomly two secret parameters d and s in Z_q^*, where (d,s) represents its private key sk. The opening manager generates a random secret parameter $u \in Z_q^*$ that represents its tracing key $trace \leftarrow u$. It also has its private/public key pair (sk_o, pk_o). The group public key is equal to $gpk = (D,S,U)$ where $D = d.P_1$, $S = s.P_2$ and $U = u.P_1$. In the other side, the fog node generates a random secret parameter $sk_{fn} \in Z_q^*$ that represents its private key, and computes the corresponding public key $pk_{fn}=sk_{fn}.P_1$. We assume that each smart device has its private/public key pair (d_j, D_j) where $d_j \in Z_q^*$ and $D_j=d_j.P_1$.

2) Member enrollment phase: This phase is initiated by the user who sends, via a secure channel, an enrollment request containing its chosen ID_i, HPW_i and biometric b_i to the Group Manager. This phase outputs the group members private keys $gsk_i=(x_i, Z_i)$ and the hardware token that stores $\{\alpha, \delta, u_i, A_i, Z_i, f(),PUF_i()\}$. These private keys are uniquely generated using PUFs functions and will be used by group members to anonymously sign their transactions before being sent to the blockchain network. Fig. 1 shows the details of the member enrollment phase.

3) Login and mutual authentication: This phase is performed every time a remote group member needs to access or control a smart device. Fig. 2 shows the login and mutual authentication phase.

- Login: To achieve a secure mutual authentication between the fog node and the group member, the latter needs to successfully login by inserting the correct ID_i, PW_i and b_i. This login phase is performed locally via the hardware token.
- Request transaction: Once the group member is logged in, his/her private key is computed using the physical unclonable function characteristics. A one-time private/public key pair (m_i, M_i) is generated where $m \in Z_q^*$ is a random secret and $M_i=m_i.P_1$. The transaction is structured as follows: Computes the message $Msg = txnumber||to||M_i||D_j||request$, where $txnumber$ is the transaction number and to is the address of the smart contract. Then it computes $E_{msg}=Enc_{pkfn}(Msg)$ where Enc is an AES Encryption, and sends the transaction $Tx= \{data, GSign_{gski}(data)\}$ where $data=(to,E_{msg}, TS_i, status)$ to the blockchain. TS_i is the current timestamp. The status=0 which means that it is a pending transaction. The smart contract is invoked and the consensus nodes verify the signed transaction using the GVerify algorithm. If the verification fails, i.e. the signer is not a group member, then the transaction is discarded. Else, the opening manager monitors the blockchain to find the pending transactions. It verifies then if the signer is a revoked member, i.e. the corresponding $tag_i \in RL$ by computing $tag_i=x_i.Z=C_2-u.C_1$. The revoked transaction is discarded from the blockchain. The allowed transaction Tx' status is set to 1 and is sent to the blockchain. The consensus nodes verify the validity of this transaction by operating the Elliptic Curve Digital Signature Algorithm (ECDSA) [29] Verify using the opening manager public key pk_o.
- Transaction chaining: The transaction is considered valid when at least two-third of total consensus nodes verify the transaction successfully through $\text{GVerify}_{\text{gpk}}$ (for Tx) and $\text{Verify}_{\text{pko}}$ (for Tx'). All valid transactions recorded through a predefined period of time are chained into a pending block that can be chained into the blockchain when the PBFT consensus is reached among more than two-third of total consensus nodes.
- Response delivery: The Fog node monitors the blockchain to retrieve the new valid transactions and decrypts the message $Msg = Dec_{skfn}(E_{msg})$, where Dec is an AES Decryption, and computes the signature $R=Sign_{skfn}(M_i \mid\mid request)$ using its private key and sends $\{M_i, request, R\}$ to the targeted IoT device. After successfully verifying the received request, the device encrypts the response using the one time group member's public key M_i, and signs it with its private

key d_j. If the response passes the signature verification Verify_{Dj} (RES, E_{res},), then the fog node computes MAC = MAC_{SK} (E_{res}) using the secret session key SK and sends {E_{res}, MAC} to the group member. The latter compares the $MAC'=MAC_{SK'}(E_{res})$ to the received MAC. If the equality holds, this means that the secure mutual authentication is successfully established between the fog node and the group member, who can then decrypt the response using its one-time private key.

4) Member revocation: The group manager maintains the RL that contains $tag_i=x_i.Z_i$ of revoked users. If the behavior of a group member is malicious, or if he does not belong to the group anymore, the group manager can then revoke its membership and add the corresponding tag_i in the public RL. This revocation list is shared only between the group manager and the opening manager.

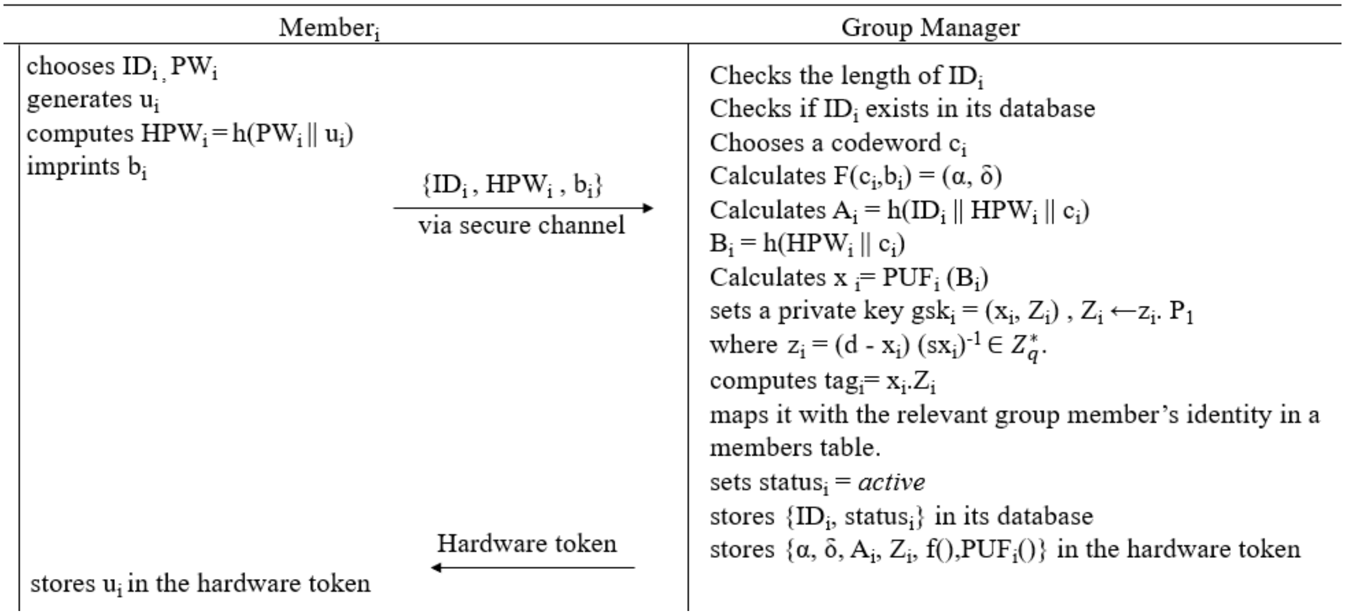

Member_i
Group Manager
chooses ID_i , PW_i
generates u_i
computes $HPW_i = h(PW_i \| u_i)$
imprints b_i
$\{ID_i , HPW_i , b_i\}$
via secure channel
Checks the length of ID_i
Checks if ID_i exists in its database
Chooses a codeword c_i
Calculates $F(c_i,b_i) = (\alpha, \delta)$
Calculates $A_i = h(ID_i \| HPW_i \| c_i)$
$B_i = h(HPW_i \| c_i)$
Calculates $x_i = PUF_i (B_i)$
sets a private key $gsk_i = (x_i, Z_i)$, $Z_i \leftarrow z_i. P_1$
where $z_i = (d - x_i) (sx_i)^{-1} \in Z_q^*$.
computes $tag_i = x_i.Z_i$
maps it with the relevant group member's identity in a members table.
sets $status_i = active$
stores $\{ID_i, status_i\}$ in its database
stores $\{\alpha, \delta, A_i, Z_i, f(), PUF_i()\}$ in the hardware token
Hardware token
stores u_i in the hardware token

Fig. 1. Member enrollment phase.

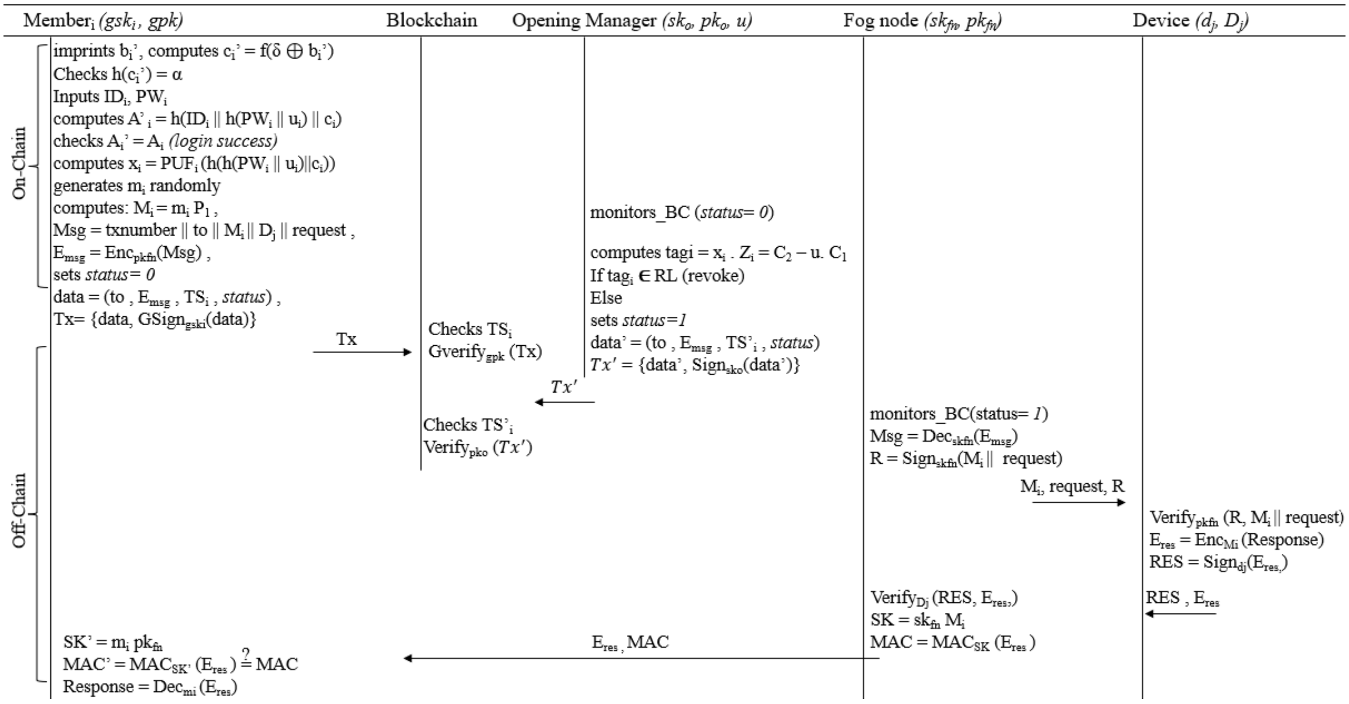

Member_i (gsk_i, gpk)
Blockchain
Opening Manager (sk_o, pk_o, u)
Fog node (sk_{fn}, pk_{fn})
Device (d_j, D_j)
On-Chain
Off-Chain
imprints b_i', computes $c_i' = f(\delta \oplus b_i')$
Checks $h(c_i') = \alpha$
Inputs ID_i, PW_i
computes $A'_i = h(ID_i \| h(PW_i \| u_i) \| c_i)$
checks $A_i' = A_i$ *(login success)*
computes $x_i = PUF_i (h(h(PW_i \| u_i)\|c_i))$
generates m_i randomly
computes: $M_i = m_i P_1$,
$Msg = txnumber \| to \| M_i \| D_j \| request$,
$E_{msg} = Enc_{pkfn}(Msg)$,
sets *status= 0*
data = (to , E_{msg} , TS_i , *status*) ,
Tx= {data, $GSign_{gski}$(data)}
Tx
Checks TS_i
$Gverify_{gpk}$ (Tx)
monitors_BC (*status= 0*)
computes tagi = $x_i . Z_i = C_2 - u. C_1$
If $tag_i \in RL$ (revoke)
Else
sets *status=1*
data' = (to , E_{msg} , TS'_i , *status*)
Tx' = {data', $Sign_{sko}$(data')}
Tx'
Checks TS'_i
$Verify_{pko}$ (Tx')
monitors_BC(status= *1*)
$Msg = Dec_{skfn}(E_{msg})$
$R = Sign_{skfn}(M_i \|$ request)
M_i, request, R
$Verify_{pkfn}$ (R, $M_i \|$ request)
$E_{res} = Enc_{Mi}$ (Response)
$RES = Sign_{dj}(E_{res},)$
RES , E_{res}
$Verify_{Dj}$ (RES, E_{res},)
$SK = sk_{fn} M_i$
$MAC = MAC_{SK} (E_{res})$
E_{res} , MAC
$SK' = m_i pk_{fn}$
$MAC' = MAC_{SK'} (E_{res}) \overset{?}{=} MAC$
Response = $Dec_{mi} (E_{res})$

Fig. 2. Login and mutual authentication phase.

VI. Security Analysis

A. Informal Security Analysis

In this section, we provide an informal overview of the security aspects of our protocol, highlighting its robustness against various security threats and its important security features. Furthermore, we present a comparison of the proposed BC-Auth scheme with other competing schemes in terms of functionality and security features. The results of this comparison are summarized in Table I.

1) Mutual authentication: First, the consensus nodes and the opening manager authenticate the user by verifying the request transactions via group signature scheme. Then, the IoT device authenticates the fog node by verifying its digital signature (ECDSA), confirming that the fog node is authorized to access the device's data. Finally, mutual authentication is achieved between the fog node and group members through the use of MAC that provides a way to verify the authenticity and integrity of messages exchanged between the two parties. By using MAC, the fog node can authenticate the group members and vice versa, ensuring that only authorized parties can access the system.

2) Single registration: The process of single registration involves the issuance of private keys to each group member by the group manager, which is based on a unique identifier (PUF). Even when some members are revoked by the group manager, legitimate members can continue to use their private keys for signing transactions, which eliminates the need for multiple registrations and minimizes the risk associated with key management.

3) Suitable for IoT: The proposed protocol is designed to be compatible with the constraints and requirements of IoT devices, which typically have limited processing power, memory, and energy resources. To address these constraints, the proposed protocol leverages the fog node as an intermediary between IoT devices and the blockchain network, allowing IoT devices to offload some of the computation and communication tasks to the fog node. This approach reduces the computational burden on IoT devices and enables them to participate in the blockchain network securely and efficiently. Additionally, the fog node can act as a gateway for IoT devices that are not directly connected to the internet, providing them with secure and reliable access to the blockchain network. Therefore, the proposed protocol is suitable for IoT applications that require secure and efficient communication with the blockchain network.

TABLE I. Security Requirements Comparison

Security requirement	[30]	[31]	[32]	Our protocol
Anonymity	No	Yes	No	Yes
Traceability	Yes	No	Yes	Yes
Confidentiality	No	Yes	Yes	Yes
Revocation	No	Yes	No	Yes
Mutual Authentication	Yes	Yes	No	Yes
Timely Tempo detection	No	No	No	Yes

4) Resistance to cloning and counterfeiting: The proposed system uses PUF to generate members' private keys, which are then securely stored in a hardware token. PUF is a technique that leverages the inherent randomness of physical systems to generate unique and unclonable keys. By utilizing PUF, the system can ensure that the private keys of group members cannot be cloned or counterfeited, which provides a higher level of security. Even if an attacker gains access to the hardware token, they will not be able to clone or counterfeit the private key, as it is generated using PUF, which is a unique physical characteristic of the device. This resistance to cloning and counterfeiting is important because it ensures that the private keys of group members cannot be compromised, which would otherwise compromise the entire system.

5) Resistance to man in the middle attack: In the proposed protocol, a MAC is used to verify the authenticity and integrity of the exchanged messages between the communicating parties. MAC is generated by computing a cryptographic hash function over a shared secret key and the message. This shared secret key is only known to the legitimate parties, ensuring that any changes made to the message by an attacker will be detected by the receiving party. This ensures that the messages cannot be tampered with or intercepted by a malicious party without detection, preventing man-in-the-middle attacks.

6) Resistance to stolen hardware token: In the proposed system, the private keys of group members are securely generated and stored in hardware tokens using PUF. Additionally, biometric authentication is used to ensure that only the legitimate owner of the hardware token can access the private key. In the event of a hardware token being stolen, the group manager can revoke the token, rendering the private key unusable. This approach provides resistance to stolen hardware tokens and protects the system against attacks that attempt to use a stolen token to impersonate a legitimate group member.

7) Session key agreement and resistance to replay attacks: In the proposed system, session key agreement is used to establish secure communication between the fog node and the user. In every session, a fresh private/public key pair is generated for every user. This ensures that each session has a unique session key, which is used to encrypt responses and to generate a session MAC between the fog node and the user. By using a fresh private/public key pair for every session, the system resists replay attacks. If the same key is used in every session, an attacker could use a previously intercepted session key to forge or replay messages, which would compromise the security of the system. However, by using a fresh private/public key pair for every session, the system ensures that each session has a unique session key, which makes it much more difficult for an attacker to replay messages.

8) Timely tempo detection: In the proposed system, there is a login phase that locally checks the user credentials before granting access to the system. This means that the system can quickly detect and reject unauthorized users who try to access the system without valid credentials. By doing so, the system can prevent potential security breaches and minimize the communication and computation costs. Therefore, the timely tempo detection helps to enhance the security of the system and protect it from unauthorized access.

9) Traceability: In the proposed system, group signatures are utilized to provide traceability for suspicious transactions. Only the opening manager has the ability to trace these transactions through a mechanism called GTrace. Other parties who wish to trace suspicious transactions must compromise the ElGamal encryption, which is infeasible under the current security assumptions. Thus, the use of group signatures in the system allows for traceability by authorized parties while preserving the anonymity of group members for regular transactions.

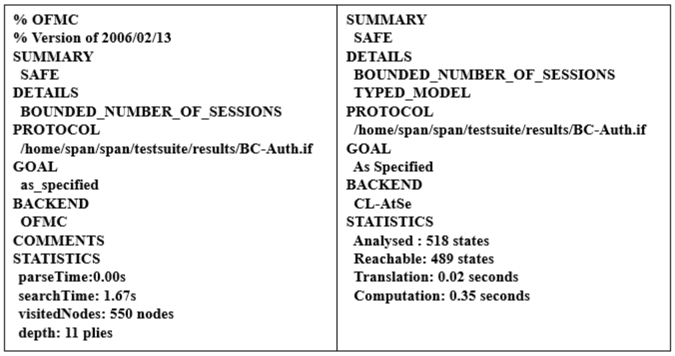
10)User anonymity: User anonymity in this context refers to the fact that the proposed system ensures that users can sign transactions without revealing their real identities. The system utilizes a group signature scheme, where a member's identity is not disclosed, and only the opening manager knows the true identity of the member who signed the transaction. Additionally, for each new transaction, a one-time public key is used, which further obscures the identity of the signer.

The comparison results in Table I show that our proposed protocol provides anonymity, traceability, confidentiality, revocation, mutual authentication, and timely tempo detection, making it a more comprehensive and robust solution for securing IoT systems than protocols [30], [31], and [32]. The lack of these features in the other protocols makes them vulnerable to security breaches, privacy violations, and man-in-the-middle attacks.

B. Formal Security Analysis using AVISPA Tool

In this section, we provide a formal security analysis of our protocol, Fig. 3, based on the widely used Automated Validation of Internet Security Protocols and Applications (AVISPA) Tool [33]. This latter takes in input a formal model of a security protocol and verifies its robustness against a set of security properties. AVISPA tool can also be used to perform security analysis on blockchain protocols. It is capable of analyzing smart contracts and consensus protocols. Avispa supports four backends for analysis: OFMC (On the fly Model Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols). Each backend has its own strengths and weaknesses.

In our analysis we adopt the OFMC and CL-AtSe backends and we use the syntax provided by the High-Level Protocol Specification Language (HLPSL) supported by AVISPA. In this HLPSL specification, we define the principal roles representing our model: group manager, group opener, group member, consensus node, fog node and IoT device. In addition, there are two composed roles, session and environment, and a goals section where the security goals are specified. As you can see in Fig. 3, the obtained analysis results show that our proposed protocol is safe under the OFMC and CL-AtSe backends.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/BC-Auth.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime:0.00s
 searchTime: 1.67s
 visitedNodes: 550 nodes
 depth: 11 plies
```

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /home/span/span/testsuite/results/BC-Auth.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed : 518 states
 Reachable: 489 states
 Translation: 0.02 seconds
 Computation: 0.35 seconds
```

Fig. 3. Analysis result using OFMC and CL-AtSe backends.

VII. CONCLUSION

In this paper, we proposed a blockchain-based secure mutual authentication system for IoT using PUFs and group signatures. The proposed BC-Auth scheme provides several advantages, including the use of permissioned blockchain for more control and privacy, a group signature scheme with two authorities for enhanced security, three-factor authentication for stronger user verification, PUFs for unique private key generation, and fog computing for improved security and efficiency. The proposed system aims to strengthen IoT security and efficiency by mitigating risks and making it more difficult for attackers to obtain user credentials.

We also proved the security of our protocol informally and formally using the AVISPA tool and provided a security comparison with other blockchain based authentication protocols.

In our future work, we are working on:

1) The practical implementation of the proposed BC-Auth protocol using the Hyperledger Fabric which is a permissioned blockchain platform designed for enterprise use cases with modular architecture and privacy features,

2) The practical simulation of the proposed BC-Auth protocol to prove its performance in terms of computational and communication costs.

3) Evolving our protocol by proposing a blockchain based multiple managers' group signature scheme to avoid the risks related to the single authority in the case of one group manager model.

REFERENCES

[1] S. Selvaraj and S. Sundaravaradhan, "Challenges and opportunities in IoT healthcare systems: a systematic review," SN Applied Sciences, vol. 2, no. 1. Springer Nature, Jan. 01, 2020, doi: 10.1007/s42452-019-1925-y.

[2] P. Fraga-Lamas, T. M. Fernández-Caramés, M. Suárez-Albela, L. Castedo, and M. González-López, "A Review on Internet of Things for Defense and Public Safety," Sensors (Basel, Switzerland), vol. 16, no. 10. Oct. 05, 2016, doi: 10.3390/s16101644.

[3] K. Szum, "IoT-based smart cities: A bibliometric analysis and literature review," Eng. Manag. Prod. Serv., vol. 13, no. 2, pp. 115–136, Jun. 2021, doi: 10.2478/emj-2021-0017.

[4] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges," IEEE Access, vol. 8. Institute of Electrical

and Electronics Engineers Inc., pp. 32031–32053, 2020, doi: 10.1109/ACCESS.2020.2973178.

[5] M. Aqeel, F. Ali, M. W. Iqbal, T. A. Rana, M. Arif, and M. R. Auwul, “A Review of Security and Privacy Concerns in the Internet of Things (IoT),” J. Sensors, vol. 2022, pp. 1–20, Sep. 2022, doi: 10.1155/2022/5724168.

[6] P. P. Ray, “A survey on Internet of Things architectures,” Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 3. King Saud bin Abdulaziz University, pp. 291–319, Jul. 01, 2018, doi: 10.1016/j.jksuci.2016.10.003.

[7] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Comput. Networks, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[8] L. Babun, K. Denney, Z. B. Celik, P. McDaniel, and A. S. Uluagac, “A survey on IoT platforms: Communication, security, and privacy perspectives,” Computer Networks, vol. 192. Elsevier B.V., Jun. 19, 2021, doi: 10.1016/j.comnet.2021.108040.

[9] N. Sharma, M. Shamkuwar, and I. Singh, “The history, present and future with iot,” in Intelligent Systems Reference Library, vol. 154, Springer Science and Business Media Deutschland GmbH, 2019, pp. 27–51.

[10] S. Tanwar, N. Gupta, C. Iwendi, K. Kumar, and M. Alenezi, “Next Generation IoT and Blockchain Integration,” Journal of Sensors, vol. 2022. Hindawi Limited, 2022, doi: 10.1155/2022/9077348.

[11] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, “Blockchain and iot integration: A systematic survey,” Sensors (Switzerland), vol. 18, no. 8. MDPI AG, Aug. 06, 2018, doi: 10.3390/s18082575.

[12] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, “A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling,” ACM Computing Surveys, vol. 53, no. 1. Association for Computing Machinery, Feb. 01, 2020, doi: 10.1145/3372136.

[13] H. Sabireen and V. Neelanarayanan, “A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges,” ICT Express, vol. 7, no. 2, pp. 162–176, Jun. 2021, doi: 10.1016/j.icte.2021.05.004.

[14] O. Umoren, R. Singh, S. Awan, Z. Pervez, and K. Dahal, “Blockchain-Based Secure Authentication with Improved Performance for Fog Computing,” Sensors, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228969.

[15] F. Sabrina, N. Li, and S. Sohail, “A Blockchain Based Secure IoT System Using Device Identity Management,” Sensors, vol. 22, no. 19, Oct. 2022, doi: 10.3390/s22197535.

[16] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, “A Blockchain-Based Mutual Authentication Scheme for Collaborative Edge Computing,” IEEE Trans. Comput. Soc. Syst., vol. 9, no. 1, pp. 146–158, Feb. 2022, doi: 10.1109/TCSS.2021.3056540.

[17] C. Huang and K. Yan, “A Blockchain Based Fast Authentication Framework for IoT Networks with Trusted Hardware,” in Proceedings - 2020 IEEE 22nd International Conference on High Performance Computing and Communications, IEEE 18th International Conference on Smart City and IEEE 6th International Conference on Data Science and Systems, HPCC-SmartCity-DSS 2020, Dec. 2020, pp. 1050–1056, doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00141.

[18] S. C. Cha, J. F. Chen, C. Su, and K. H. Yeh, “A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things,” IEEE Access, vol. 6, pp. 24639–24649, Jan. 2018, doi: 10.1109/ACCESS.2018.2799942.

[19] M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C. M. Chen, “An Improved Blockchain-Based Authentication Protocol for IoT Network Management,” Secur. Commun. Networks, vol. 2020, 2020, doi: 10.1155/2020/8836214.

[20] Y. Hu, D. Yin, and C. Huang, “BBSF: A blockchain based secure framework for the internet of things with user revocation,” in Proceedings of 2020 IEEE International Conference on Progress in Informatics and Computing, PIC 2020, Dec. 2020, pp. 358–362, doi: 10.1109/PIC50277.2020.9350772.

[21] H. H. Pajooh, M. Rashid, F. Alam, and S. Demidenko, “Multi-layer blockchain-based security architecture for internet of things,” Sensors (Switzerland), vol. 21, no. 3, pp. 1–26, Feb. 2021, doi: 10.3390/s21030772.

[22] D. Li, W. Peng, W. Deng, and F. Gai, A Blockchain-based Authentication and Security Mechanism for IoT, 2018 27th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2018.

[23] A. M. Antonopoulos, Mastering Bitcoin : unlocking digital crypto-currencies. 2015.

[24] M. Castro, M. Research, and B. Liskov, “Practical Byzantine Fault Tolerance and Proactive Recovery,” 2002.

[25] D. C. Eugtne van Heyst, “Group Signatures,” 1991.

[26] T. H. Ho, L. H. Yen, and C. C. Tseng, “Simple-Yet-Efficient Construction and Revocation of Group Signatures,” Int. J. Found. Comput. Sci., vol. 26, no. 5, pp. 611–624, Aug. 2015, doi: 10.1142/S0129054115500343.

[27] A. Juels and M. Wattenberg, “Fuzzy commitment scheme,” in Proceedings of the ACM Conference on Computer and Communications Security, 1999, pp. 28–36, doi: 10.1145/319709.319714.

[28] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, “Physical unclonable functions and applications: A tutorial,” Proc. IEEE, vol. 102, no. 8, pp. 1126–1141, 2014, doi: 10.1109/JPROC.2014.2320516.

[29] D. Hankerson, Menezes Alfred, and Vanstone Scott, Guide to Elliptic Curve Cryptography. 2004.

[30] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, “Bubbles of Trust: A decentralized blockchain-based authentication system for IoT,” Comput. Secur., vol. 78, pp. 126–142, Sep. 2018, doi: 10.1016/j.cose.2018.06.004.

[31] C. Lin, D. He, X. Huang, K. K. R. Choo, and A. V. Vasilakos, “BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0,” J. Netw. Comput. Appl., vol. 116, pp. 42–52, Aug. 2018, doi: 10.1016/j.jnca.2018.05.005.

[32] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, “Blockchain for Large-Scale Internet of Things Data Storage and Protection,” IEEE Trans. Serv. Comput., vol. 12, no. 5, pp. 762–771, Sep. 2019, doi: 10.1109/TSC.2018.2853167.

[33] A. Armando et al., “The AVISPA tool for the automated validation of internet security protocols and applications,” in Lecture Notes in Computer Science, 2005, vol. 3576, pp. 281–285, doi: 10.1007/11513988_27.

Self-adapting Security Monitoring in Eucalyptus Cloud Environment

Salman Mahmood[1], Nor Adnan Yahaya[2], Raza Hasan[3], Saqib Hussain[4], Mazhar Hussain Malik[5], Kamal Uddin Sarker[6]
School of Information Technology, Malaysia University of Science and Technology, Selangor, Malaysia[1, 2]
Computing and Information Technology, Global College of Engineering and Technology, Muscat, Oman[3, 4]
Department of Computer Science and Creative Technologies, University of West of the England Bristol, Bristol, England[5]
Department of Computer Science, American International University Bangladesh, Dhaka, Bangladesh[6]

***Abstract*—This paper discusses the importance of virtual machine (VM) scheduling strategies in cloud computing environments for handling the increasing number of tasks due to virtualization and cloud computing technology adoption. The paper evaluates legacy methods and specific VM scheduling algorithms for the Eucalyptus cloud environment and compare existing algorithms using QoS. The paper also presents a self-adapting security monitoring system for cloud infrastructure that takes into account the specific monitoring requirements of each tenant. The system uses Master Adaptation Drivers to convert tenant requirements into configuration settings and the Adaptation Manager to coordinate the adaptation process. The framework ensures security, cost efficiency, and responsiveness to dynamic events in the cloud environment. The paper also presents the need for improvement in the current security monitoring platform to support more types of monitoring devices and cover the consequences of multi-tenant setups. Future work includes incorporating log collectors and aggregators and addressing the needs of a super-tenant in the security monitoring architecture. The equitable sharing of monitoring resources between tenants and the provider should be established with an adjustable threshold mentioned in the SLA. The results of experiments show that Enhanced Round-Robin uses less energy compared to other methods, and the Fusion Method outperforms other techniques by reducing the number of Physical Machines turned on and increasing power efficiency.**

***Keywords*—Component; VM scheduling; cloud computing; Eucalyptus; virtualization; power efficiency; self-adapting security monitoring system; tenant-driven customization; dynamic events; adaptation manager; master adaptation drivers**

I. Introduction

Cloud computing is a technology that provides on-demand access to a pool of resources (such as networks, servers, storage, applications, and services) through networks. It is offered by companies like Google, Amazon, and SalesForce and eliminates the need for users to handle administration and IT maintenance [1]. Resource scheduling and allocation can be challenging for cloud providers due to the dynamic behavior of services and multiple types of cloud systems available [2]. Cloud computing is a form of computing as a service rather than a product, providing customers with access to software, resources, and information as a utility. It is cost-effective, with lower upfront costs and a pay-per-use model [3]. Virtualization technology is used by cloud providers to improve cost-efficiency and energy efficiency. Cloud computing is used in various applications such as website hosting, scientific methods, customer relationship management, and high-performance computing [4].

Server virtualization allows for the allocation of computer resources (such as CPU and RAM) on demand through a pay-as-you-go model, where clients (tenants) only pay for what they use. Infrastructure as a Service (IaaS) is a popular cloud model using virtual machines (VMs) and virtual networks to provide tenants access to compute, storage, and network resources. By outsourcing some information systems through the virtual infrastructure on the cloud provider's physical infrastructure, businesses can enjoy automated management, flexible resource allocation, and the illusion of unlimited computing and networking capabilities, as outlined in the Service Level Agreement signed by tenants and the cloud provider.

Despite the potential cost and efficiency benefits of cloud adoption, security remains a major concern. Multi-tenancy, an essential aspect of cloud architecture, enables the coexistence of trustworthy and hostile virtual machines, making the cloud vulnerable to attacks from both inside and outside the environment [5]. A successful attack could lead to alteration of data stored in the cloud, including login credentials, and even complete control of the cloud infrastructure for illicit purposes. Traditional security solutions like traffic filtering and inspection are insufficient against sophisticated threats targeting virtual infrastructures. To ensure cloud security, an automated self-contained security architecture incorporating multiple protection and monitoring technologies is necessary [6].

In Infrastructure as a Service (IaaS) cloud architecture, tenants are responsible for managing their virtual information systems while the provider manages the physical infrastructure. Tenants have concerns about security monitoring of their virtualized infrastructure and need a solution that considers their specific security requirements and can respond to dynamic events in the cloud environment. This research aims to create a self-adaptable security monitoring framework to address these concerns and ensure adequate security monitoring for tenants' virtual infrastructures. The research work focuses on creating a self-adaptive security monitoring framework for cloud infrastructure. The framework should respond to dynamic events in the cloud and adjust its components accordingly, while maintaining a balance between security, performance, and cost. It should incorporate tenant-driven customization and meet tenant-defined thresholds and security requirements. The framework should not create new vulnerabilities and should not significantly impact performance or regular cloud operations. The research work also assesses existing cloud computing approaches and scheduling methodologies, and explores Eucalyptus cloud scheduling methods. Two independent Eucalyptus virtual machine scheduling techniques are proposed and evaluated, which aim to improve energy efficiency in cloud data centers.

Cloud computing is becoming increasingly prevalent in various industries, and efficient resource allocation and security are critical for maintaining acceptable throughput and revenue. Therefore, it is important to evaluate and compare existing VM scheduling algorithms and develop a comprehensive self-adapting security monitoring system that meets the specific needs of each tenant in a cloud environment. While there are many existing virtual machine scheduling algorithms for cloud computing environments, the specific context of the Eucalyptus cloud environment has not been extensively studied. Therefore, this paper aims to evaluate and compare existing VM scheduling algorithms in the Eucalyptus cloud environment, with a focus on energy efficiency. The paper addresses are the lack of a comprehensive self-adapting security monitoring system that takes into account the specific requirements of each tenant in a cloud environment. The paper proposes a framework for such a system that combines precise security monitoring with self-adaptation.

This paper is organized as follows: Section II presents a literature review and an overview of related works in this field. Section III presents the design and methodology used in the study. Section IV presents the discussion on the study and results. Finally, Section V draws a conclusion and proposes future research.

II. Literature Review

A. New Approach to Cloud Computing

Cloud computing is a software-based network infrastructure that enables users to access and store data on demand. It allows for flexible, elastic and cost-effective use of IT resources without the need for new hardware or software. The five core properties of cloud computing include independence, resource pooling, on-demand self-provisioning, rapid adaptation, and a consistent network as shown in Fig. 1. Cloud computing also includes three delivery options: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). There are four deployment strategies for cloud computing: public, private, communal, and hybrid clouds [7].

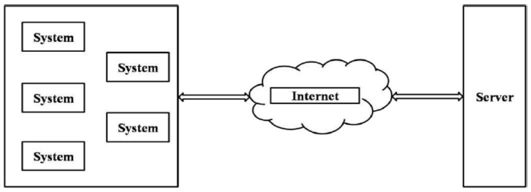


Fig. 1. Basic architecture of cloud.

B. Cloud Computing Models based on Services

The Cloud Security Alliance has identified security and privacy concerns as major obstacles to trusted cloud computing. Different security levels are needed for public and private clouds and Service-Level Agreements (SLAs) define customer and cloud provider responsibilities. Key safeguards include data integrity, vendor trust, consumer confidentiality, individual users and user groups. There are three delivery methods for cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS provides a layer of middleware and OS services, while PaaS offers a web-based platform to run apps as shown in Fig. 2. SaaS is the pinnacle of cloud computing technology, provided as a service and accessed online [8]. PaaS has the advantage of cost-effectiveness, but there is a risk of lock-in if it uses proprietary interfaces or programming languages.

C. Cloud Computing Security Issues and Challenges

Cloud computing is a rapidly growing field in IT that provides benefits such as improved infrastructure and cost-efficient access to services. However, it also presents risks such as security threats, privacy concerns, and reliability issues [9]. To address these, there is a move towards open standards, better compatibility, and compliance resources from cloud service providers. When considering cloud computing, it is important to also consider its long-term viability. There are challenges to cloud computing solutions such as lack of interoperability, compatibility with existing programs, difficulty in meeting regulations, and insufficient security. Lack of standardization and proprietary applications can lead to complexity and high costs, as well as security concerns with shared infrastructure not providing enough security [10].

D. Resource Allocation

Network resources or shared resources can include files, the challenge in sharing resources such as data, audio and video, and hardware in a cloud environment can be addressed through resource management. This involves coordinating IT resources by controlling templates, managing virtual IT resources, performing load balancing, resource replication and failover, and monitoring the operational conditions of IT resources [11]. The cloud provider or user accesses these functionalities through a cloud resource administrator, and the virtual machine image repository is located in the Virtual Infrastructure Manager (VIM) as shown in Fig. 3.

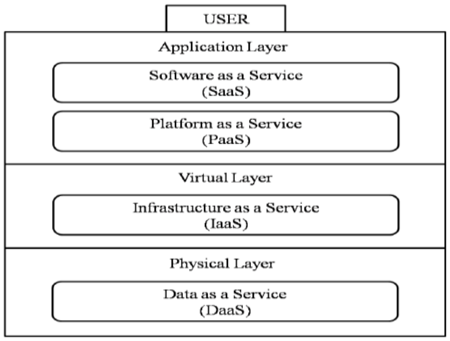


Fig. 2. Service model of cloud computing.

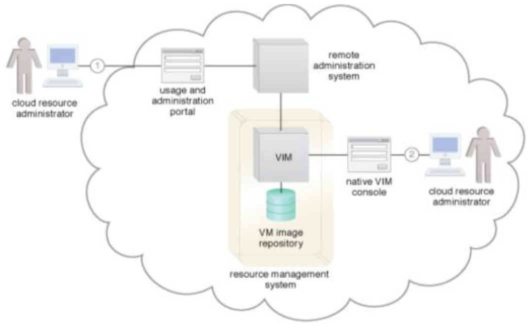

cloud resource administrator
1
usage and administration portal
remote administration system
VIM
native VIM console
2
cloud resource administrator
VM image repository
resource management system

Fig. 3. Resource management architecture.

E. Primary IaaS Systems

The OpenStack platform is a popular open-source cloud management system that is used by tenants to access, build, and manage their resources through a web interface or command line clients [12]. The system is designed to be modular, with a central controller node and compute nodes that report on the status of deployed virtual machines as shown in Fig. 4. The cloud infrastructure is dynamic, and changes may occur at various levels including services, topologies, and traffic. There are four types of cloud deployment models - private, public, community, and hybrid clouds. In a cloud environment, tenants can easily deploy virtual machines and create services that are accessible to others [13]. The cloud infrastructure is dynamic and changes happen frequently, including topology-related events caused by virtual machine life cycle commands, and traffic-related events due to changes in network demand.

F. Virtualization

The three main architectural layers in an IaaS system are the physical layer, the hypervisor, and the virtual machine layer. The security monitoring architecture focuses on the virtual machine layer. There are four main server virtualization strategies: emulation, full virtualization, par virtualization, and OS-level virtualization. OS-level virtualization uses containers and is a popular option for minimal overhead. Network virtualization is crucial for IaaS cloud architecture and manages IP addresses and communication. Network virtualization is achieved through MPLS, VLANs, Flat Networking, and GRE encapsulation, and is simplified by Software Defined Networking (SDN), with OpenFlow being the most popular example. The SDN architecture separates the control and data planes and allows for centralized control of the network [14][15] as shown in Fig. 5.

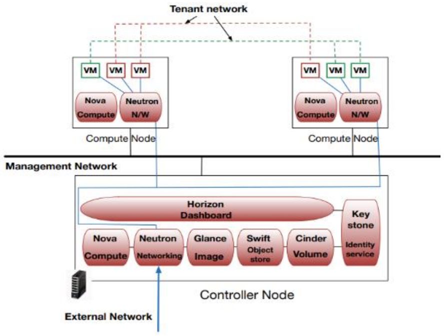


Fig. 4. The modular architecture of openstack.

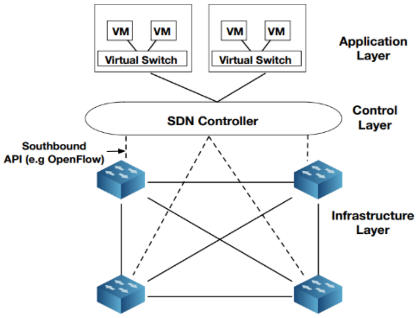


Fig. 5. The architecture of SDN.

SDN (Software Defined Networking) is a network management concept that allows for programmatic control and administration of the network through a programmatic interface. Examples of SDN controllers include OpenDaylight and Floodlight. Network virtualization in IaaS (Infrastructure as a Service) clouds allows for programmatic creation, modification, and deletion of network objects (such as networks, subnets, ports, etc.) without affecting the underlying hardware infrastructure [16]. The Neutron component of OpenStack is responsible for managing tenant networks and providing VMs with networking capabilities, with the ML2 plugin creating virtual bridges to connect VMs to networks with either GRE encapsulation or VLAN tagging to differentiate network traffic among tenants. In a typical cloud implementation, three types of networks are created: management network, tenant networks, and external network.

G. Security Threats

Different types of security threats to information systems, including the application level, network level, operating system level, and cloud environment. Threats can include SQL injection attacks, cross-site scripting, buffer overflows, impersonation, denial of service attacks, man-in-the-middle attacks, and exploitation tactics. In cloud environments, both tenants and providers face security risks such as risks from other tenants, the provider's infrastructure, and the API. Multi-tenancy can also create new security threats, such as side channel attacks and exploitation of shared resources. The security of virtual machines in a cloud environment is dependent on the stability of the hypervisor, which can be a target for security vulnerabilities and malware attacks. A successful attack on the control interface can result in full compromise of the account and stored information [17].

H. Security Monitoring

Information systems are continuously threatened at several layers of their infrastructure. To prevent significant damage, it's crucial to have a security monitoring system in place, such as an Intrusion Detection System (IDS). IDSs are the main component of a security monitoring framework and they detect security holes by collecting, processing and reporting data. There are two types of IDSs: signature-based and anomaly-based. Signature-based IDSs use string comparison to match observed events to known attack patterns, while anomaly-based IDSs compare observed events to a normal profile of activity to detect potential security breaches. Additionally, there are two types of IDSs based on location: network-based and host-based. Network-based IDSs monitor network traffic, while host-based IDSs monitor a single host for suspicious activity [18]. In cloud environments, security monitoring must be automatic to adjust to changing events. Different cloud security monitoring strategies focus on either the tenant's data system or the provider's infrastructure. Provider infrastructure monitoring includes hypervisor or kernel-based IDSs as shown in Fig. 6. The trend in hypervisor security is to lower the Trusted Code Base, but this does not guarantee the complete integrity of the system.

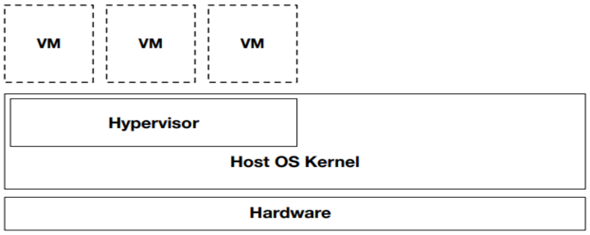


Fig. 6. The kernels of the host and hypervisor.

In the context of security monitoring in cloud environments, there are two main areas of focus: the provider's infrastructure and the tenant's information system. The provider's infrastructure is monitored by hypervisor or kernel-based intrusion detection systems (IDSs), which focus on securing the integrity of the hypervisor and kernel [19]. However, these systems may not adapt to changes in the programs running in the monitored system. On the other hand, the tenant's information system can be monitored through virtual machine introspection, which allows for real-time monitoring of the health of the underlying operating system and active processes in deployed virtual machines. However, designing cloud-tailored intrusion detection systems for the tenant's information system can be challenging due to the complexity and variety of the cloud environment and the conflicting security requirements of tenants.

It is evident from the literature review that there are still several areas that require further research, including the development of standardized SLAs, the development of cloud computing architectures for a wider range of applications, the effectiveness of cloud computing security measures, and the environmental impact of cloud computing.

III. Design and Methodologies

Eucalyptus is an open-source software platform used to create an IaaS (Infrastructure as a Service) for private or hybrid cloud settings. It is scalable and distributed, with each component serving a small number of users, making it suitable for enterprises of all sizes. The platform offers virtualized cloud resources like infrastructure, network, and storage as a service. The name "Eucalyptus" stands for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems.

A. Setting up Eucalyptus Cloud

Eucalyptus cloud setup consists of the Cloud Controller (CLC) and Walrus, which manage various clusters of real computers that host virtual instances. Each cluster has a Cluster Controller (CC), a Storage Controller (SC), and multiple physical machines known as Nodes [20]. The Node Controller regulates the hypervisor on each node to manage virtual instances. In the study mentioned, all components were co-located on the same system except the NC, with one machine hosting CLC, Walrus, CC, and SC, and five machines hosting NC as shown in Fig. 7.

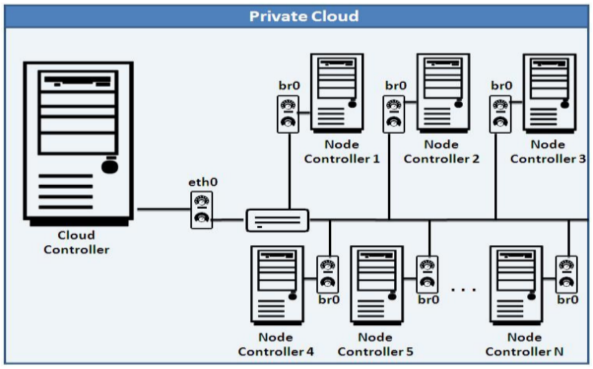


Fig. 7. The NC service is hosted by the domain-0 kernel in the xen setup.

B. IaaS Clouds based Self-Adaptable Framework for Monitoring Security

We consider an IaaS cloud environment with a global cloud Tenants have control over a networked group of virtual machines (VMs) and can specify unique monitoring requirements through a Service Level Agreement (SLA) or API. The cloud controller provides networking capabilities and the tenant receives both an external and internal IP address. The study focuses on software attacks from inside or outside the cloud infrastructure and the potential for an attacker to exploit a deployed VM and compromise the victim's infrastructure [21]. Trust is placed in the cloud provider's infrastructure being physically safe and not affected by malicious viruses. Attacks that weaken the cloud management system are not taken into account.

C. Goals Designed

The objective of the research is to develop a self-adaptive security monitoring platform for virtualized information systems of tenants that meets several requirements including self-adaptation, tenant-driven customization, security and accuracy, and cost savings. The framework should be able to update its components automatically in response to changes in the cloud environment and to allow for customization based on tenant needs [22]. It should minimize security risks and costs for both the provider and the tenants. The framework should consider the different sources of adaptation, including changes in services, topology, monitoring load, and tenant requirements. The framework should ensure that reconfigurations do not compromise security or monitoring accuracy, while minimizing resource use and performance impact on tenant applications.

D. Methods

The research presents a self-adaptive security monitoring framework for IaaS cloud with a three-tier architecture including a controller, two computing nodes, network IDS, edge firewall, local firewalls, and a log aggregator. The three tiers are the tenant, adaptability, and monitoring devices with the monitoring devices consisting of log collectors, aggregators, and probes as shown in Fig. 8.

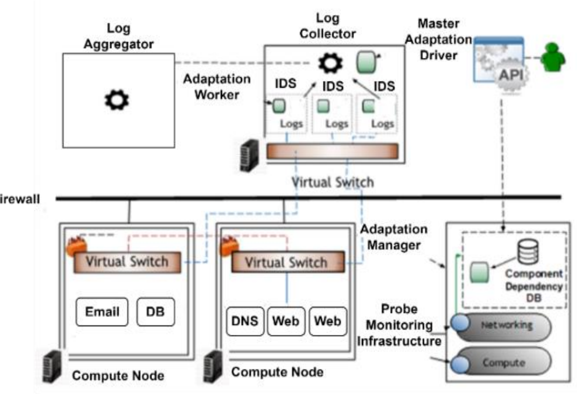


Fig. 8. The framework's architecture [23].

The research presents a self-adaptive security monitoring framework for IaaS cloud with three main tiers: tenant, adaptability, and monitoring devices. The adaptation level is responsible for planning and enforcing the adaptation process and is made up of the Adaptation Manager in the cloud controller, Master Adaptation Drivers in the nodes, and a dependency database. The Adaptation Worker in each monitoring device enforces the reconfiguration settings and the Infrastructure Monitoring Probes discover topology changes. The tenant API provides access to all monitoring features as shown in Fig. 9.

The research presents a self-adaptive security monitoring framework for IaaS cloud with a controller, two computing nodes, network IDS, edge firewall, local firewalls, and log aggregator. The framework has three main tiers: tenant, adaptability, and monitoring devices. The tenant tier has access to a tenant API, which allows tenants to express their monitoring requirements in a high-level language. The API is broken down into two parts: the tenant-exposed part and the translation part. The tenant-exposed part allows tenants to access the list of monitoring services, add or remove a monitoring service, and alter monitoring metrics. The translation part translates the high-level tenant requirements into framework-specific information, which is used by the framework to make adaption decisions.

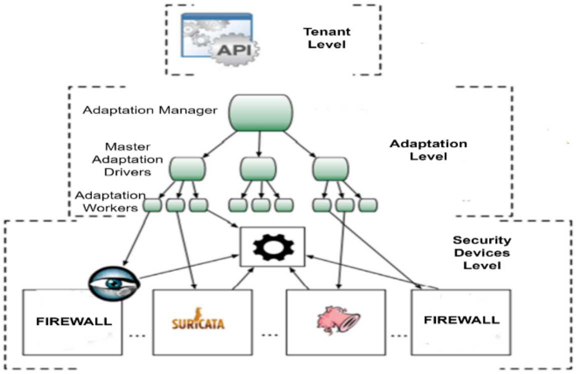


Fig. 9. The framework’s various levels [23].

Example scenario: Consider the following instance as an illustration: On various compute nodes, there are two unique VMs installed, each of which belongs to a different tenant. An ssh server and an Apache server with SQL support are hosted by the first VM with ID 24 that is installed on node A. Although its private IP address is 192.168.1.3, the virtual machine's public IP address is 182.12.34.201. A port with the name "qvo1432" is used to link the VM to the compute node's virtual switch. On node P-20, the second VM, ID 63, is configured and is only used to host an ssh server as a service. The VM's secret IP address is 192.168.1.3, whereas its public IP address is 182.12.34.199. In this condensed example, we solely use firewalls and network-based IDSs as monitoring tools.

The security of a cloud computing infrastructure is monitored by security devices such as firewalls, vulnerability scanners, antivirus programs, and others. These devices are able to create log files that are managed by log collectors and aggregated by the framework administrator for searching for specific patterns. The Adaptation Manager is a key component of the monitoring framework that is responsible for selecting the adaptations to the monitoring tools, maintaining an acceptable degree of monitoring, and managing dynamic events within the cloud architecture. The Adaptation Manager performs algorithms in Algorithm 1 in response to dynamic events and makes decisions on whether to adapt based on the topological and functional overviews of the monitoring framework.

Algorithm 1 The choice algorithm for adaptation

```
1: procedure ADAPTATIONS (dynamic activity)
2:     services list ← MAP (dynamic activity.VMid, vm info file)
3:     affected equipments, agents ← MAP (dynamic activity.VMid)
4:     for j in affected equipments do
5:         reconfiguration needed ← DECIDE (j, services list)
6:     PROPAGATE DECISION (agents, reconfiguration needed)
```

- Connect the ID of the VM that is affected by the modification to the list of services that are currently executing within the VM (line 2 in Algorithm 1). To do this, the data in the file that the API generated is parsed.
- Determine the monitoring systems in charge of the impacted VM. These will be the monitoring tools that are modified. Utilizing data from the Component Dependency Database, this is accomplished. The vm information file is a single file that contains both the list of monitoring devices that will be updated and the list of active services.
- Select the necessary reconfiguration type (line 5 in Algorithm 1). Different reconfiguration kinds may be required depending on the monitoring device type and incident category.
- Distribute the reconfiguration parameters to the personnel in charge of upholding the adaption choice (line 6 in Algorithm 1).

E. *Infrastructure Monitoring Probes*

IMPs (Instance Monitoring Processes) are components of the cloud engine that monitor topology changes, such as placement and VM lifecycle alterations. They collect VM-related data from the cloud engine and provide information to the adaptation manager, which then decides which adaptation to deploy. IMPs do not affect normal cloud operations when adjusted.

F. *Component Dependency Database*

The article discusses the challenges of security issues in complex security monitoring frameworks composed of various components. The methodology states that a decision taken in response to a dynamic event can impact both an active and passive monitoring device, requiring both to be changed. The Dependency Database, a component of the cloud controller, lists all security devices for each monitored VM and gives the Application Manager both functional and topological viewpoints. The VM information table can be used by the Adaptation Manager as a key to access the list of monitoring tools in charge of a VM, making it easier to identify impacted devices during an adaptation. The VM information table for the VMs in the previously discussed scenario is found in Table I.

For example, the VM with ID 24 can see a host IDS, a network IDS, and two firewalls: one inside the local switch called f-p-20 and one on the edge called f-ext1. Multiple IDS types can monitor a single VM (host- and network-based).

TABLE I. THE VM INFO TABLE

VM ID	Network IDS	Host IDS	External-firewall	Switch-firewall
24	S-79	24	Mar24	f-p-20
14	S-99	14	Aprs14	f-p-63

TABLE II. THE EQUIPMENT INFO TABLE

Equipment Name	Location	Equipment Type
S-65	182.12.34.201	signature based

Device-specific information is kept in the equipment info table. The Adaptation Manager gathers the following data using each device name: both the device's location and its type Table II contains the S-65 IDS equipment information table. This instance demonstrates that the signature-based NIDS suricata65 is situated on a node with the IP address 182.12.34.201. The Dependency Database's information is used by the AM to compile a complete list of all devices that are impacted by a dynamic event. The AM updates the two tables with a corresponding entry for every new monitoring device that is created, including all the relevant data.

G. *Self-Adaptable System for Intrusion Detection in IaaS Cloud*

A self-adaptable intrusion detection system has been proposed for IaaS cloud environments. This system offers features such as self-adaptation, customization, scalability and security, and correctness. The system allows for adaptation to changing conditions in the cloud, enables tenants to modify monitored events, and adjusts the number of IDS systems based on network traffic and infrastructure size. The system ensures proper level of detection during adaptation to maintain security and correctness.

1) Proposed framework: Detailed design of a proposed framework, including its components and the system and threat model used. It starts with a general overview of the framework, followed by a detailed explanation of how each component works.

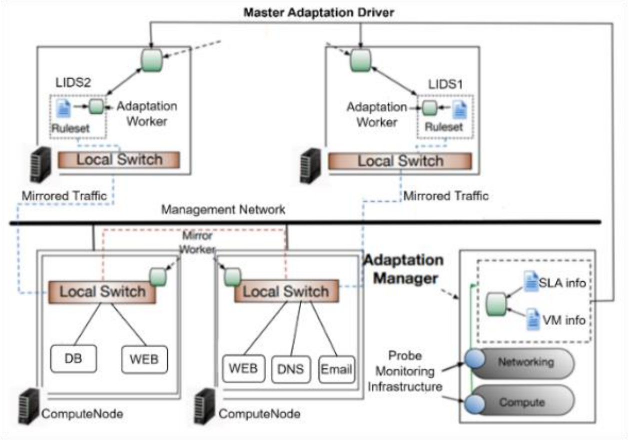


Fig. 10. The proposed framework for self-adaptable system.

The framework described in Fig. 10 is composed of four main parts: Adaptation Worker (AW), Local Intrusion Detection Sensors (LID), Mirror Worker (MW) and Master Adaptation Driver (MAD). LIDs are deployed on separate nodes and are used to collect and analyze network packets, either using anomaly-based or signature-based techniques. The AW updates the applicable ruleset, monitors the performance of the LID, and updates the MAD after a successful reconfiguration [24]. The MAD handles the reconfiguration of multiple LIDs, manages their lifecycle, and collects performance metrics. The MW ensures accurate mirroring of traffic to the matching LID node and constructs a mirroring endpoint if needed. A safety mechanism on each compute node ensures that VMs don't become active before the accompanying LID has been reconfigured.

The proposed framework for security in cloud computing has both potential weaknesses in its architecture and contributions to the service provider's infrastructure. A vulnerability in the framework is the configuration files that translate adaptation arguments into rule names, which are simple text or XML files and could be manipulated by attackers to produce fake adaptation arguments. The framework includes parsers for these files. The power utilization model evaluates the power usage of physical machines and is based on power meters built into the physical machines [25]. Research has shown that power consumption is inversely correlated with the number of fully utilized cores. The migration model describes the methods and costs of relocating virtual machines as shown in Fig. 11.

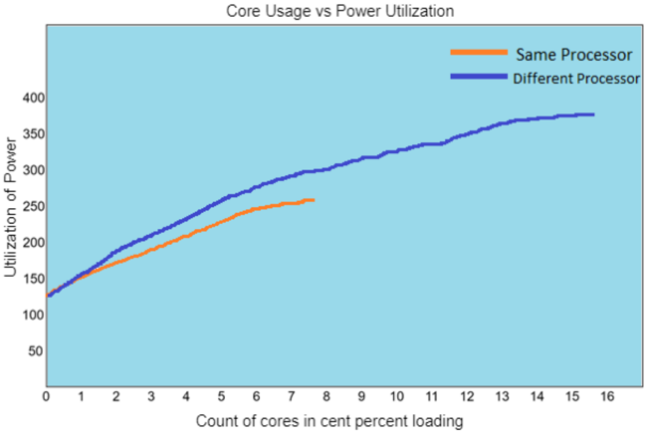


Fig. 11. Power utilization demonstration with various processor loads.

The Top power (Tp) is the maximum power used by a PM (Processing Module) when all 16 cores are fully loaded. The Inactive power (Ip) is the power consumption when none of the cores are loaded. The Top power is about half of the Inactive power. The linear power model is used to fairly assess the power utilization (PU) of PM.

$$PU = \left[\frac{C_v}{C_p}(1-\beta)+\beta\right]T_p \qquad (1)$$

Where β displays the ratio of dormant to active power. Consider Fig. 11 above, β is set to half, Cv is the total number of cores required by resident VMs, and Cp is the total number of cores required by PM.

The Migratory technique uses live migration to allow a server administrator to move an active virtual machine (VM) to a different PM without affecting its performance. Although live migration does not affect the execution time of the VM, it does increase energy consumption during the transfer due to increased load on the receiving PM [26]. The energy consumption during relocation has been quantified based on experimental data.

$$U_E = U_S + U_D \qquad (2)$$

$$U_S = \left[\frac{C_v}{C_p}(1-\beta)+\beta\right]T_p M_t \qquad (3)$$

$$U_D = \left\{\left[\frac{C_v'}{C_p}+L\right](1-\beta)+\beta\right\}T_p M_t \qquad (4)$$

The U_E is the percentage load of the sending and receiving machines. Cv is the number of cores required to migrate a VM, and Cv/Cp and Cv'/Cp are the percentage load of the sending and receiving machines. The migration time (Mt) and additional burden (L) imposed by the migration procedure must be considered when deploying VMs with various hardware requirements on PMs with specific hardware limitations. The First-Fit approach may be used to address the fusing problem, though it is more challenging. It seeks to deploy a VM to the first PM that has space for it and can be simple to apply, but may not be the best solution.

This study investigates a more challenging VM fusing problem where each VM has resource requirements, Time of Execution and Time of Arrival restrictions, and a limited availability window during its execution. The idle cores used by a VM after it has ceased to function can lead to the need for additional PMs. The Eucalyptus architecture provides two scheduling options, Enhanced Round-Robin (ERR) and a Fusion mechanism, to lower the number of PMs needed and save power consumption. The ERR approach uses two criteria to aid in VM fusion: retiring PMs cannot have additional VMs added to them, and retiring PMs that cannot complete all VMs before the deadline must transfer and shut down. The Fusion method combines the ERR and First-Fit approaches, depending on the rate of incoming VMs, to save more energy. The biggest challenge in implementation is choosing a reasonable Limit of Resigning (LR), calculated by adding migration time and time left for the VM to execute. The Fusion method with LR calculation can result in energy savings by transferring VMs during non-peak hours.

$$U_S = \left[\frac{C_v}{C_p}(1-\beta)+\beta\right]T_p R_t \qquad (5)$$

$$U_D = \left[\frac{C_v'}{C_p}(1-\beta)+\beta\right]+T_p R_t \qquad (6)$$

Where U_E and U_D are, respectively, the energy expenses associated with sending and receiving PMs when the VM is not being migrated. Instead of deciding to relocate every VM by its remaining time for execution after the PM resigned, Rt, that's a mystery, the limit α below is the resignation limit.

$$\alpha = (1-L)M_t + \frac{L}{\beta} \qquad (7)$$

Assuming a VM with Rt smaller than indicates that not migrating saves more energy, and the VM will eventually finish before PM starts moving VMs. On the other hand, in the unlikely event that Rt greater than α, VMs will be transferred after the resigning limit is exceeded, indicating that moving is the better option. Execution time that is left over Rt While using the ERR approach, it is not essential. The limit α has been established and fixed. After a brief pause, α., a resigning VM will be shut down. Regardless of how much time is left for execution, each incomplete VM would be transferred.

2) Self-Adaptable framework for monitoring security: The proposed framework consists of four main components: Adaptation Worker, Local Intrusion Detection Sensors, Mirror Worker, and Master Adaptation Driver. The Adaptation Manager, a part of the proposed framework, is implemented using a multi-threaded model in Python, using Open-Stack as the cloud management system [27] and Open vSwitch for network traffic mirroring. The AM receives notifications of topology changes from the Infrastructure Monitoring Probes and creates a worker thread to manage the potential adaptation event as shown in Listing 1. The worker thread reads information about the impacted guest VM from a vm information file and retrieves the list of active services and tenant-specific security requirements using the VM ID as an identifier.

Listing 1 Adaptation during VM migration

```
procedure ADAPTATION (VM network information)
    SPAWN ADAPTATION THREAD
    services list ← INFORMATION PARSER (VM network info.VM id, vm information file)
    affected equipments, locations ← INFORMATION PARSER (VM network info.VMid, VM network info.source node,VM network info.destination node, topology.txt)
    for p,qj in affected devices, locations do
    args.txt ← DECIDE (services list, p)
    IDS CONN (q, args.txt, +/-)
```

The parameters (such as what types of rules will be activated/deactivated, what is the acceptable tenant drop rate, etc.) are put to a specific file called args.txt when the AM decides to adapt. A separate file (topology.txt) that contains the topological and functional views required by the AM is extracted by the worker to obtain the names, kinds, and locations of the impacted security probes.

One NIDS is used on each computing node in the monitoring technique shown in this section. The single node hosts all of the deployed NIDSs. Following a VM migration, the master thread receives network-related information from the IMP, for example for the VM with ID 24.

The worker thread parses the topology.txt and vm information file.xml files as soon as it receives this information to extract the services that are currently running in the migrated VM (sshd, apache2, sqld), any additional tenant-defined monitoring requirements (worm), tenant-specific monitoring metrics, and finally the names of the NIDS responsible for monitoring the traffic in the source and destination nodes (S-9 and S-65, respectively), as well as their host IP. Adaptation is necessary for these two NIDS. The adaptation parameters are then written by the worker thread to adaptation args.txt. Listing 2 displays the findings of the NIDS monitoring traffic to and from the destination. Listing 2 shows the file holding an NIDS's adaption arguments.

Listing 2

```
signature based
S-65
apache 2
sql
ssh 192.168.1.1, 192.168.1.3
worm
5
```

- Using a secure connection, the worker thread sends the dedicated file to a MAD in the node(s) holding the affected security devices. The name and IP address of the node housing the security device are used to create a connection by the ids conn function.
- The file containing the adaptation arguments must additionally contain a specialised operator if the adaptation calls for the activation or deactivation of monitoring parameters (such as + or -), as stated by the AM. In our illustration, a + denotes that the monitoring parameters need to be activated by the operator that was sent with the file in Listing 2. To allow for the concurrent transmission of the adaptation file, a separate thread is created for each security component that will be touched by the adaptation choice.

H. *Self-Adaptable System for Intrusion Detection*

A private cloud framework prototype was created using OpenStack and Open vSwitch as a virtual switch. The framework uses GRE tunnels to separate VMs into tenant networks and mirrors traffic using signature-based LID nodes in Docker containers [28]. An Adaptation Worker was developed to manage the LID nodes and communicates with the Master Adaptation Driver (MAD) using a shared folder. MAD uses a multithreaded method to manage and reconfigure the LID nodes and includes configuration files to convert adaptation parameters into rules for the IDS. The plug vifs function was used to delay the creation of virtual interfaces until the LID reconfiguration is finished to ensure network access for the VMs. [15].

IV. Experimental Results and Discussions

The experiment was performed on an eight-node cluster, each with a quad-core CPU and a power model that uses half of peak power in idle mode. The cost of migration was calculated as 0.025 * 1/8 * 0.2. Data for VM arrival and execution times, and whether they were small or large-scale, was used in the test. The arrival and execution times were estimated using a normal distribution of two to ten hours. Small-scale tests were conducted to study power and migration models using the Xen hypervisor on each node. The power usage was measured using the proposed power model and actual power consumption and an average power usage was calculated. The results of the proposed ERR strategy were compared to the First-Fit power-saving strategy and showed little variance between the estimated and measured power. Thus, different scheduling techniques can be evaluated using the results of the recommended model as shown in Table III.

A simulation was conducted to evaluate the performance and energy consumption of a system with 500 octa-core servers and 3000 virtual machines. The impact of different resigning limitations (10, 20, 30) was tested, and the results of energy usage were compared between the ERR method and the RR and First-Fit methods. The results showed that power usage decreases as the resignation limit decreases and are represented in a graphical form in Fig. 12, with First-Fit as the baseline.

TABLE III. Power Utilization ERR and First-Fit Power Save Results

	Estimated Power (W)	Power Meters Provide Actual Power
ERR	744.53	695.20
Save with First-Fit Power	698.30	659.28
Improvements were made.	46.23	35.92

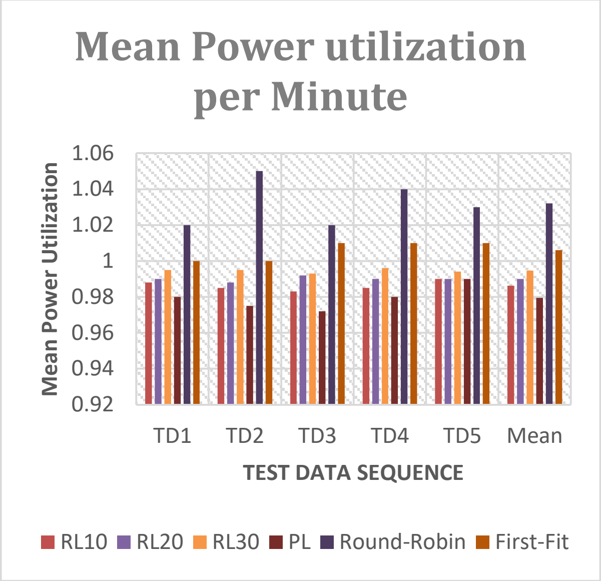


Fig. 12. Using ERR to calculate mean power utilization under different limits of resigning.

This simulation compares five different energy-saving solutions (Best-Fit, RR, First-Fit, ERR, and Fusion Method) by considering metrics like mean power use, mean PM count, and migration count. During busy hours, the Fusion Methodology is used. A graphical analysis is shown in Fig. 13 which demonstrates that the Fusion Methodology performs better in terms of energy conservation compared to the other approaches, as it uses PM loads as the resignation limit.

The paper compares the performance of five different algorithms for VM Scheduling in Eucalyptus cloud: RR, Greedy, PowerSave (similar to First-Fit), ERR and Fusion Method. The analysis compares the number of powered-on PMs and power consumption for each of these algorithms, with the results presented in Fig. 14 and 15. The paper shows that the use of the recommended strategies (ERR and Fusion Method) leads to a significant reduction in the number of powered-on PMs and power usage compared to the three basic Eucalyptus scheduling algorithms (RR, Greedy, and PowerSave).

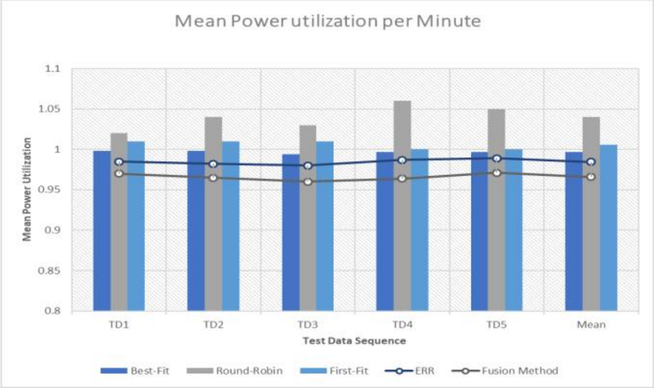


Fig. 13. Comparing the proposed technique's mean power utilization to that of other techniques.

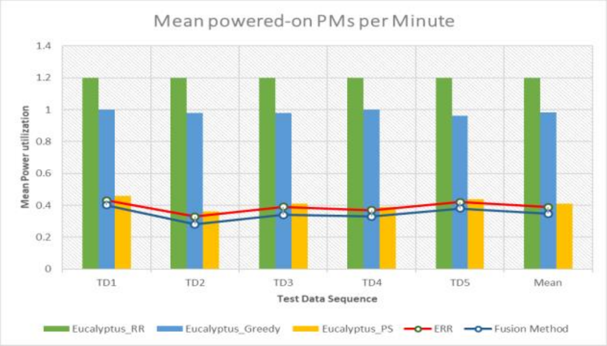


Fig. 14. Usage of RR base to analyze mean powered-on PMS.

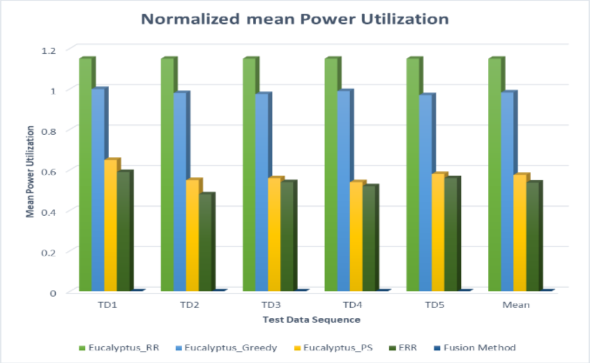


Fig. 15. The usage of RR base to analyse the normalised mean power utilisation.

A. *Eucalyptus with Walrus*

This section discusses the private cloud patterns controlled by Eucalyptus and the use of WALRUS, a data storage service for customer data. The Eucalyptus web interface supports "admin" and "user" accounts, and after registration, customers receive X509 certificates, secret key, and Query Id. The client credentials, including RSA private and public keys and X509 certificates, are stored in a file called "eucarc". WALRUS can be accessed through SOAP or REST via HTTP with the help of various utilities and is managed by ACLs and client credentials [10] [29] as shown in Fig. 16.

WALRUS is a data storage service used in Eucalyptus and can be accessed using tools like s3cmd, s3curl, s3fs, cloud berry s3 via SOAP or REST via HTTP. Access to data stored in WALRUS is managed by Access Control Lists and client credentials and secured using the MD5 hashing method. This section also mentions potential security attacks on private cloud systems powered by Eucalyptus that target cloud databases.

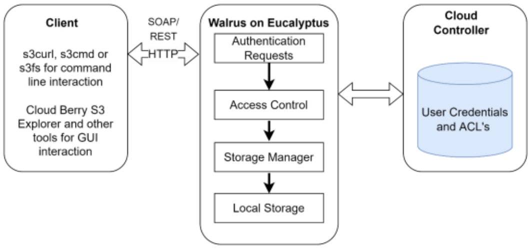

Client
s3curl, s3cmd or s3fs for command line interaction
Cloud Berry S3 Explorer and other tools for GUI interaction
SOAP/ REST HTTP
Walrus on Eucalyptus
Authentication Requests
Access Control
Storage Manager
Local Storage
Cloud Controller
User Credentials and ACL's

Fig. 16. Architectural demonstration of Eucalyptus-WALRUS.

1) Attacks related to buckets: The Eucalyptus private cloud system can be vulnerable to security attacks targeting cloud databases. Fig. 17 shows the credentials used for querying user interface names, such as EC2_SECRET_KEY and EC2_ACCESS_KEY, which are stored in the AUTH USERS table with attributes AUTH_USER_SECRETKEY, AUTH_USER_QUERY_ID, and AUTH_USER_NAME, respectively. One specific attack is the use of the file "eucarc," which is obtained from the eucalyptus auth.script catalogue. As shown in Fig. 18, an attacker must upload a new file called "eucarc" along with an S3 URL set that includes the IP address of the cloud controller and the values of EC2_SECRET_KEY and EC2_ACCESS_KEY, which are represented by AUTH_USER_SECRETKEY and AUTH_USER_QUERY_ID, respectively.

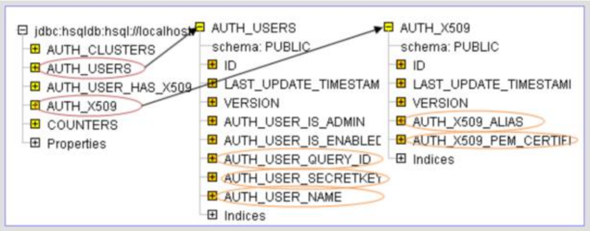
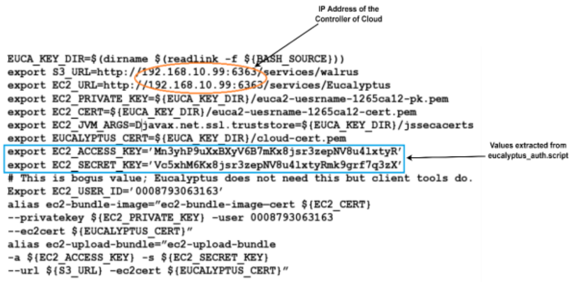


Fig. 17. Illustrations from the table script for Eucalyptus_auth.



```
EUCA_KEY_DIR=$(dirname $(readlink -f ${BASH_SOURCE}))
export S3_URL=http://192.168.10.99:6363/services/walrus
export EC2_URL=http://192.168.10.99:6363/services/Eucalyptus
export EC2_PRIVATE_KEY=${EUCA_KEY_DIR}/euca2-uesrname-1265ca12-pk.pem
export EC2_CERT=${EUCA_KEY_DIR}/euca2-uesrname-1265ca12-cert.pem
export EC2_JVM_ARGS=Djavax.net.ssl.truststore=${EUCA_KEY_DIR}/jssecacerts
export EUCALYPTUS_CERT=${EUCA_KEY_DIR}/cloud-cert.pem
export EC2_ACCESS_KEY='Mn3yhP9uXxBXyV6B7mKx8jsr3zepNV8u4lxtyR'
export EC2_SECRET_KEY='Vc5xhM6Kx8jsr3zepNV8u4lxtyRmk9grf7q3zX'
# This is bogus value; Eucalyptus does not need this but client tools do.
Export EC2_USER_ID='0008793063163'
alias ec2-bundle-image="ec2-bundle-image—cert ${EC2_CERT}
--privatekey ${EC2_PRIVATE_KEY} -user 0008793063163
--ec2cert ${EUCALYPTUS_CERT}"
alias ec2-upload-bundle="ec2-upload-bundle
-a ${EC2_ACCESS_KEY} -s ${EC2_SECRET_KEY}
--url ${S3_URL} -ec2cert ${EUCALYPTUS_CERT}"
```

Fig. 18. The “eucarc”: credentials compressed file constituent.

The remaining parts of the "eucarc" paper might be omitted because they are not necessary for attacks involving buckets. The attacker essentially needs to obtain the most recent version of the eucarc document after it has been prepared and use the command s3curl to create a bucket that impersonates the client whose login details are used, or to gain access to a significant number of buckets that the client has reserved.

2) Attacks related to objects: Before launching attacks on an object, an intruder must be aware of the precise name of the bucket containing the target object. The eucalyptus-walrus can be used in two distinct ways to determine the precise name of a bucket. One possibility is the script catalogue. Fig. 19 shows how the names of the parent bucket, the individual objects, and the bucket owner are all shown in the OBJECTS table of this catalogue under the attributes OBJECT KEY, BUCKET NAME, and OWNER ID.

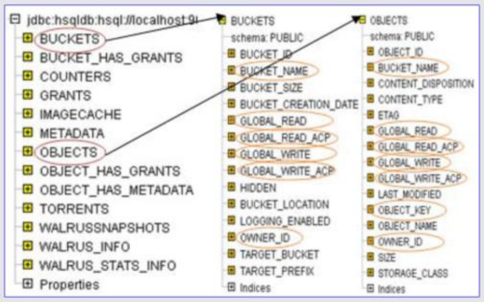


Fig. 19. Eucalyptus-walrus.script table illustrations.

The attacker must use the victim's interface certificates to create a new eucarc document and use s3curl to insert an object into the victim's bucket. The attacker must determine the object's size, MD5 checksum, and last update time and decide whether to read or delete the object. However, these actions require the owner to be an "administrator."

3) Attacks related to ACLs: In Eucalyptus, every WALRUS object and bucket has an associated Access Control List (ACL) in the form of a sub-resource. To launch ACL attacks, the attacker must first have access to ACL-related subresources. The ACL can be obtained using the s3curl command, and can be modified or a new one can be created with desired access control privileges. Attackers can also exploit the distribution of access control privileges to all cloud-registered users. The attacker can change the ACL file to grant access to all users by setting attributes in the eucalyptus walrus.script catalogue.

4) Attacks related to log file: Customers in Eucalyptus can create access logs for their buckets and decide where to send the logs. The logs can be treated like other objects, with the ability to list, remove, and read them. The logging data is stored in the eucalyptus walrus.script catalogue's properties TARGET PREFIX, TARGET BUCKET, and LOGGING ENABLED. If LOGGING ENABLED is set to TRUE, the TARGET PREFIX will add the client-specified prefix to the end of the log file names, and the TARGET BUCKET will be the client-selected bucket where the access log files will be saved. An attacker who gains access to the bucket containing the log entries can use the log entries as they see fit.

B. Analysis of Self-Adaptable System for Intrusion Detection

A data center with five physical nodes (1 controller, 1 network node, 2 compute nodes, and 1 LID host node) running Ubuntu Server 14.04 connected by a 1Gb/s network was set up on the Grid5000 platform for testing. Experiments were conducted using a memory-intensive workload with a 1024MB working set and 10 executions of the LMBench benchmark suite. The proposed framework's overhead during VM migration was tested with two new rule types related to ssh traffic.

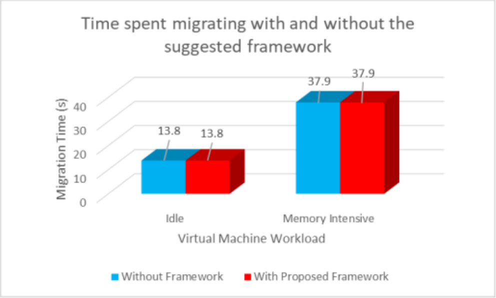


Fig. 20. Time spent migrating with and without the suggested framework.

The outcomes are displayed in Fig. 20. Our initial expectation that the proposed framework imposes minimal overhead on ordinary cloud operations is confirmed by the fact that the imposed overhead in both scenarios of an idle virtual machine and 0.0s represents a virtual machine with a memory-intensive workload. Fig. 21 and Fig. 22 display a breakdown of the two separate adaptation instances (new LIDS with traffic distribution and ruleset reconfiguration only) per phase.

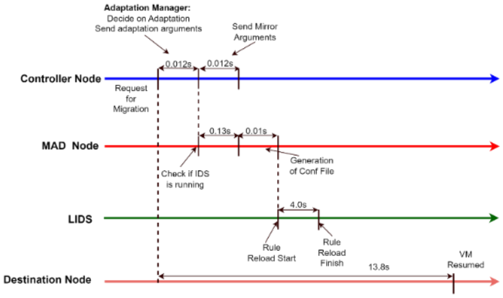


Fig. 21. Breakdown of the adaptation time when the proposed framework only modifies the imposed inside the LIDS ruleset.

Both instances involve the safety mechanism being in operation, however, when the plug vifs is called, the LIDS reconfiguration is finished significantly more quickly (4.14s and 0.97s respectively while the plug vifs function is called always after the 10th second).

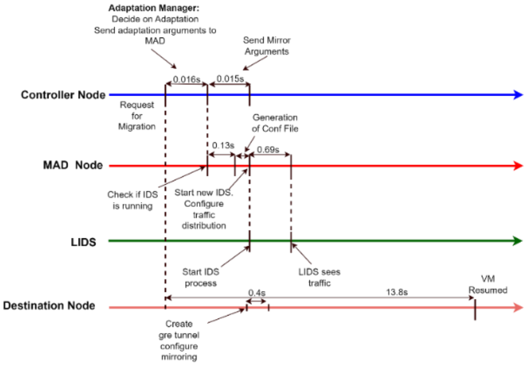


Fig. 22. A breakdown of how long it takes framework to build a mirroring tunnel, distribute traffic, and start a new LIDS.

The proposed framework reduces waiting time when restarting a virtual machine (VM) and completes a full adaptation cycle faster than migration. It takes 4.14 seconds to reconfigure the imposed ruleset in the first scenario and 0.97 seconds to access traffic in the second scenario when a new LID needs to be created. The total time needed by the framework is less than migration and creating a new IDS is easier than changing an old one.

Multiple LIDSs and MADs: A specialized script is created to replicate migration events by creating precise inputs from the Infrastructure Monitoring Probe for the Adaptation Manager. This allows for concurrent production of multiple adaptation events. During experimentation, a single instance of the Master Adaptation Driver (MAD) is configured to handle multiple LID instances, with the maximum number theoretically handled by a single MAD instance shown in Fig. 23.

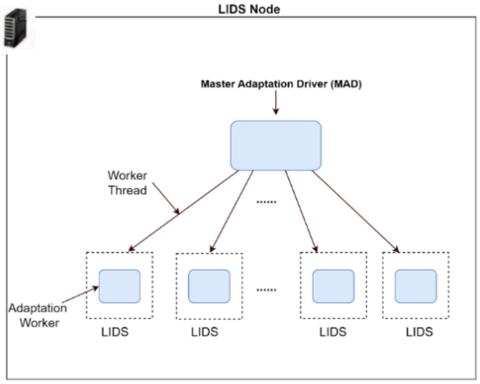


Fig. 23. Setup for MAD scalability.

Our findings demonstrate that up to 50 LIDS can be supported by a single MAD instance running on a dedicated node with 24GB RAM. Fig. 24 displays the MAD agent's typical response time under various LIDS loads.

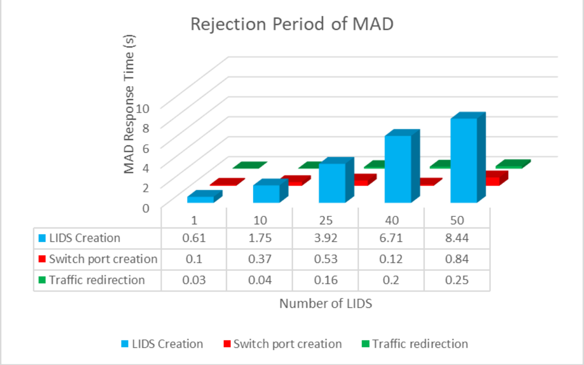


Fig. 24. MAD reaction periods.

The results show that the longest creation time for a new LID container requires interaction with the Docker daemon. The proposed framework can handle 50 simultaneous LID launching requests and still has a faster reaction time than the average migration time for an idle VM. The framework does not include the time for building the new LID configuration file or testing its functionality due to their low impact on the overall duration. Each LID is typically assigned one core in a production setting to maintain performance. The framework was tested with 10 concurrent adaptation requests to simulate a commercial setting. The maximum number of LID's that a single Master Adaptation Driver instance can manage simultaneously was determined, and there is room for nearly 100 worker threads for the Adaptation Manager. One AM instance is configured to control multiple MADs in Fig. 25.

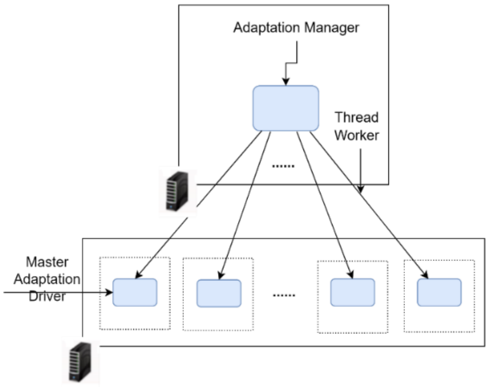
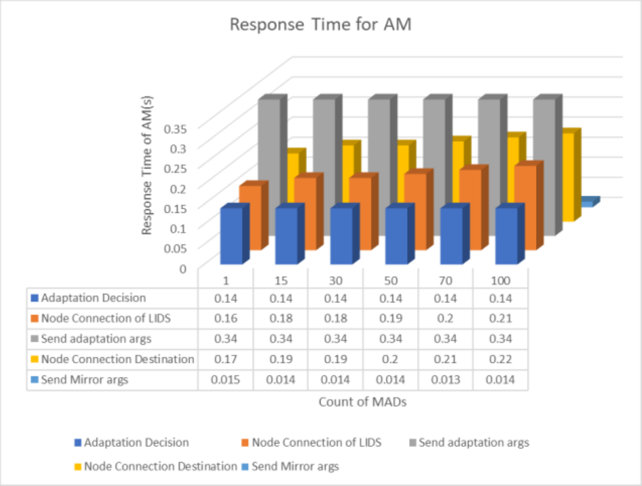


Fig. 25. AM scaling configuration.

The chosen monitoring mechanism for a single VM is a single LID. The framework was tested by simulating 50 dynamic events for 50 different VMs, providing adaptation requirements for 50 LID of each thread. A worker thread relocates all of its virtual machines to the same destination node to target the LID under the same MAD instance it is handling. The worker thread parses the vm info.xml file containing all the VM-related data to obtain settings for each of the 50 VMs it is responsible for. The minimum number of VM entries in the vm info.xml file required is calculated as maximum AM worker threads multiplied by the number of VMs per thread (in this case 100 AM worker threads and 50 VMs per thread, requiring 5000 entries). Justifications for each LID adaptation are recorded in a separate file. The worker thread delivers 50 files, one for each LID, to the MAD in charge of those 50 LID after establishing a secure connection. Only one file is required in the experiment as all virtual machines for a single worker thread are moved to the same compute node. The experiment aims to understand how the AM scales with the quantity of MADs, not the number of compute nodes.



	1	15	30	50	70	100
■ Adaptation Decision	0.14	0.14	0.14	0.14	0.14	0.14
■ Node Connection of LIDS	0.16	0.18	0.18	0.19	0.2	0.21
■ Send adaptation args	0.34	0.34	0.34	0.34	0.34	0.34
■ Node Connection Destination	0.17	0.19	0.19	0.2	0.21	0.22
■ Send Mirror args	0.015	0.014	0.014	0.014	0.013	0.014

Fig. 26. AM response time.

Fig. 26 presents the outcomes. As the data show, the formation of the secure connection is the phase that is most impacted by raising the MADs' load for the AM. This is due to the fact that each MAD has its own IP address and is kept in a distinct container, necessitating the use of a separate secure connection. On the AM side, we track the time it takes to send the adaption arguments. No network contention-related delay is seen in the outcome since we don't wait for each instance to validate that it has received the files. However, because each MAD instance is essentially run on a separate container on the same node, there may be many processes running on the node that are causing severe ssh connection formation delays. Since each MAD instance would operate in a distinct, less-loaded node in a real-world scenario, the findings of our experiment are unsatisfactory. The adaption decision time is not greatly shortened by the multi-threading method because all VM-related data is retained in a single file.

Up to 5000 LIDS instances can be managed by a single AM instance and still respond to thread requests in under one second, according to the results. The testbed's memory capacity is the single factor limiting the number of LIDS instances that can be employed for our research. If framework is implemented in a different configuration with production nodes having memory capacities that are substantially a minimum of 24 GB of RAM per node, the number of instances could rise.

The pidstat programme from the sysstat suite, a utility used to measure the resource utilisation of a given job running in an OS, is utilized to compute the resource consumption of an AM in terms of CPU and memory handling multiple MADs. Each experiment asks the first worker thread to run pidstat as soon as the adaption parameters are received, and the monitoring is stopped after the last worker thread has finished its task. This method ensures that during the adaptation process, we will only calculate the resource usage of each worker thread. No other framework-related processes use resources since the worker thread in charge of that adaptation request handles all the adaptation-related duties in that request. One second was chosen as the monitoring interval. The results are shown in Table IV. The Graphical illustration of resources consumption by AM is analysed in the Fig. 27.

The CPU use grows as the number of AM worker threads rise since there is a one-time CPU cost for starting a new ssh session. For each worker thread, the measurements compute the worst-case situation, which is to create a new connection. When the framework has to modify an existing LIDS, it can transfer the file containing the adaption arguments using an already established connection as a result of the anticipated decrease in CPU usage.

TABLE IV. CONSUMPTION OF RESOURCES BY THE AM COMPONENT

Number of MADs	Usr%	Sys%	CPU%	Memory (MB)
15	18.15	2.35	20.35	189.66
30	24.30	3.39	27.08	189.83
50	25.19	3.71	30.12	189.74
70	27.63	3.82	31.98	189.61
100	29.4	3.98	33.45	189.89

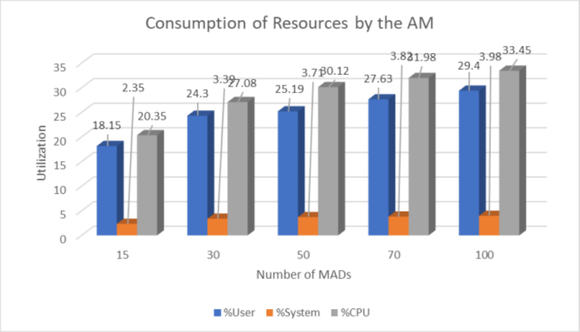


Fig. 27. Graphical illustration of resources consumption by AM.

The proposed framework for self-adapting security monitoring in cloud infrastructure is better than previous approaches for several reasons as below

1) Customization for each tenant: The framework takes into account the specific security monitoring requirements of each tenant, allowing for customized security monitoring configurations. This is an improvement over previous approaches that had a one-size-fits-all security monitoring approach that may not have been optimal for all tenants.

2) Self-adaptation: The framework includes an Adaptation Manager that can adapt to changing conditions in the cloud infrastructure and serve as a coordinator of the adaptation process. This allows the framework to respond dynamically to events in the cloud environment and adjust security monitoring configurations as needed. This is an improvement over previous approaches that did not have a self-adapting mechanism and required manual adjustments.

3) Cost efficiency: The framework aims to achieve cost efficiency by sharing monitoring resources between the tenants and the provider. The equitable sharing of monitoring resources is established with an adjustable threshold mentioned in the SLA. This is an improvement over previous approaches that may have been resource-intensive and costly.

4) Comprehensive solution: The framework provides a comprehensive solution that combines precise security monitoring with self-adaptation. The design of the framework ensures that it does not introduce new security weaknesses or affect the performance of the infrastructure. This is an improvement over previous approaches that may have been limited in scope or may have introduced new security weaknesses.

Overall, the proposed framework is an improvement over previous approaches because it is customized for each tenant, includes a self-adapting mechanism, achieves cost efficiency, and provides a comprehensive solution for security monitoring in cloud infrastructure.

V. Conclusion and Future Works

In this paper, the increasing number of tasks in clouds due to virtualization and cloud computing technology adoption is discussed. VM scheduling strategies are important for determining the allocation of cloud resources to handle these tasks for maintaining acceptable throughput and revenue. The paper evaluates the current knowledge on legacy methods and specific virtual machine scheduling algorithms for the Eucalyptus cloud environment and compares some existing algorithms using specific measures for a better understanding.

The Eucalyptus cloud uses two methods for scheduling VMs: Fusion Method and Enhanced Round-Robin. The experiment showed that Enhanced Round-Robin uses less energy compared to other methods and that using Physical Machine load as a limit of resigning saves the most energy. The Fusion Method outperforms other techniques by reducing the number of Physical Machines turned on and increasing power efficiency. The authors also developed a self-adapting security monitoring system with goals of security, cost savings, tenant-driven customization, and self-adaptation. The Adaptation Manager is the main element that can adapt to changing conditions in the cloud infrastructure and serves as a coordinator of the adaptation process.

This paper presents a self-adapting security monitoring system for cloud infrastructure that takes into account the specific monitoring requirements of each tenant. The system uses Master Adaptation Drivers to convert the tenant requirements into configuration settings and the Adaptation Manager to coordinate the adaptation process. The framework ensures security, cost efficiency, and responsiveness to dynamic events in the cloud environment. The design of the framework is such that it does not introduce new security weaknesses or affect the performance of the infrastructure. The system provides a comprehensive solution that combines precise security monitoring with self-adaptation.

The current security monitoring platform needs improvement to support more types of monitoring devices such as network traffic analysis and inside-the-host activity monitoring. The platform is also limited to supporting firewall functionality and doesn't cover the consequences of multi-tenant setups. The future work includes incorporating log collectors and aggregators and addressing the needs of a super-tenant (the provider) in the security monitoring architecture. The equitable sharing of monitoring resources between the tenants and the provider should be established with an adjustable threshold mentioned in the SLA.

Acknowledgment

The authors would like to thank Malaysia University of Science and Technology, for the support and encouragement to carry out this study.

References

[1] P. C. Yang, J. H. Chiang, J. C. Liu, Y. L. Wen, and K. Y. Chuang, “An efficient cloud for wellness self-management devices and services,” Proc. - 4th Int. Conf. Genet. Evol. Comput. ICGEC 2010, pp. 767–770, 2010, doi: 10.1109/ICGEC.2010.194.

[2] S. Zhang, S. Zhang, X. Chen, and X. Huo, “The comparison between cloud computing and grid computing,” ICCASM 2010 - 2010 Int. Conf. Comput. Appl. Syst. Model. Proc., vol. 11, 2010, doi: 10.1109/ICCASM.2010.5623257.

[3] N. Sadashiv and S. M. D. Kumar, “Cluster, grid and cloud computing: A detailed comparison,” ICCSE 2011 - 6th Int. Conf. Comput. Sci. Educ. Final Progr. Proc., pp. 477–482, 2011, doi: 10.1109/ICCSE.2011.6028683.

[4] E. Raggi, K. Thomas, T. Parsons, A. Channelle, and S. van Vugt, “Social Networks and Cloud Computing,” Begin. Ubuntu Linux, pp. 337–348, 2010, doi: 10.1007/978-1-4302-3040-3_15.

[5] “The Treacherous 12 Top Threats Working Group,” 2016. [Online]. Available: https://cloudsecurityalliance.org/download/the-treacherous-twelve.

[6] Dave Shackleford, “9 cloud migration security considerations and challenges | TechTarget,” Voodoo Security, Nov. 23, 2021. https://www.techtarget.com/searchcloudcomputing/tip/9-cloud-migration-security-considerations-and-challenges (accessed Feb. 01, 2023).

[7] S. A. Bello et al., “Cloud computing in construction industry: Use cases, benefits and challenges,” Autom. Constr., vol. 122, p. 103441, Feb. 2021, doi: 10.1016/J.AUTCON.2020.103441.

[8] I. M. Khalil, A. Khreishah, and M. Azeem, “Cloud Computing Security: A Survey,” Comput. 2014, Vol. 3, Pages 1-35, vol. 3, no. 1, pp. 1–35, Feb. 2014, doi: 10.3390/COMPUTERS3010001.

[9] M. Carroll, A. Van Der Merwe, and P. Kotzé, “Secure cloud computing: Benefits, risks and controls,” 2011 Inf. Secur. South Africa - Proc. ISSA 2011 Conf., 2011, doi: 10.1109/ISSA.2011.6027519.

[10] A. Waqar, A. Raza, and H. Abbas, “User Privacy Issues in Eucalyptus: A Private Cloud Computing Environment,” in 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, Nov. 2011, pp. 927–932. doi: 10.1109/TrustCom.2011.128.

[11] M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Nawaz, and U. Farooq, “Resource Allocation Techniques in Cloud Computing: A Review and Future Directions,” Elektron. ir Elektrotechnika, vol. 26, no. 6, pp. 40–51, Dec. 2020, doi: 10.5755/J01.EIE.26.6.25865.

[12] L. Wang and D. Zhang, “Research on OpenStack of open source cloud computing in colleges and universities’ computer room,” IOP Conf. Ser. Earth Environ. Sci., vol. 69, no. 1, p. 012140, Jun. 2017, doi: 10.1088/1755-1315/69/1/012140.

[13] M. E. Suliman, “A Brief Analysis of Cloud Computing Infrastructure as a Service (IaaS),” 2021. [Online]. Available: www.ijisrt.com.

[14] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” Proc. IEEE, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[15] McKeownNick et al., “OpenFlow,” ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008, doi: 10.1145/1355734.1355746.

[16] H. Wang, A. Srivastava, L. Xu, S. Hong, and G. Gu, “Bring your own controller: Enabling tenant-defined SDN apps in IaaS clouds,” Proc. - IEEE INFOCOM, Oct. 2017, doi: 10.1109/INFOCOM.2017.8057137.

[17] M. Johns, “Code-injection vulnerabilities in web applications - Exemplified at cross-site scripting,” IT - Inf. Technol., vol. 53, no. 5, pp. 256–260, Sep. 2011, doi: 10.1524/ITIT.2011.0651/MACHINEREADABLECITATION/RIS.

[18] T. Grandison and E. Terzi, “Intrusion Detection Technology,” Encycl. Database Syst., pp. 1568–1570, 2009, doi: 10.1007/978-0-387-39940-9_209.

[19] S. Lata and D. Singh, “Intrusion detection system in cloud environment: Literature survey &amp; future research directions,” Int. J. Inf. Manag. Data Insights, vol. 2, no. 2, p. 100134, Nov. 2022, doi: 10.1016/j.jjimei.2022.100134.

[20] D. Nurmi et al., “The Eucalyptus Open-source Cloud-computing System.”

[21] H. Alshaer, “An overview of network virtualization and cloud network as a service,” Int. J. Netw. Manag., vol. 25, no. 1, pp. 1–30, Jan. 2015, doi: 10.1002/NEM.1882.

[22] M. Yassin, H. Ould-Slimane, C. Talhi, and H. Boucheneb, “Multi-Tenant Intrusion Detection Framework as a Service for SaaS,” IEEE Trans. Serv. Comput., vol. 15, no. 05, pp. 2925–2938, Sep. 2022, doi: 10.1109/TSC.2021.3077852.

[23] A. Giannakou et al., “Self-adaptable Security Monitoring for IaaS Cloud Environments,” Jul. 2017, Accessed: Feb. 05, 2023. [Online]. Available: https://hal.inria.fr/tel-01653831.

[24] I. C. Lin, C. C. Chang, and C. H. Peng, “An Anomaly-Based IDS Framework Using Centroid-Based Classification,” Symmetry 2022, Vol. 14, Page 105, vol. 14, no. 1, p. 105, Jan. 2022, doi: 10.3390/SYM14010105.

[25] G. Chen et al., “Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services.”

[26] C. C. Lin, P. Liu, and J. J. Wu, “Energy-efficient virtual machine provision algorithms for cloud systems,” Proc. - 2011 4th IEEE Int. Conf. Util. Cloud Comput. UCC 2011, pp. 81–88, 2011, doi: 10.1109/UCC.2011.21.

[27] S. Knox, P. Meier, J. Yoon, and J. J. Harou, “A python framework for multi-agent simulation of networked resource systems,” Environ. Model. Softw., vol. 103, pp. 16–28, May 2018, doi: 10.1016/J.ENVSOFT.2018.01.019.

[28] T. Madi et al., “ISOTOP,” ACM Trans. Priv. Secur., vol. 22, no. 1, Oct. 2018, doi: 10.1145/3267339.

[29] A. Rath, B. Spasic, N. Boucart, and P. Thiran, “Security Pattern for Cloud SaaS: From System and Data Security to Privacy Case Study in AWS and Azure,” Comput. 2019, Vol. 8, Page 34, vol. 8, no. 2, p. 34, May 2019, doi: 10.3390/COMPUTERS8020034.

Optimal Training Ensemble of Classifiers for Classification of Rice Leaf Disease

Sridevi Sakhamuri[1]*, K Kiran Kumar[2]

Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur Dist, AP, India[1]
Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur Dist, AP, India[2]

***Abstract*—Rice is one of the most extensively cultivated crops in India. Leaf diseases can have a significant impact on the productivity and quality of a rice crop. Since it has a direct impact on the economy and food security, the detection of rice leaf diseases is the most important factor. The most prevalent diseases affecting rice leaves are leaf blast, brown spots, and hispa. To address this issue, this research builds a new classification model for rice leaf diseases. The model begins with a preprocessing step that employs the Median Filter (MF) process. Improved BIRCH is then utilized for picture segmentation. Features such as LBP, GLCM, color, shape, and modified Median Binary Pattern (MBP) are retrieved from segmented images. Then, an ensemble of three classification models, including Bi-GRU, Convolutional Neural Network (CNN), and Deep Maxout (DMN) is utilized. By adjusting the model weights, the suggested Opposition Learning Integrated Hybrid Feedback Artificial and Butterfly algorithm (OLIHFA-BA) will train the model to improve the performance of the proposed work.**

Keywords—Rice leaf; modified MBP; Bi-GRU; improved BIRCH; OLIHFA-BA Algorithm

I. Introduction

Agriculture is more important to the Indian economy than other industries. Rice production is a significant aspect of agriculture [1] [2] [3] [4]. 20% of India's GDP is contributed by rice agriculture [5] [6] [7] [8]. Rice is cultivated in most Indian states (Odessa, Uttar Pradesh, Punjab, West Bengal, Assam, Tamil Nadu, Bihar, etc.) [9] [10] [11] [12]. Due to the several diseases that can impact rice plants, rice productivity is currently falling. Farmers have only a limited comprehension of the disease. This crisis diminishes the efficiency of rice production, hence reducing the agricultural income [13] [14] [15].

Not only are rice leaf diseases prevalent in India, but they are also prevalent in other countries. There are several types of leaf diseases, such as brown spot, tungro, bacterial blight, blast, etc. Farmers have no control over these infections. Consequently, visual examination or laboratory tests are used to detect illnesses on leaves [16] [17] [18]. Visual analysis of this issue is time-consuming for a specialist. In addition, when chemical reagents are necessary, the experimental procedure becomes more difficult.

Certain strategies are employed to simplify these matters. The Deep Learning (DL) algorithm is applicable to agricultural problems such as root segment, fruit count, seed selection, disease classification, etc. [19] [20] [21]. DL algorithms are sophisticated versions of ML for detecting crop infections. With this method, the inputs were automatically learned, and the output was generated based on the decision criteria. CNN technology was utilized during the development of the visual image. In addition, "Rice Doctor" and "Rice Xpert" smart phone applications for farmers were introduced using the internet and mobile technology. The "Rice Doctor" app serves as a questionnaire for farmers [22] [23] [24] [25].

This study considered the most prevalent rice leaf diseases, including brown spot, leaf blast, and bacterial blight. The CNN model was calibrated to improve its accuracy. It is exceptionally accurate. Only the disorders were treated with the tuned model. We have to implement advanced DL algorithms [26] [27] [28] [29] due to the need to identify different forms of rice leaf disease and raise the degree of accuracy.

The contributions are detailed as follows:

- Proposed a new classification model for rice diseases with enhanced BIRCH-based segmentation.
- Utilizes an ensemble model based on OLIHFA-BA with a defined feature set consisting of enhanced MBP features, Local Binary Pattern (LBP), Gray Scale Co-Occurrence Matrix (GLCM), color, and shape features.

The structure of the paper is as follows: Section II describes standard works. Section III describes the adopted phases of the suggested classification strategy, whereas Section IV describes characteristics. Section V displays optimised ensemble classifiers, while Section VI depicts an assisted OLIHFA-BA optimization algorithm. Results and conclusions are provided in Sections VII and VIII.

II. Literature Review

A. Related Works

In 2021, Krishnamoorthy N et al. [1] conducted study on rice leaf disease classification. Moreover, fifty percent of the world's population consumes rice. Therefore, rice is the world's principal source of energy. Rice plant diseases, which are caused by viruses, bacteria, fertile soil, pests, temperature fluctuations, and so on, are the most significant obstacles in rice cultivation. Finding and treating rice plant illnesses is a challenging undertaking for farmers. In this investigation, sickness identification was performed using a DL approach. CNNs were used for object segmentation, image classification, and image analysis in Deep Learning. The

*Corresponding Author.

recommended model achieved the highest accuracy, 95.67 percent.

Kumar Sethy et al. [2] looked at how to find diseases on rice leaves in 2020. This study talks about the four diseases that can hurt rice leaves: bacteria blight, blast, tungro, and brown spot. DL technology is used to find out if someone is sick. In this study, TL with CNN models, deep features, and SVM were used to figure out how to group things. Two conclusions were similar, but the deeper features of SVM performed much better.

Chen et al. [3] conducted rice plant disease detection research in 2021. Using CNN models constructed with the DL approach presented various technical challenges, such as picture identification and classification. In this work, however, MobileNet-V2 was employed, and numerous techniques were employed to assess the significance of spatial sites for input characteristics and inter-channel interaction. Using DL-based CNN approaches, most of the technical challenges associated with picture recognition and classification have been resolved. Transfer learning and the enhanced loss function were repeated on two separate occasions. The public dataset [48] utilized for this investigation was 99.67% accurate. In difficult research environments, the accuracy of rice plant disease diagnosis was determined to be 98.48%. Therefore, the suggested method was more efficient at identifying rice plant diseases.

Madhavi and M.A. Saleem [4] studied the identification of rice leaf diseases in 2021. To preserve the growth of the agricultural sector, the first step of plant disease identification was taken. The comparison between automatic and manual plant monitoring is required and beneficial. CNN is frequently used for this kind of categorization. It does a good job of classifying and diagnosing plant diseases by utilizing highly accurate data collected from a range of sources.

Jjang et al. [5] studied the leaf diseases of wheat and rice plants in 2021. To reduce plant growth loss, this issue would be diagnosed quickly and precisely. In this study, the Image NET pre-training model, alternating learning, and the VGG16 implementation were used to facilitate multitask learning, TL, and recognition. This model's accuracy for rice plants was 98% and for wheat plants it was 99%. This model has proved the excellent performance of the VGG16 model and the multi-task TL, which is accurate in recognizing plant diseases.

Jiang et al. [6] conducted a study on image-based disease diagnosis in rice leaf images using Support Vector Machine (SVM) and DL in 2020. Combining these two approaches has helped to effectively address the issue while also improving precision. The authors of this study have utilized CNN to derive the images of the relevant leaf. During the last round of the evaluation, the classic Back Propagation Neural Network (BPNN) models were compared to the more accurate SVM models.

Zhang et al. [7] conducted study in 2020 utilizing spectral image technology to identify leaf illnesses in rice crops. This method was utilized to determine the severity of rice leaf explosions. In this work, a hyperspectral imaging method was used to distinguish between images of afflicted and healthy leaves. The data was then reconstructed using the SRR method. This model's precision was approximately 98%.

In 2021, Bakade et al. [8] conducted a study on preventing bacterial illness in rice leaves. Xoo is the primary cause of this problem. This investigation revealed the interconnected actions of genes and plant immune pathways, which could be leveraged to develop resistant rice cultivars.

B. Review

Once, the only means for diagnosing a disease was a manual analysis of the leaf. This was accomplished manually by examining plant leaves or consulting a book to identify the disease [5]. This method has three major drawbacks: it is imprecise, it cannot study every leaf, and it is time-consuming. Several approaches for effectively identifying these ailments have been created because of the advancement of science and technology. Image processing and deep learning are methodologies. Image processing includes a range of techniques, including filtering, clustering, histogram analysis, and image processing algorithms, to discover damaged areas and diagnose diseases. In contrast, DLNN are used to identify diseases. There are two principal causes of plant diseases. The first is a bacterial or fungal attack, and the second is an unanticipated shift in the weather [6].

When addressing rice infections, we must consider a few critical elements. Collecting samples from a damaged rice plant is one of the crucial and critical duties. To do this, multimedia sensors may be deployed across the farm. This permits routine monitoring of rice plants. Additionally, the effects of climate change on rice plants can be monitored and studied. This approach has several disadvantages, including the necessity for frequent system maintenance and low precision due to shadows in the gathered photos. It is crucial to accurately identify rice infections in order to avert the disease's devastating effects on crop productivity. However, the present methods for diagnosing illnesses in rice are neither exact nor effective, necessitating the need of supplementary equipment.

III. Adopted Phases in Proposed Classification Approach

The following are the adopted phases of the proposed rice leaf disease classification:

- In the very first phase, the input image is submitted to MF for the aim of pre-processing. Then, Improved BIRCH is implemented to segment the images.
- The LBP, GLCM, colour, and improved MBP-based feature set is derived from segmented images. Then, ensemble model-based classification is performed with three classifiers, including Bi-GRU, CNN, and DMN as shown in Fig. 1.
- The final classification results are determined by the combined averaged outcome.

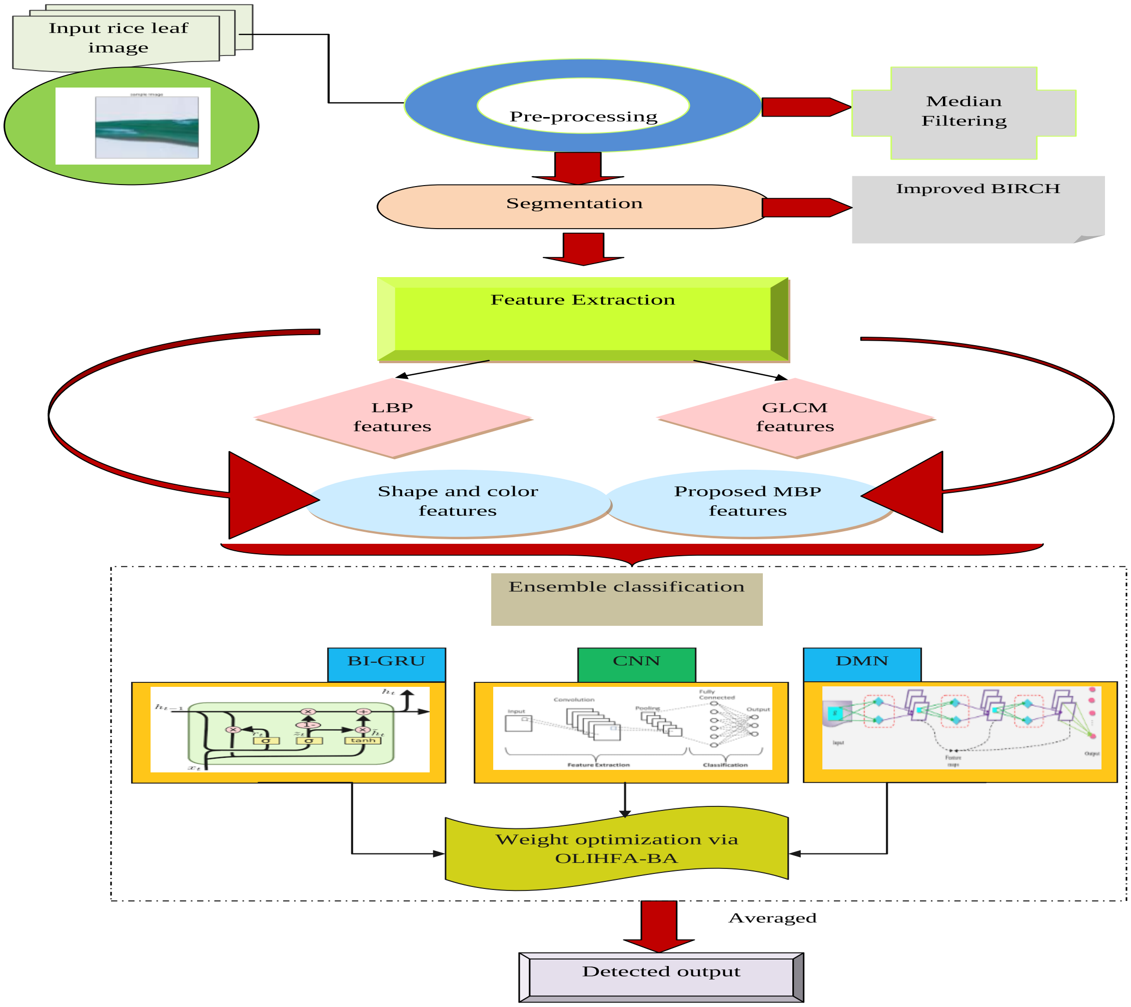


Fig. 1. A diagram of the rice disease classification model that was used.

A. *Pre-processing*

In this research, median filtering is utilised to pre-process the image input.

MF [30]: The median filter is a common non-linear digital filtering technique used to remove noise from a picture or signal. This type of noise reduction is a typical pre-processing technique used to improve the results of future processing (edge recognition in image). MF is widely utilised in the processing of digital photos, and in certain instances it preserves edges while lowering noise.

The MF oriented pre-processed image is designated as $\left(ig^{mf}\right)$.

Following image processing, improved BIRCH is utilized for image segmentation.

B. *Modified BIRCH Model*

Clustered features [31] store the information necessary for data grouping and provide a concise description of a set of points in feature space. Consider the set of points in the dimension [31].

Conservatively, the size, of a set is described as the average distance between two points, as shown in Eq. (1).

$$Di=\sqrt{\frac{\sum_{i=1}^{M}\left\|s_i-t_i\right\|_2^2}{M(M-1)}} \tag{1}$$

As per improved BIRCH, Di is modeled as shown in Eq. (2).

$$Di(s_i,t_i)=\sqrt{\frac{\sum_{i=1}^{M}(s_i-t_i)^2}{\sum_{i=1}^{M}(s_i)^2+\sum_{i=1}^{M}(t_j)^2-\sum_{i=1}^{M}s_it_j}} \tag{2}$$

The representation of clustering feature, cf, of set X is specified in Eq. (3).

$$cf=(M,L,x) \tag{3}$$

Here, M points out points in X, $L=\sum_{i=1}^{M}s_i$ refers to summation of every points in X and the scalar $x=\sum_{i=1}^{M}\left\|s_i\right\|_2^2$ refers to summation of square of every element of every points in X. Thus, an individual point is indicated by a cluster feature. A cluster feature represented the set X, and retained data to calculate the centroid, s_c, of X, the size Di of set, and the distance, $d(X_1, X_2)$, among 2 sets. Conservatively, these can be expressed as in Eq. (4).

$$s_c = \frac{L}{M} \tag{4}$$

As per enhanced BIRCH, s_c is computed as exposed in Eq. (5), in which, $corr$ points out correlation that is modeled as in Eq. (6).

$$s_c = \frac{L}{M} * corr \tag{5}$$

$$corr = 1 - \left(\frac{\sum_{i=1}^{M} (s_i - \bar{s})(t_i - \bar{t})}{\sqrt{\sum_{i=1}^{M} (s_i - \bar{s})^2} \sqrt{\sum_{i=1}^{M} (t_i - \bar{t})^2}} \right)^2 \tag{6}$$

The improved BIRCH image is denoted as ig^{IBIRCH}.

IV. Extracting LBP, GLCM, Color, Shape and Improved MBP Features

From ig^{IBIRCH}, the feature set including LBP, GLCM, color, shape and improved MBP are extracted.

A. Shape Features

The primary source of information used for object identification is shape [32]. Without shape, a visual item cannot be effectively recognised. Without understanding shape, an image is incomplete. Although the two items cannot have the exact same shape, we may identify comparable shapes by utilising a variety of techniques. Triangle, Circle, Rectangle, Square, Oval, and Diamond are some of the available shapes. The features of the extracted shapes are indicated by fs^{Sh}.

B. Colour Features

Colour space characterizes colour in the type of intensity value [32]. By employing the colour space approach, we can define, see, and produce colour. The colour histogram shows the image from various angles. The colour histogram used to describe the frequency distribution of colours in the image counts and stores related pixels. Every statistical colour frequency in an image is examined using the colour histogram. The colour histogram not only focuses on specific areas of an image, but also solves difficulties with translation, rotation, and angle of view changes. The local colour histogram is simple to calculate and robust to minute image fluctuations, making it crucial for the retrieval and indexing of image databases.

C. GLCM Features

GLCM is employed to evaluate the spatial association among the pixel [33]. The constraints in GLCM are given in Table I.

TABLE I. GLCM Features

S. No.	Features	Arithmetical term
1.	Energy	$E = \sum_{\omega}\sum_{\xi} \upsilon_{\omega\xi}^2$ here $\upsilon_{\omega\xi}$ is the $(\omega,\xi)^{th}$ entry in GLCM
2.	Entropy	$Et = -\sum_{\omega}\sum_{\xi} \upsilon_{\omega\xi} \log_2 \upsilon_{\omega\xi}$
3.	Variance	$Var = \sum_{\omega}\sum_{\xi} (\omega - \mu)^2 \upsilon_{\omega\xi}$, where μ specifies the mean of $\upsilon_{\omega\xi}$
4.	Contrast	$Con = \sum_{\omega}\sum_{\xi} (\omega - \xi)^2 \upsilon_{\omega\xi}$
5.	Correlation	$Cor = \frac{\sum_{\omega}\sum_{\xi} (\omega\xi)\upsilon_{\omega\xi} - \mu_x\mu_y}{\sigma_x\sigma_y}$, where σ_x, σ_y, μ_x, μ_y are the std deviations and mean of υ_x, υ_y
6.	Sum Average	$SA = \sum_{\omega=2}^{2N_\upsilon} \omega.\upsilon_{x+y}(\omega)$, where N_υ indicates the varied gray levels in image.
7.	Homogeneity	$Hom = \sum_{\omega}\sum_{\omega} \frac{1}{1 + (\omega - \xi)^2} \upsilon_{\omega\xi}$
8.	Sum Variance	$SV = \sum_{\omega=2}^{2N_\upsilon} (\omega - SE)^2 s_{x+y}(\omega)$
9.	Sum Entropy	$SE = \sum_{\omega=2}^{2N_\upsilon} \upsilon_{x+y}(\omega) \log\{\upsilon_{x+y}(\omega)\}$
10.	Difference Variance	$DV =$ variance of $\upsilon_{\omega x-y}$
11.	MCC	$MCC = \sum_{k} \frac{g(\omega,k)g(b,k)}{g_x(\omega)g_y(k)}$
12.	Difference Entropy	$DE = \sum_{\omega=0}^{N_\upsilon - 1} \upsilon_{x-y}(\omega) \log\{\upsilon_{x-y}(\omega)\}$
13.	IMC 1	$IMC1 = \frac{hxy - hxy1}{\max\{hx, hy\}}$
14.	IMC 2	$IMC2 = \sqrt{(1 - \exp[-2.0[hxy1 - hxy]])}$ $hxy = -\sum_{\omega}\sum_{\xi} \upsilon_{\omega\xi} \log_2 \upsilon_{\omega\xi}$, where $hxy1 = -\sum_{\omega}\sum_{\xi} \upsilon_{\omega\xi} \log_2 \{\upsilon_x(\omega)\upsilon_y(\xi)\}$,,

D. Modified MBP Features

The MBP [34] attempts to determine the LBP by thresholding pixels with a median value above the threshold. In this level of filtering, the centre pixel is evaluated, providing 29 potential structures. MBP is conventionally represented using Eq. (7).

$$MBP = \sum_{i=1}^{J} f(b_i) \times 2^i \tag{7}$$

As per modified MBP, it is modelled as in Eq. (8). Here, we point out weight function that is evaluated by means of cubic chaotic map.

$$MBP = \sum_{i=1}^{J} f(b_i) \times 2^i * we \tag{8}$$

Conventionally, $f(b_i)$ is modeled based upon median as in Eq. (9).

$$f(b_i) = \begin{cases} 1; & if\ b_i \geq med \\ 0; & otherwise \end{cases} \tag{9}$$

As per modified MBP, $f(b_i)$ is modeled based upon median absolute deviation (MAD) as in Eq. (10) and Eq. (11).

$$f(b_i) = \begin{cases} 1; & if\ b_i \geq MAD \\ 0; & otherwise \end{cases} \tag{10}$$

$$MAD = Median\left|b_i - \bar{b}\right| \tag{11}$$

E. LBP Features

In a variety of comparison analyses, the patterns of LBP [35] are provided with a high level of discrimination and simplicity. The fundamental LBP is used to derive the differential features between a certain reference pixel and its neighbours with radius. The resultant LBP for a pixel is given by Eq. (12), where is the geometric mean of nearby pixels and represents the grey values of the middle pixel and surrounding pixels with radius.

$$LBP_{P,R}(q_e, r_e) = \sum_{P=0}^{P-1} s * 2^P \tag{12}$$

The derived LBP, GLCM, color, shape and modified MBP are totally implied by fs, i.e., $fs^{Sh} + fs^{Cl} + fs^{glcm} + fs^{IMBP} + fs^{LBP} = fs$.

V. Optimized Bi-GRU, CNN and DMN Models

The derived fs is then given as input to three classifiers such as Bi-GRU, CNN and DMN.

A. Bi-GRU

It [36] is a sort of Recurrent Neural Network (RNN) that facilitates the handling of data from successive and previous time steps in order to provide output predictions based on the present state. Eq. (13) to (16) expose the BI-GRU calculation by displaying the sigmoid function, hidden, and input vectors as, and respectively. The reset data is represented as, weight, time interval, and the condition of the cell at the previous time stamp.

$$F_t = \left(WG_i * \left[v_t, hi_{t-1},\right]\right)\sigma \tag{13}$$

$$r_t = \left(WG_i * \left[v_t, hi_{t-1}\right]\right)\sigma \tag{14}$$

$$hi_t = \left(WG_c * \left[r_t . v_t, hi_{t-1}\right]\right)\tanh \tag{15}$$

$$hi_t = .C_{t-1}\left(1 - F_t\right) + hi_t F_t \tag{16}$$

B. CNN

The features of CNN [37] are depicted in Eq. (17).

$$D_{r,t,w}^{l} = \varpi_w^{l\ T} PA_{r,t}^{l} + B_w^{l} \tag{17}$$

In Eq. (17), ϖ_w^l → weight tuned optimally by means of OLIHFA-BA model, B_w^l →bias. At core location (r,t) of l^{th} layer, the input patch is signified as $PA_{r,t}^l$, $D_{r,t,w}^l$ → convolution features and $\left(act_{r,t,w}^l\right)$ → activation value.

$$act_{r,t,w}^l = act\left(D_{r,t,w}^l\right) \tag{18}$$

Pooling layer: CNN loss PL is given in Eq. (20), here, θ → term linked with W_w^l and B_w^l . The output, the labels and h^{th} input feature is termed as $F^{(h)}$, $H^{(h)}$ and $PA^{(h)}$ and $nn_{r,t}$ signifies (r,t) near neighbor position.

$$H_{r,t,w}^l = pool\left(act_{m,h,w}^l\right), \forall (m.h) \in nn_{r,t} \tag{19}$$

$$PL = \frac{1}{wn} \sum_{h=1}^{wn} l\left(\theta; H^{(h)}, F^{(h)}\right) \tag{20}$$

C. DMN Classifier

DMN [38], a form of NN, is utilized well in a few applications. A NN contains candidate components in each neuron. It is intended to use an extendable maximum value component for neuron activation [39]. Mark as the node of the hiding layer and each of its components. Eq. (21) and Eq. (22) demonstrate their relationship to one another:

$$J_m^i = \max_{j \in 1,2,\ldots,u} O_m^{ij} \tag{21}$$

By forwarded propagation, O_m^{ij} is modeled as in Eq. (22).

$$O_m = W^{*G}_{m-1} J_{m-1} + f_m \tag{22}$$

Here, $O_1 \in L^O$ refers to l^{th} layer vector

$J_{m-1} \in L^{H}$ and $W^{*}{}_{m-1} \in L^{H\times O}$ refers to max out activation vector and weight matrix of $m-1$ layer.

$f_m \in L^{O}$ refers to bias vector of m^{th} layer.

The Bi-GRU, CNN and DMN outputs are averaged to get final result.

VI. OLIHFA-BA Assisted Algorithm for Optimization

Objective: The objective is to diminish error as in Eq. (23).

$$Objective = \min(Error) \tag{23}$$

Solution Encoding: The weights of BI-GRU (WG), CNN (ϖ) and DMN (k) are elected optimally by OLIHFA-BA scheme as given in Fig. 2.

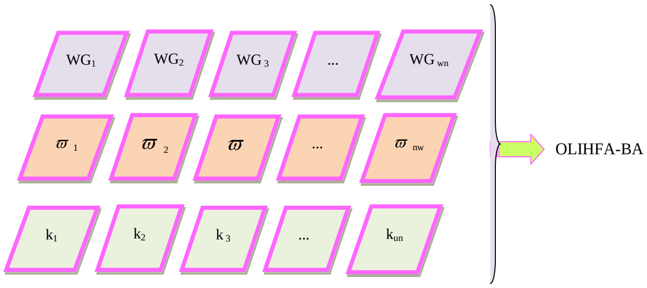


Fig. 2. Solution encoding.

Existing Butterfly Optimization Algorithm (BOA) [39] recognizes the best options, however it is not precise. FAT [40] is merged with BOA [39] to develop OLIHFA-BA, which addresses the shortcomings of original BOA [39] [41] [42] [43] [44]. The concepts of hybridization show considerable potential for advancing with improved outcomes [41] [42] [43] [44].

The butterfly mating and feeding behaviors inspired BOA. The distinctive characteristic of BOA represented by Eq. (24) is the fragrances with diverse aromas in butterflies, where the power exponent (between 0 and 1) relates to the index of sensor modality and indicates stimulus magnitude.

$$q = sl^{\theta} \tag{24}$$

Butterflies are capable of accurately locating scent sources and share this knowledge with one another. There were three stages of OLIHFA-BA, and they were as follows:

1) Initialization: The constraints and objectives would all be initialized. Additionally, chaotic-based OBL is produced according to the model in Eq. (25), which α refers to chaotic map function that uses a tent map.

$$\bar{y}_i = a + b - y_i * \alpha \tag{25}$$

2) Iteration: Both local and international searches were performed.

The traditional numerical depiction for butterfly global search movement is in Eq. (26), herein, t points out iteration, y_i^t points out i^{th} butterfly position at t, q_i points out scent of i^{th} butterfly and G_{best} points out finest position of global butterfly.

$$y_i^{t+1} = y_i^t + \left(ra^2 \times G_{best} - y_i^t\right) \times q_i \tag{26}$$

In OLIHFA-BA model, it is formulated depending upon FAT's branch update as in Eq. (27), in which, d points out smaller constant, (d = 0.382) and $rand(0,1)$ points out random numeral. Moreover, we use FAT based crossover operation to generate new branches.

$$y_i^{t+1} = y_i^t + \begin{pmatrix} rand(0,1) * y_{best} \\ -rand(0,1) * y_i \end{pmatrix} * d + levy(\beta) \tag{27}$$

Numerous typical events might hinder butterfly movement and fragrance dispersal. Local search is simulated by the butterfly positions according to Eq. (28).

$$y_i^{t+1} = y_i^t + \left(ra^{2} \times y_j^t - y_k^t\right) \times q_i \tag{28}$$

3) Termination: OLIHFA-BA terminates once the utmost iterations were arrived. The OLIHFA-BA model is shown in Algorithm 1.

Algorithm 1: OLIHFA-BA model

```
Initializing populace
Intensity evaluation
While stop criterion wasn't met
    for every butterfly $BF$ do
        Evaluating scent
    end for
    Find finest $BF$
    for each butterfly $BF$
    Assume $ra$
        if $ra < pr$
            Update positions based upon FAT update as in Eq. (27)
        else
            Update positions based upon Eq. (28)
        end if
    end for
    update $\theta$
end while
```

VII. Results and Discussion

A. Simulation Set Up

Matlab was used to create the proposed Ensemble Classifiers (EC) (Bi-GRU, CNN, and DMN) + OLIHFA-BA focused classification for rice illness. On numerous measures, EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA was compared against Improved Crossover Monarch Butterfly Optimization (ICRMBOS) [45], Transfer Learning based Deep CNN (TL-DCNN) [46], LMBWO [47], Deep Belief Network (DBN), Recurrent Neural Network (RNN), Quantum Neural Networks (QNN), SVM, RF, and Long Short Term Memory (LSTM), EC + DHO, EC + SMO (Spider Monkey Optimization), EC + CMBO (Cat and Mouse Optimization), EC + BMO (Blue Monkey Optimization), and EC + MFO (Mouth Flame Optimization). Fig. 3 displays the illustration of the sample image.

B. Performance Analysis

The performances of the proposed EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA over existing met heuristic models are calculated and displayed in Fig. 4 to 6 for various measures. The EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA model is compared to the EC + DHO, EC + SMO, EC + CMBO, EC + BMO, and EC + MFO models for several LPs. Table II compares EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA with existing classifiers such as DBN, RNN, QNN, SVM, RF, LSTM, ICRMBOS [45], TL-DCNN [46], and LMBWO [47]. The proposed EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA model has produced superior results compared to differentiated approaches for optimization and classification models. The improved prediction rate should result in slight negative results and bigger positive values. As seen in Fig. 4, the outputs of the EC model for all positive metrics grow, whilst the outputs for negative metrics decrease. Specifically for current and prospective schemes, the 50th LP yielded the best results. The proposed system at the 50th LP achieved the highest accuracy values (0.96), whereas other schemes achieved low precision levels. This improvement in classifier analysis and optimization analysis by EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA is mostly due to the incorporation of enhancements in features, segmentation, and classifiers.

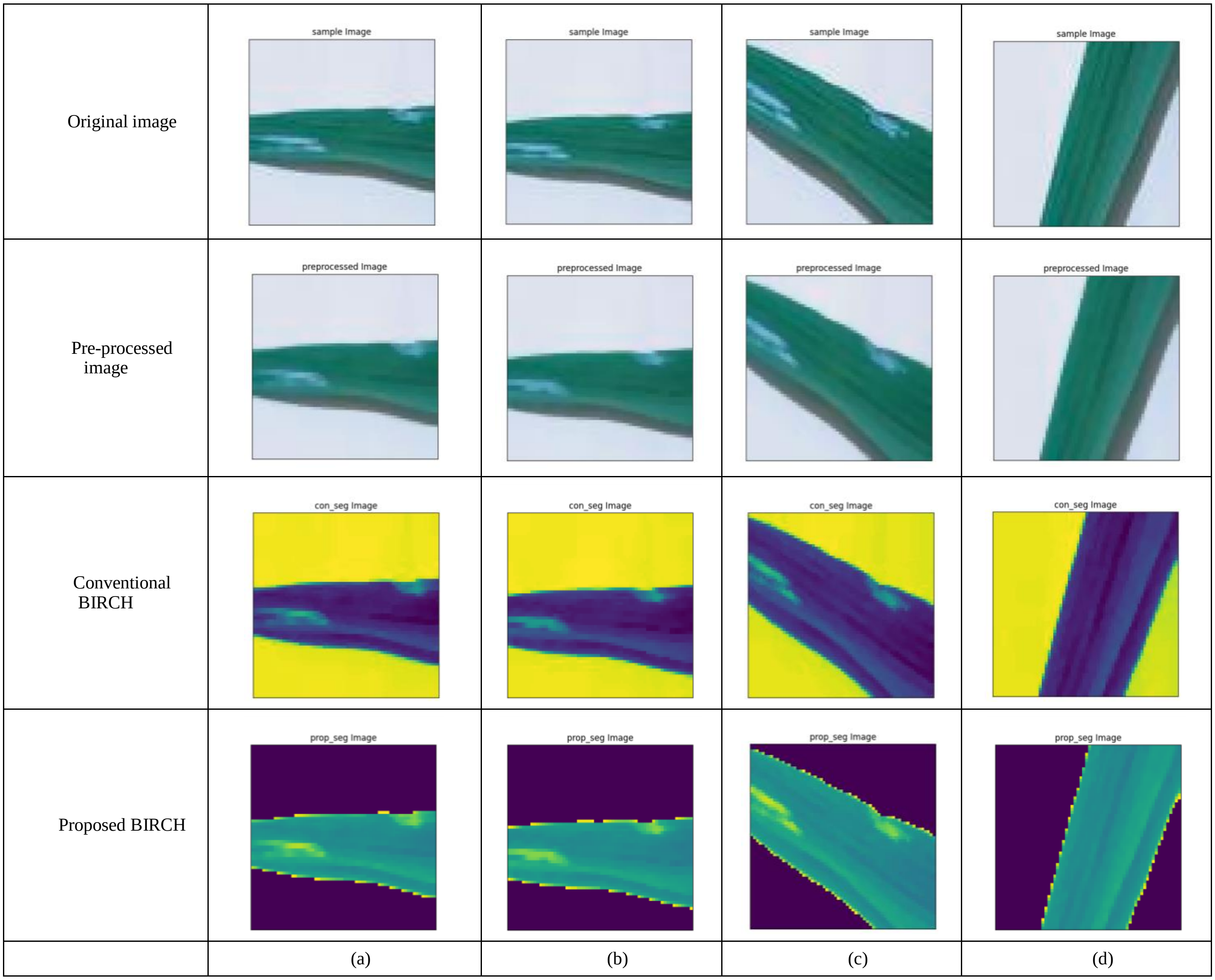
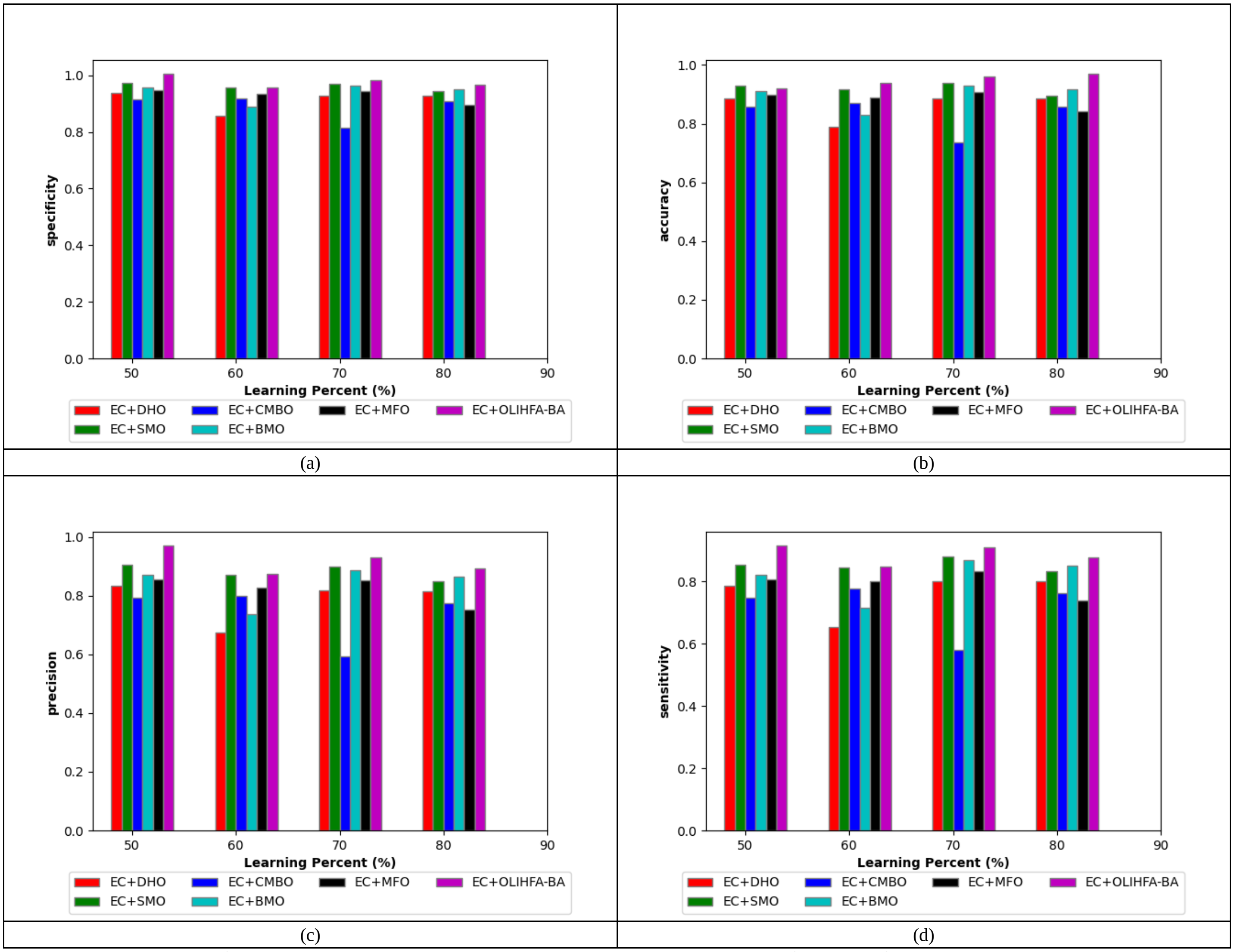


Fig. 3. Illustration sample of images for (a) 1 (b) 2 (c) 3 (d) 4.



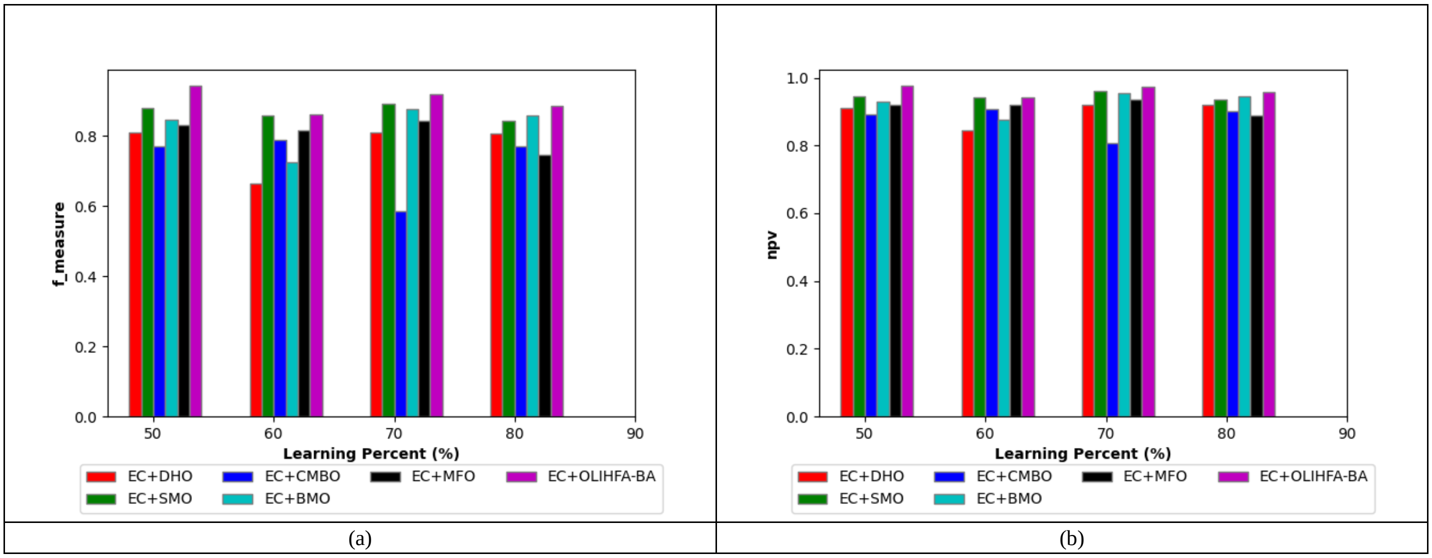
(a)

(b)

(c)

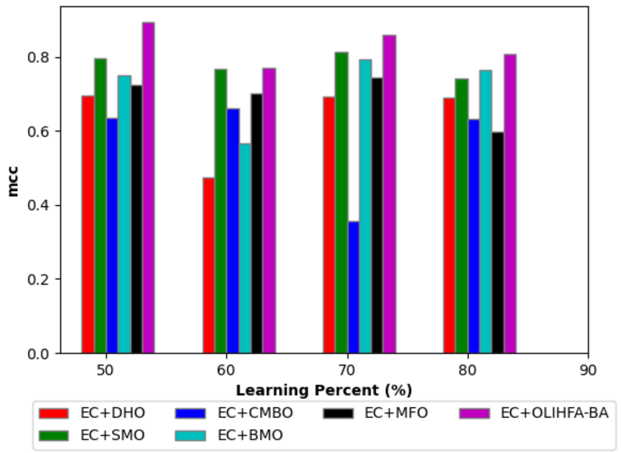
(d)

Fig. 4. Comparison of EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA to optimization techniques for (a) Specificity (b) Accuracy (c) Precision and (d) Sensitivity.



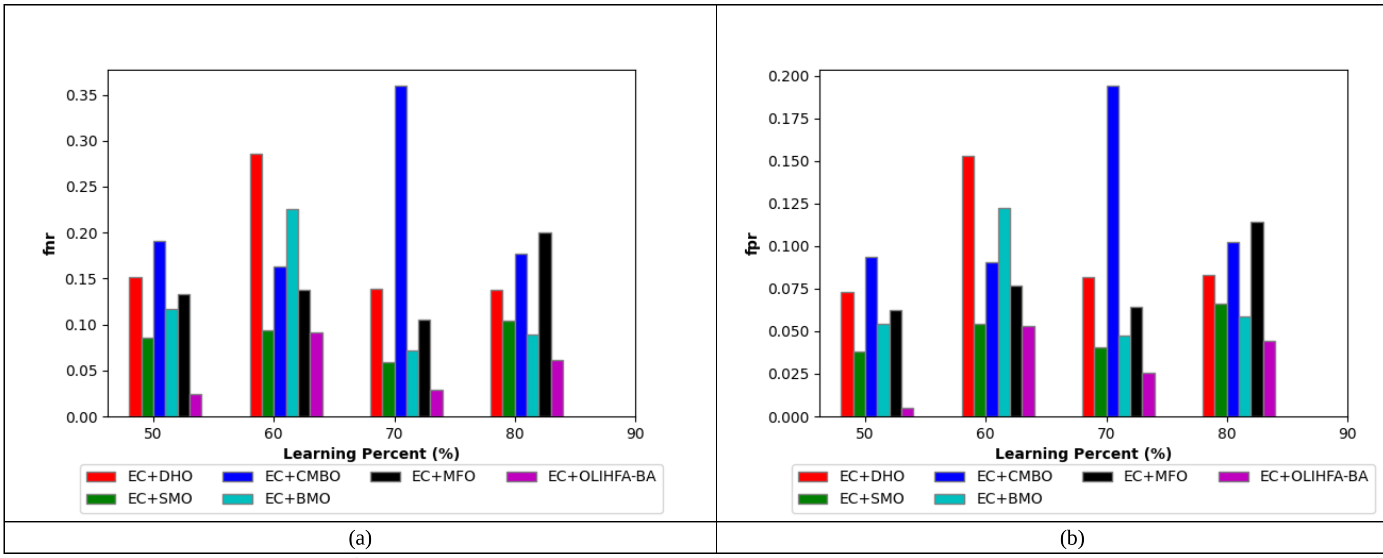
(a)

(b)



(c)

Fig. 5. Comparison of EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA to optimization techniques for (a) F-measure, (b) NPV, and (c) MCC.



(a) (b)

Fig. 6. Analysis of EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA over optimization approaches on (a) FNR and (b) FPR

TABLE II. ANALYSIS ON PROPOSED AS WELL AS EXISTING VARIANTS

Metrics	DBN	RNN	QNN	SVM	RF	LSTM	ICRMBOS [45]	TL-DCNN [46]	LMBWO [47]	EC (Bi-GRU, CNN and DMN) + OLIHFA-BA
NPV	0.810489	0.863751	0.886403	0.848648	0.820849	0.864672	0.801257	0.854521	0.896886	0.958836
Sensitivity	0.869236	0.836111	0.811791	0.808681	0.881809	0.889946	0.79587	0.795495	0.795339	0.878906
FNR	0.260323	0.244131	0.291496	0.268448	0.289677	0.273445	0.23056	0.286253	0.204661	0.061094
Precision	0.823257	0.81249	0.896509	0.800818	0.872516	0.821655	0.80614	0.863017	0.807613	0.891852
FPR	0.465444	0.45827	0.410539	0.430812	0.421072	0.478837	0.458594	0.40966	0.096193	0.044074
MCC	0.800157	0.793013	0.866155	0.895368	0.872298	0.79213	0.866973	0.893905	0.701818	0.807754
Accuracy	0.835872	0.84927	0.805443	0.794296	0.865515	0.839268	0.844073	0.83556	0.856622	0.969837
F measure	0.806143	0.886593	0.824477	0.836322	0.898775	0.804303	0.854824	0.846638	0.801429	0.885329
Specificity	0.790013	0.865374	0.878082	0.886596	0.792012	0.822298	0.853153	0.848949	0.903807	0.965926

C. Convergence Study

Fig. 7 shows the completed cost analysis. EC (CNN, Bi-GRU, and DMN) is evaluated over EC + DHO, EC + SMO, EC + CMBO, EC + BMO, and EC + MFO. In view of Fig. 7, the cost of EC (Bi-GRU, CNN, and DMN) has increased little. From iterations 0 through 5, the CMBO model achieves a high cost of 1.095. In addition, between iterations 0 and 5, the cost of EC (Bi-GRU, CNN, and DMN) is approximately 1.084%. Thus, EC (Bi-GRU, CNN, and DMN) was able to obtain BIRCH at a reduced cost and with expanded features.

D. Conventional VS. Proposed Analysis

Table III compares the adopted EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA scheme to EC without OLIHFA-BA, EC with traditional BIRCH, and EC with traditional MBP. Observing the Table, the recommended EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA had superior values than the EC without OLIHFA-BA, the EC with conventional BIRCH, and the EC with conventional MBP. This demonstrates the effect of BIRCH enhancements and hybrid optimization.

E. Analysis on DICE and Jaccard Scores

Table IV provides a study of dice scores and Jaccard scores. The dice and Jaccard scores for suggested BIRCH are greater than those for FCM, K-mean, and conventional BIRCH.

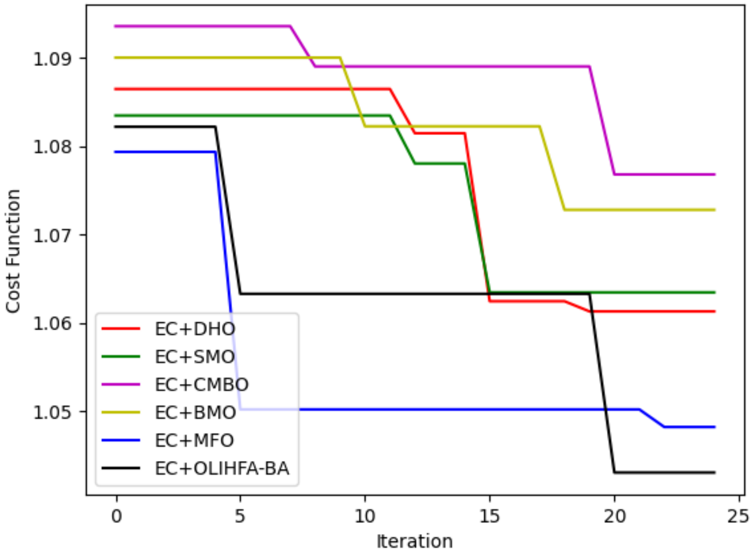


Fig. 7. Convergence study.

TABLE III. ANALYSIS ON PROPOSED AND EXISTING VARIANTS

Metrics	EC with no optimization	EC with conventional BIRCH	EC with conventional MBP	EC (Bi-GRU, CNN and DMN) + OLIHFA-BA
Precision	0.81814	0.812267	0.858096	0.891852
Specificity	0.800007	0.841113	0.848045	0.965926
MCC	0.80554	0.801643	0.841539	0.807754
FNR	0.304129	0.297652	0.316598	0.061094
F measure	0.808805	0.852687	0.818805	0.885329
FPR	0.393089	0.391654	0.382108	0.044074
NPV	0.811176	0.840228	0.852583	0.958836
Sensitivity	0.843219	0.825152	0.811886	0.878906
Accuracy	0.825021	0.832329	0.808423	0.969837

TABLE IV. ANALYSIS ON DICE AND JACCARD SCORES

	FCM	K-mean	Conventional BIRCH	Proposed BIRCH
Dice scores	0.525935	0.600694	0.506402	0.813585
Jaccard scores	0.494048	0.49131	0.501836	0.748445

VIII. Conclusion

We have developed a novel classification technique for rice leaf diseases in which the image was preprocessed using MF. The image was then segmented using BIRCH with enhancements. LBP, GLCM, colour, shape, and enhanced MBP-based features were recovered from segmented images. The data was then classified using three classifiers: Bi-GRU, CNN, and DMN. The results of the Bi-GRU, CNN, and DMN systems were averaged to determine the results. In addition, the Bi-GRU, CNN, and DMN schemes' weights were determined optimally using the OLIHFA-BA method. The 50th LP proposal produced the highest accuracy values (0.96), but other systems achieved low precision levels. This improvement in classifier analysis and optimization analysis employing EC (Bi-GRU, CNN, and DMN) + OLIHFA-BA is mostly attributable to the incorporation of enhanced features, segmentation, and classifiers. In the future, illness types should be analyzed.

Acknowledgment

I would like to express my deep gratitude to Professor Dr. K Kiran Kumar, my research supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. I'd also want to thank the technicians at department's laboratory for their assistance in providing me with the resources I needed to run the application.

References

[1] Krishnamoorthy N a, L.V. Narasimha Prasad b, C.S. Pavan Kumar c, Bharat Subedi d, Haftom Baraki Abraha e, Sathishkumar V E a, "Rice leaf diseases prediction using deep neural networks with transfer learning", Environmental Research11 May 2021Volume 198 (Cover date: July 2021) Article 111275.

[2] Prabira Kumar Sethy , Nalini Kanta Barpanda , Amiya Kumar Rath , Santi Kumari Behera, “Deep feature-based rice leaf disease identification using support vector machine”, Computers and Electronics in Agriculture, Volume 17517 , June 2020.

[3] Junde Chen, Defu Zhang, Adnan Zeb, Yaser A. Nanehkaran. “Identification of rice plant diseases using lightweight attention networks”, Expert Systems with Applications, Volume 169, 5 January 2021.

[4] Rallapalli, S., Saleem Durai, M.A, “A contemporary approach for disease identification in rice leaf”, Int J Syst Assur Eng Manag (2021).

[5] Zhencun Jiang, Zhengxin Dong , Wenping Jiang , Yuze Yang , “Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning”, Computers and Electronics in Agriculture 186 (2021) 106184 Available, Received 7 December 2020, Received in revised form 19 April 2021, Accepted 23 April 2021.

[6] Feng Jiang a, Yang Lu , Yu Chen , Di Cai , Gongfa Li, "Image recognition of four rice leaf diseases based on deep learning and support vector machine", Computers and Electronics in Agriculture 179 (2020) 105824 Available, Received 5 April 2020, Received in revised form 28 August 2020; Accepted 3 October 2020.

[7] GuoSheng Zhang, TongYu Xu, YouWen Tian, Han Xu, JiaYu Song, Yubin Lan, “Assessment of rice leaf blast severity using hyperspectral imaging during late vegetative growth”, Australasian Plant Pathology: 9 August 2020. p.571–578.

[8] Rahul Bakade, Kishor D. Ingole, Sanjay Deshpande, Garima Pal, Swathi S. Patil, Saikat Bhattacharjee et.al, “Comparative Transcriptome Analysis of Rice Resistant and Susceptible Genotypes to Xanthomonas oryzae pv. oryzae Identifies Novel Genes to Control Bacterial Leaf Blight” , Molecular Biotechnology, https://doi.org/10.1007/s12033-021-00338-3 Received: 19 March 2021 / Accepted: 11 May 2021.

[9] Zarbafi, S.S., Rabiei, B., Ebadi, A.A. et al. “Association mapping of traits related to leaf blast disease in rice (Oryza sativa L.)”, Australasian Plant Pathology, 49, 31–43 (2020).

[10] Lu, L., Yang, D., Tang, D. et al. “Transcriptome analysis of different rice cultivars provides novel insights into the rice response to bacterial leaf streak infection”, Funct Integr Genomics 20, 681–693 (2020).

[11] Long Tian , Bowen Xue , Ziyi Wang , Dong Li , Xia Yao , Qiang Cao , Yan Zhu , Weixing Cao , Tao Cheng ,” Spectroscopic detection of rice leaf blast infection from asymptomatic to mild stages with integrated machine learning and feature selection”, Remote Sensing of Environment 257 (2021) 112350, Received 31 August 2020; Received in revised form 5 February 2021; Accepted 9 February 2021.

[12] Haniyam Badi B. Manoj Kumar, Chikka Balli A. Deepa K, Kodihally M, Harini Kumar, M. P. Rajanna, Belthur Chethan A, “Molecular profiling of blast resistance genes and evaluation of leaf and neck blast disease reaction in rice”, Journal of Genetics, 2020.,

[13] Nan Jiang, Jun Fu, Qin Zeng , Yi Liang, Yanlong Shi, Zhouwei Li, Youlun Xiao, Zhizhou He, Yuntian Wu, Yu Long, Kai Wang, Yuanzhu Yang, Xionglun Liu, Junhua Peng. Genome-wide association mapping for resistance to bacterial blight and bacterial leaf streak in rice. Planta; 2021.

[14] Shrivastava, V.K., Pradhan, M.K. Rice plant disease classification using color features: a machine learning paradigm. J Plant Pathol 103, 17–26 (2021).

[15] Verma, T., Dubey, S, “Prediction of diseased rice plant using video processing and LSTM-simple recurrent neural network with comparative study”, Multimed Tools Appl 80, 29267–29298 (2021).

[16] Zhang, J., Lin, G., Yin, X. et al. “Application of artificial neural network (ANN) and response surface methodology (RSM) for modeling and optimization of the contact angle of rice leaf surfaces”, Acta Physiol Plant 42, 51 (2020).

[17] Sethy, P.K., Barpanda, N.K., Rath, A.K. et al. “Nitrogen Deficiency Prediction of Rice Crop Based on Convolutional Neural Network”, J Ambient Intell Human Comput 11, 5703–5711 (2020).

[18] Wani, J.A., Sharma, S., Muzamil, M. et al. “Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges”, Arch Computat Methods Eng (2021).

[19] Shu, X., Wang, A., Jiang, B. et al. “Genome-wide association study and transcriptome analysis discover new genes for bacterial leaf blight resistance in rice (Oryza sativa L.)”, BMC Plant Biol 21, 255 (2021).

[20] Rao, T.B., Chopperla, R., Methre, R. et al. “Pectin induced transcriptome of a Rhizoctonia solani strain causing sheath blight disease in rice reveals insights on key genes and RNAi machinery for development of pathogen derived resistance”, Plant Mol Biol 100, 59–71 (2019).

[21] Goel, S., Goswami, K., Pandey, V.K. et al, “Identification of microRNA-target modules from rice variety Pusa Basmati-1 under high temperature and salt stress”, Funct Integr Genomics 19, 867–888 (2019).

[22] Angel Prathyusha K., Mahitha Y., Prasanna Kumar Reddy N., Raja Rajeswari P, “A survey on prediction of suitable crop selection for agriculture development using data mining classification techniques”, International Journal of Engineering and Technology(UAE); 2018. p. 107-109.

[23] Vishnu B.V., Srinivas C. (2018),’Metaheuristic Algorithms Based Crop Classification’,Proceedings of the 3rd International Conference on Communication and Electronics Systems, ICCES 2018, (),PP. 1140-1144.

[24] Bhuyan H.K., Sirajul Huque M.D. (2018), “Sub-Feature Selection Based Classification”, Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018, (),PP. 210-216.

[25] Sridevi S., Bindu Prathyusha M., Krishna Teja P.V.S.J. User behavior analysis on agriculture mining system. International Journal of Engineering and Technology (UAE);7(2).2018. p. 37- 40.

[26] Balram G., Kiran Kumar K.(2018),”Smart farming: Disease detection in crops”,,International Journal of Engineering and Technology(UAE),7(2),PP. 33- 36.

[27] Kalavala S.S., Sakhamuri S., Prasad B.B.V.S.V. (2019), “An efficient classification model for plant disease detection”, International Journal of Innovative Technology and Exploring Engineering, 8(7), PP.126-129.

[28] Sakhamuri, S., Kumar, K.K, “ DeepLearning and Metaheuristic Algorithm for Effective Classification and Recognition of Paddy Leaf Diseases”, Journal of Theoretical and Applied Information Technology; 2022. p. 1127–1137.

[29] Sakhamuri, S., Kompalli, V.S, “An Overview on Prediction of Plant Leaves Disease using Image Processing Techniques”, IOP Conference Series: Materials Science and Engineering; vol. 981(2); 2020. P.022-024.

[30] https://en.wikipedia.org/wiki/Median_filter.

[31] Siddharth Madan, Kristin J. Dana1, “Modified balanced iterative reducing and clustering using hierarchies (m-BIRCH) for visual clustering”, THEORETICAL ADVANCES, Pattern Anal Applic, Received: 15 December 2013 / Accepted: 5 March 2015.

[32] Dhanashree S. Kalel , Pooja M. Pisal, Ramdas P. Bagawade, “Color, Shape and Texture feature extraction for Content Based Image Retrieval System: A Study”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 4, April 2016.

[33] Punal M. Arabi, Gayatri Joshi, N. Vamsha Deepa,"Performance evaluation of GLCM and pixel intensity matrix for skin texture analysis",Perspectives in Science, vol.8,pp.203-206,September 2016.

[34] Adel Hafiane, Guna Seetharaman, and Bertrand Zavidovique, “Median Binary Pattern for Textures Classification”, Springer-Verlag Berlin Heidelberg 2007, ICIAR 2007, LNCS 4633, pp. 387–398, 2007.

[35] Kuo-Chin Fan and Tsung-Yung Hung, "A Novel Local Pattern Descriptor—Local Vector Pattern in High-Order Derivative Space for Face Recognition", Ieee Transactions On Image Processing, vol. 23, no. 7, pp. 2877-89, July 2014.

[36] L. Tong, H. Ma, Q. Lin, J. He and L. Peng, "A Novel Deep Learning Bi-GRU-I Model for Real-Time Human Activity Recognition Using Inertial Sensors," IEEE Sensors Journal, doi: 10.1109/JSEN.2022.3148431.

[37] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu et.al, "Recent advances in convolutional neural networks", Pattern Recognition, vol. 77, pp354-377, 2018.

[38] M. Cai, Y. Shi and J. Liu, "Deep maxout neural networks for speech recognition," IEEE Workshop on Automatic Speech Recognition and Understanding, 2013, pp. 291-296.

[39] S, Arora & Singh, Satvir, “Butterfly optimization algorithm: a novel approach for global optimization”, Soft Computing, 2018.

[40] Li, Q.Q., He, Z.C. & Li, E. “The feedback artificial tree (FAT) algorithm”, Soft Comput 24, 13413–13440 (2020). https://doi.org/10.1007/s00500-020-04758-2.

[41] M. Marsaline Beno, Valarmathi I. R, Swamy S. M and B. R. Rajakumar, “Threshold prediction for segmenting tumour from brain MRI scans”, International Journal of Imaging Systems and Technology, Vol. 24, No. 2, pages 129-137, 2014, DOI: https://doi.org/10.1002/ima.22087.

[42] Devagnanam J,Elango N M, "Optimal Resource Allocation of Cluster using Hybrid Grey Wolf and Cuckoo Search Algorithm in Cloud Computing", Journal of Networking and Communication Systems, Vol.3,No.1, pp.31-40,2020.

[43] Renjith Thomas and MJS. Rangachar, "Hybrid Optimization based DBN for Face Recognition using Low-Resolution Images", Multimedia Research, Vol.1,No.1, pp.33-43,2018.

[44] SK.Mahammad Shareef and Dr.R.Srinivasa Rao, "A Hybrid Learning Algorithm for Optimal Reactive Power Dispatch under Unbalanced Conditions", Journal of Computational Mechanics, Power System and Control, Vol.1,No.1, pp.26-33,2018.

[45] Nandhini, S., Ashokkumar, K.,” Improved crossover based monarch butterfly optimization for tomato leaf disease classification using convolutional neural network”, Multimed Tools Appl 80, 18583–18610 (2021).

[46] Raja sekaran Thangaraj, S. Ananda murugan, Vishnu Kumar Kaliappan, “Automated tomato leaf disease classification using transfer learning-based deep convolution neural network”, Journal of Plant Diseases and Protection; 2020.

[47] Sridevi S,K Kiran Kumar,”A Novel Rice Leaf Disease Recognition and Classification Model: Neuro Fuzzy Model and Optimized DNN with Improved Kapur Segmentation”, unpublished.

[48] https://www.kaggle.com/minhhuy2810/rice-diseases-image-dataset.

Optimal Land-cover Classification Feature Selection in Arid Areas based on Sentinel-2 Imagery and Spectral Indices

Mohammed Saeed[1*], Asmala Ahmad[2], Othman Mohd[3]
Faculty of Communication and Information Technology, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia[1, 2, 3]
Geological Survey and Mineral Resources Authority, Sana'a, Yemen[1]

***Abstract*—Adding spectral indices to Sentinel-2 spectral bands to improve land-cover (LC) classification with limited sample size can affect the accuracy due to the curse of dimensionality. In this study, we compared the performance metrics of Random Forest (RF) classifier with three different combinations of features for land cover classification in an urban arid area. The first combination used the ten Sentinel-2 bands with 10 and 20 m spatial resolution. The second combination consisted of the first combination in addition to five common spectral indices (15 features). The third combination represented the best output of features in terms of performance metrics after applying recursive feature elimination (RFE) for the second combination. The results showed that applying RFE reduced the number of features in combination 2 from 15 to 8 and the average F_1-score indicator increased by nearly 8 and 6 percent in comparison with using the other two combinations respectively. The findings of this study confirmed the importance of feature selection in improving LC classification accuracy in arid areas through removing the redundant variable when using limited sample size and using spectral indices with spectral bands, respectively.**

***Keywords*—Feature selection; land cover; sentinel-2; arid areas; random forest; accuracy**

I. Introduction

Extracted information from satellite imagery regarding LC and its changes are essential in many applications. They are used as inputs for many models as hydrological models [1], ecosystem modelling [2] and land surface modelling [3]. Thus, the accuracy of LC is critical for these products, as it affects the final results of these models [4].

Adding auxiliary features to improve the classification of LC is a common practice in remote sensing community such as in Landsat imagery data [5], [6]. These auxiliary features include different spectral indices, topographic data, texture, and biophysical parameters [7], [8]. Using too many features to improve LC classification requires increasing the training samples size [9] to overcome the curse of dimensionality which negatively affects the classification accuracy and increases processing time [10]. In most cases, increasing training samples is not cost-effective, and this should be met by selecting only the most relevant features to achieve the optimal classification accuracy. Feature selection is very important in LC classification to overcome the high-dimensional data to increase class separation and compensate for the limited samples used for training classification models [9]. Moreover, feature selection removes the irrelevant and redundant variables which eventually helps in reducing the training data, decreasing the processing time and decrease the requirements of the data storage [11], [12]. In addition, feature selection documented to improve prediction performance and making data more interpretable [10].

According to [13], choosing a feature selection method subjects to various consideration such as stability, simplicity, requirements of computation, accuracy and the number of reduced features. In remote sensing applications, various feature selection methods were used in different spatial areas and for different purposes. Over a global LC classification, Relieff and max-min-associated methods were useful in decreasing computation time [14]. With sentinel-2 imagery, a comparative study among various feature selection methods concluded that similarity-based methods are the best in terms of F_1-score and the optimal features number for mapping landscapes infested by the Parthenium weed in South Africa, while wrapper methods were more accurate but with larger number of the selected features [15]. In northern Germany, grouped forward feature selection helped in data interpretation and reduced processing time in crop mapping. In spite of the several methods which were developed for selecting features, Reference [16] recommended using RFE, a wrapper method, in combination with RF for feature selection due to its stability and ability to improve classification. This method also proved to be efficient in improving accuracy in both the regression and classification processes by selecting the most relevant features [17], [18].

Arid regions are different from other spatial environments as they are dominant with less precipitation, dry climate, and scattered vegetation. Thus, ecosystems in urban arid areas are fragile [19]. These systems are not stable, and their change is rapid [20]. All these conditions are being captured by satellite sensors and translated into image pixels; therefore, need special attentions since these tend to affect the accuracy of LC classification to be produced later.

Since Sentinel-2 imagery was released in 2015, it has been widely used in producing LC maps due to its high spatial resolution, its temporal resolution (5 days), and its spectral wavelength ranging from visible to near-infrared, which helped to map and distinguishing LC classes [21].

*Corresponding Author.

In urban arid areas, there is no LC classification model based on the Sentinel-2 imagery for the selected study area in the Arabian arid Peninsula. Therefore, the objective of this study was to develop an optimal LC classification model for this area with cost-effective samples. This will include using RFE with RF classifier to choose the optimal relevant features from the combination of spectral bands and the most common indices. In addition, the effect of feature selection on processing time during training the model and prediction was investigated.

The paper is organised as follows: Section II describes the study area, data, and the methodology. In Sections III and IV, results and discussion are presented, respectively. Finally, Section V represents the conclusion of this study.

II. Materials and Methods

A. Study Site and Data

The study area was chosen as part of the tile number T38RPN (Fig. 1) from sentinel-2 satellite imagery which covers the metropolitan city of Riyadh, the capital of Kingdom of Saudia Arabia as a urban arid area. The image was selected on 4 July 2022 with ID: L1C_T38RPN_A027817_ 20220704T073152 when the cloud is minimum, and the selected part was considered to represent the variations in LC classes. While Sentinel-2 has 13 bands as shown in, only 10 bands were used in this study and the 60-meter spatial resolution bands related to atmospheric and cloud detection were dropped.

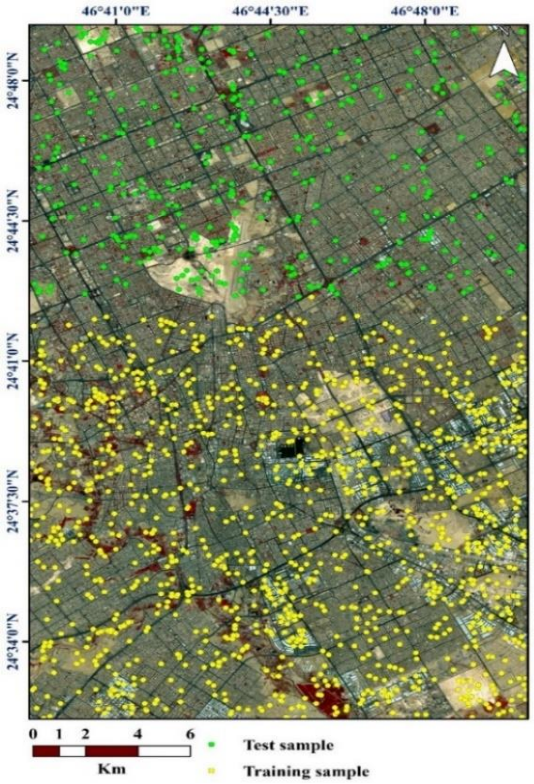


Fig. 1. Location of the study area.

B. Data Preprocessing and Preparation

In order to achieve the optimal accuracy, three steps have been performed on the raw image before clipping to the study area. First, atmospheric and topographic correction were carried out to convert the digital numbers to surface reflectance values using the FORCE algorithm [20] and to remove the effects of shadow, respectively. This preprocess step is initial and proved to contribute in improving LC accuracy. The SRTM digital elevation model (DEM) from EarthExplorer was used for topographic correction. Second, downscaling the 20-meter spatial resolution bands to 10 metre using nearest neighbor technique, which proved to be more accurate than other techniques in terms of producing LC classification accuracy [21]. After that, the image was cropped to the study area shapefile using QGIS software, version 3.

C. Classification System and Sampling

The selection of LC classes was based on the basis that confirms the inclusion of the main land types in the study area with reference to previous studies [22], [23]. In this study, the urban class has been divided into three categories: roads, industrial and building where spectral differences are unique. Table I shows the classes with their representative numbers in the selected study area.

The stratified random sampling method was used to collect training and testing samples. All samples were collected based on the per-pixel as a classification unit to avoid spatial-autocorrelation and reduce redundant data. The choice of samples was based on visual interpretation on the high spatial resolution Google Earth maps with intensive field work for validation. The number of training samples was determined to be in the range of 10–30 times the number of bands used for classification [24]. The test samples were 30% of the total samples and independent of the training samples. Fig. 1 and Table I show the distribution of training, testing samples, and their numbers in the study area, respectively.

D. Spectral Indices

In this urban arid study area, the effect of adding the following spectral indices on accuracy was investigated:

- The normalized difference vegetation index (NDVI) which is used as a monitoring and measuring index for vegetation cover from satellite imagery.
- The normalised difference built-up index (NDBI) is used to distinguish built surfaces, which receive positive values, from bare soils.
- The modified normalised difference water index (MNDWI) was proposed to detect superficial water. However, due to the relation between SWIR and wetness in soils, it can be also used to detect water in surfaces of vegetation or soil.
- The bare soil index (BSI) was proposed to enhance differentiation between bare and built-up lands.
- The soil adjusted vegetation index (SAVI): this index way to fit NDVI index to background average reflectance and minimises shadow effects.

TABLE I. LAND COVER CLASSES WITH THEIR TRAINING AND TESTING NUMBER AND DESCRIPTION

Classes	Sampling	
	Training (70%)	***Testing (30%)***
Vegetation	452	138
Roads	478	144
Bare land	474	142
Built-up	534	164
Industrial	530	160
Water	146	44
Total	2614	792

E. Classification Process and Evaluation

In order to investigate the effect of dimensionality reduction on LC classification accuracy in this study, we compared the performance of RF with three different combinations of features. The first combination consisted of the original ten spectral bands. The second combination consisted of the same features of the first combination in addition to the fifth spectral indices already mentioned in the previous subsection 2.5. The third combination represented the subset of the features which achieved the best performance metrics after applying the RFE selection method on the second combination. We referred to these combinations as model-1, model-2 and model-3 in the whole paper.

In each model, RF was used for classification due to its accurate results with less time, less sensitivity to overfitting, and because it requires few internal parameters to be tuned. RF is one of the most supervised ML algorithms widely used in both regression and classification and it can work with continuous and categorical data [26]. It belongs to the family of ensemble learning classifiers which depends on the bagging mechanism. The ntree and mtry are the most two internal parameters in the RF classifier. Each tree in FR model acts as a decision in the classification or regression process and the number of these decision trees is known as ntree and determined by the selected features from the user. In this study, the ntree was set to 500 as recommended by [27]. The mtry parameter refers to the predictors number that are randomly sampled when creating the trees at each split. In this study, mtry was set to the number of square root of the variables used as inputs for classification in each model [28].

The validation process for each model was carried out, using a 10-fold cross validation technique to avoid bias in results and conclusions. In this technique, data set is divided into 10 subsets. Next, a model is trained using a subset formed by combining these nine subsets and tested using the remaining subset. This is done 10 times each using a different subset as a test set and calculating the test set error.

The evaluation for all models was based on the performance metrics represented by the overall accuracy (OA), the user's (UA), and producer's (PA) accuracies, which were calculated from the confusion matrix. In addition, F_1-score was calculated as a balance accuracy measurement [29] and used in this study as the main index for comparing the models.

In order to explore the contribution of each feature to the improvement of the LC accuracy for all models, the built-in variable importance property of the RF classifier (randomForest package) was analysed. Thus, a useful reference can be provided to choose the appropriate features as input variables in other studies in the selected study area.

The last step of evaluation was the computational time analysis which included comparing the average processing time for the three models with their different number of features. The average of 10 running times was used in this study for processing time of the models during training and when the models were used to predict the whole image of the study area. All analyses were carried out using R programming language (version 3.6.1). We used a laptop with Intel® Core™ i7-7700HQ CPU @ 2.80GHz × 8 and 32 GiB memory in Ubuntu 20.04.5 LTS operating system.

III. RESULTS

A. Model Evaluation by Overall, User's, Producer's Accuracy Matrices and Variable Importance

1) Model-1: Table II shows the confusion matrix when the classification model used only the original 10 spectral bands. The overall accuracy of the model was 81%. Generally speaking, the greatest misclassification has occurred between vegetation and built-up classes, on the one hand, and between industrial and bare classes, on the other hand.

In this model and as shown in Fig. 2, the PA and UA are variated in values among the classes. The PA for the built-up class was the maximum with 98.8 percent, while the water class had the minimum value with 22.7 percent. Regarding the UA, the industrial and built-up classes were the highest and lowest values with 95.8 and 70.4 respectively.

Fig. 3 shows the importance of the features and their contributions in the accuracy. It is clearly that b3, b4, b12, b8 and b2 are the first five bands that contributed to achieving most of classification accuracy improvement while b6 was the less important regarding contribution in classification accuracy.

2) Model-2: Table III shows the confusion matrix when the classification model used the tenth spectral bands with the five spectral indices. The overall accuracy of this model increased by nearly 2 % compared to the previous model.

TABLE II. CONFUSION MATRIX OF MODEL-1

Classes		Predicted Values						Total
Actual values	Vegetation	82	10	2	42	0	2	138
	Roads	2	136	0	6	0	0	144
	Bare land	0	5	134	0	3	0	142
	Built-up	0	2	0	162	0	0	164
	Industrial	0	0	42	4	114	0	160
	Water	6	10	0	16	2	10	44
	Total	90	163	178	230	119	12	792
Overall accuracy 80.56 %								

TABLE III. CONFUSION MATRIX OF MODEL-2

Classes		Predicted values						Total
Actual values	Vegetation	90	7	3	38	0	0	138
	Roads	0	134	0	10	0	0	144
	Bare land	0	4	136	0	2	0	142
	Built-up	0	2	2	160	0	0	164
	Industrial	0	0	32	2	126	0	160
	Water	4	11	0	19	0	10	44
	Total	94	158	173	229	128	10	792
Overall accuracy 82.83 %								

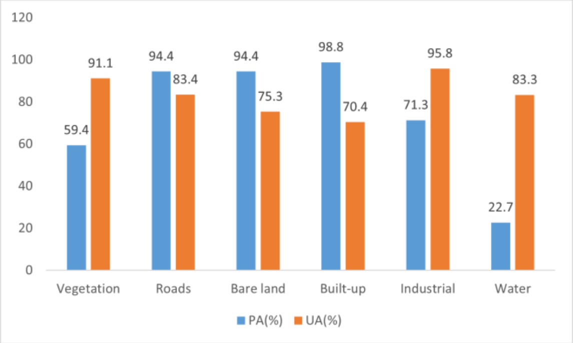


Fig. 2. Producer's and User's accuracy in model-1.

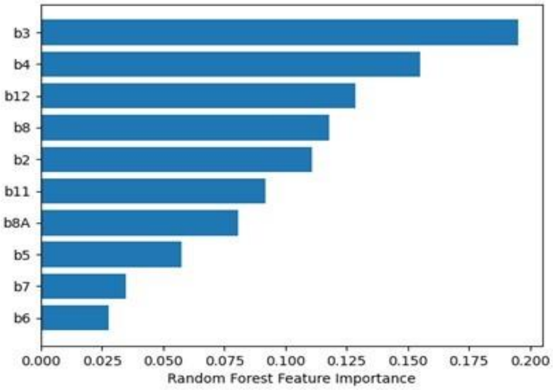


Fig. 3. Variable importance in model-1.

This improvement in overall accuracy can be interpreted by the contribution of the indices in improving the per-class accuracy.

As shown in Fig. 4, the PA and UA accuracy in this model are variated. The highest and lowest PA values were registered for the built-up and water classes, respectively, while the highest and lowest UA values were registered for water and industrial classes, respectively.

The variable importance in this model is shown in Fig. 5. From the first five features, two spectral indices: SAVI and BSI are the most important in classification accuracy. It is noticeable that NDVI has a medium importance in improving accuracy, while MNDWI and NDBI have less importance in improving classification accuracy.

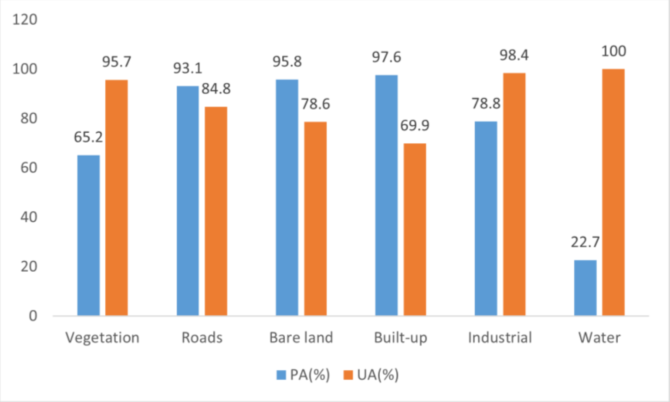


Fig. 4. Producer's and User's accuracy in model-2.

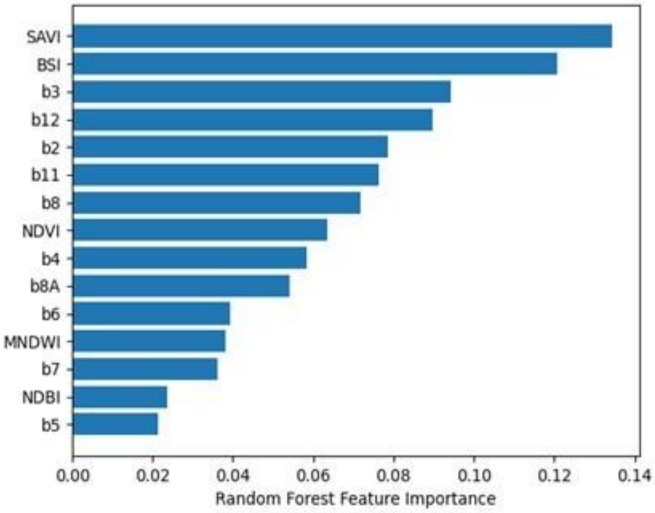


Fig. 5. Variable importance in model-2.

3) Model-3: The confusion matrix after applying the RFE to subset the feature with the best accuracy is shown in Table IV. The best accuracy derived from this model was 85.98 % using only eight features: six spectral bands and two spectral indices.

Comparing the previous two models, the accuracy in this model increased by nearly 5 and 3 percent, respectively. Most of the accuracy improvement was in both vegetation, industrial and water classes where the number of the correct instances increased by 8, 4 and 9 respectively in comparison with the same classes in the previous model.

In this model, the PA ranged between 43.2 % for the water class and 97.6 % for the built-up class. In terms of UA, the water class ranked the best, while the built-up class ranked the lowest (Fig. 6).

The first five feature importance in this model included four spectral bands: b3, b12, b2, and b8 and only one spectral index: BSI as shown in Fig. 7. After applying RFE, the feature numbers decreased nearly to half in comparison with the previous model.

TABLE IV. CONFUSION MATRIX OF MODEL-3

Classes		Predicted values						Total
Actual values	Vegetation	98	7	5	28	0	0	138
	Roads	0	137	5	2	0	0	144
	Bare land	0	3	137	0	2	0	142
	Built-up	2	2	0	160	0	0	164
	Industrial	0	0	23	7	130	0	160
	Water	3	8	0	14	0	19	44
	Total	103	158	169	213	133	19	792
Overall accuracy 85.98 %								

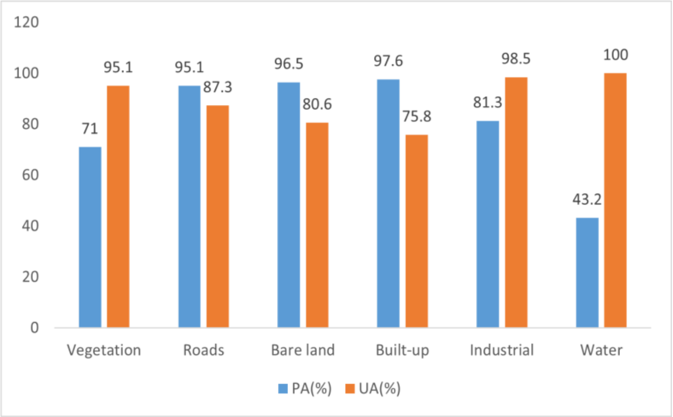


Fig. 6. Producer's and User's accuracy in model-3.

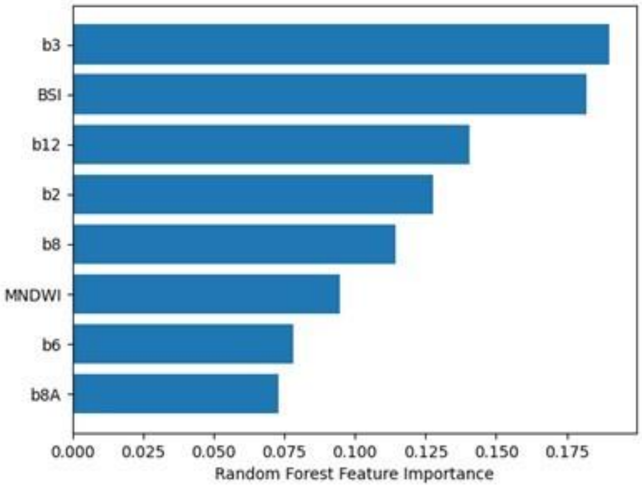


Fig. 7. Variable importance in model-3.

B. Model Evaluation by F_1-score Accuracy Metric

Fig. 8 shows the value of F_1-score accuracy for the three models compared in this study. It is very clear that the F_1-score value for most of the classes was increased from model-1 to model-3. This increase was noticeable in most of the classes after adding the spectral indices in model-2 with reference to the initial model-1 where spectral bands were only used. For instance, F_1-score in vegetation and industrial classes increased by nearly 6 % in model-2 when compared with model-1.

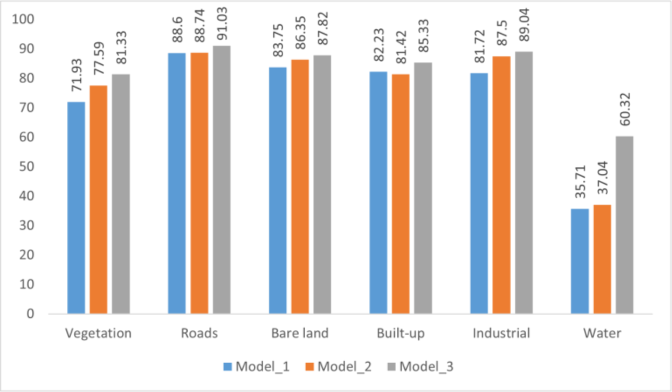


Fig. 8. F_1-score accuracy in the three models.

After applying the RFE to choose the optimal features, the F_1-score increased with different percentage between classes when compared to the values in model-2. The water class achieved the highest increase by 23 %, while the bare land class achieved the lowest increase by 1.67 %.

C. Computation Time Evaluation

Fig. 9 compares the average processing time spent for training and predict in each model aligned with the number of features used. The model-3 achieved the best rank in terms of the training and prediction processing time, where its average reached 1.56 and 1.83 mins respectively. This model used the lowest number of features (6 spectral bands plus 2 spectral indices). On the other hand, the highest training and prediction processing time was associated with model-2, which has the largest number of features (10 spectral bands plus 5 spectral indices).

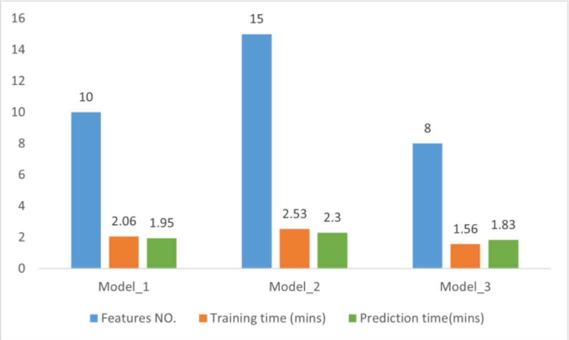


Fig. 9. Feature number, training, and prediction time in the models.

IV. DISCUSSION

This study aimed to explore the effect of feature selection on LC classification accuracy and processing time in arid areas aligned with limited sample size. In terms of the overall accuracy, average PA, UA and F_1-score as shown in Fig. 10, there was an increase in accuracy after adding the spectral indices to the spectral bands. In addition, there is a noticeable increase in the accuracy of all these metrics in model-3 after applying the RFE feature selection technique.

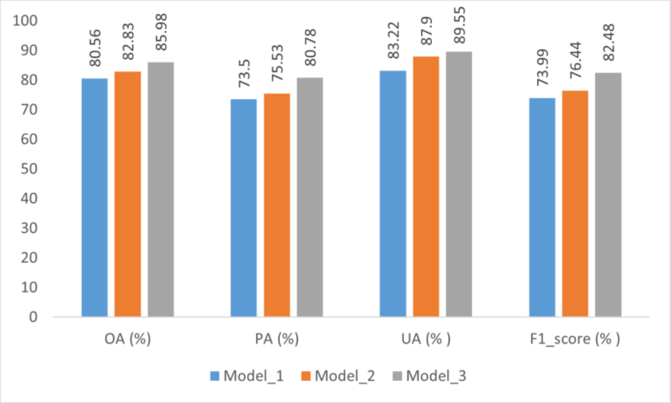


Fig. 10. Average of accuracy indicators for the three models.

Adding spectral indices proved to be effective in improving LC classification accuracy in model-2 in this study by increasing the separability between the individual classes in comparison with model-1. In this study, the SAVI and BSI indices were the two most important features in terms of improving classification accuracy in model-2. The fact of improving accuracy through adding indices is a common practice in many other studies such as in [25]–[27].

Despite the added value of indices in improving LC classification accuracy in model-2, the application of feature selection in model-3 proved to be more effective in improving all the performance metrics in model-3 and in decreasing the processing time without the need for increasing sample size. This could be interpreted by the importance of applying feature selection technique in removing redundant features which affect both the accuracy and processing time [28]. Many studies in terms of improving LC classification recommended applying feature selection methods to select the optimal relevant features and reduce the processing time [29].

The low accuracy of model-2 in comparison with model-3 indicates that the curse of dimensionality can affect the classification accuracy. Previous studies showed that increasing the number of features can lead to complexity by increasing the processing time and decreasing the potential accuracy of the model [30].

The application of RFE in combination with the property of RF variable importance in this study helped in determining the input features and their contribution in producing the optimal classification accuracy in the study area. This subset of the relevant features is a common appropriate approach for building robust learning models [31].

V. Conclusion

With a limited sample size for LC classification, adding spectral indices to improve the classification accuracy is not an ideal solution, as shown in this study. The feature selection techniques proved to overcome the limited size of samples by choosing the relevant features that increase the classes separability. In urban arid areas, the RFE technique decreased the features from 15 to 8 with best F_1-score average accuracy (82.48%) in comparison with the case when only spectral features were used in model-1 (73.99%) or when the spectral bands and indices were used in model-2 (76.44%). Furthermore, less training and prediction processing time was achieved after applying RFE (1.56 and 1.83 min) when comparing with values of model-1 (2.06 and 1.95 min) and with values in model-2 (2.53 and 2.3 min). The combination of the spectral bands: b2, b3, b6, b8, b8a, and b12 with the spectral indices: BSI and MNDWI represent the optimal variables for LC classification in terms of accuracy and computation time in this geographic study area.

The results of this study showed that feature selection is useful in reducing the dimensionality of spectral bands of Sentinel-2 and the spectral indices as well. This refers that not all indices can contribute to improving classification accuracy when sample size is limited.

Other feature selection techniques are recommended to be explored and compared in alignment with the other machine learning classifiers in urban arid areas. In addition, more multitemporal images for different seasons can be investigated to overcome the single image used in this study.

Acknowledgment

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) for all the support and cooperation given in completing this research.

References

[1] C. Chirachawala, S. Shrestha, M. S. Babel, S. G. Virdis, and S. Wichakul, “Evaluation of global land use/land cover products for hydrologic simulation in the Upper Yom River Basin, Thailand,” Sci. Total Environ., vol. 708, p. 135148, 2020.

[2] W. Chen, H. Zhao, J. Li, L. Zhu, Z. Wang, and J. Zeng, “Land use transitions and the associated impacts on ecosystem services in the Middle Reaches of the Yangtze River Economic Belt in China based on the geo-informatic Tupu method,” Sci. Total Environ., vol. 701, p. 134690, 2020.

[3] K. Hibbard et al., “Research priorities in land use and land-cover change for the Earth system and integrated assessment modelling,” Int. J. Climatol., vol. 30, no. 13, pp. 2118–2128, 2010.

[4] C. Xu, “Issues influencing accuracy of hydrological modeling in a changing environment,” Water Sci. Eng., vol. 14, no. 2, pp. 167–170, 2021.

[5] P. Hurskainen, H. Adhikari, M. Siljander, P. Pellikka, and A. Hemp, “Auxiliary datasets improve accuracy of object-based land use/land cover classification in heterogeneous savanna landscapes,” Remote Sens. Environ., vol. 233, p. 111354, 2019.

[6] L. Qu, Z. Chen, M. Li, J. Zhi, and H. Wang, “Accuracy improvements to pixel-based and object-based lulc classification with auxiliary datasets from Google Earth engine,” Remote Sens., vol. 13, no. 3, p. 453, 2021.

[7] C. Hong, X. Jin, J. Ren, Z. Gu, and Y. Zhou, “Satellite data indicates multidimensional variation of agricultural production in land consolidation area,” Sci. Total Environ., vol. 653, pp. 735–747, 2019.

[8] I. Klein, U. Gessner, A. J. Dietz, and C. Kuenzer, “Global WaterPack–A 250 m resolution dataset revealing the daily dynamics of global inland water bodies,” Remote Sens. Environ., vol. 198, pp. 345–362, 2017.

[9] Y. Huang, C. Zhao, H. Yang, X. Song, J. Chen, and Z. Li, “Feature selection solution with high dimensionality and low-sample size for land cover classification in object-based image analysis,” Remote Sens., vol. 9, no. 9, p. 939, 2017.

[10] S. Georganos et al., “Less is more: Optimizing classification performance through feature selection in a very-high-resolution remote sensing object-based urban application,” GIScience Remote Sens., vol. 55, no. 2, pp. 221–242, 2018.

[11] C. Xu et al., “A Comprehensive Comparison of Machine Learning and Feature Selection Methods for Maize Biomass Estimation Using

Sentinel-1 SAR, Sentinel-2 Vegetation Indices, and Biophysical Variables," Remote Sens., vol. 14, no. 16, p. 4083, 2022.

[12] D. Dobrinić, M. Gašparović, and D. Medak, "Evaluation of Feature Selection Methods for Vegetation Mapping Using Multitemporal Sentinel Imagery," Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 43, pp. 485–491, 2022.

[13] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Comput. Electr. Eng., vol. 40, no. 1, pp. 16–28, 2014.

[14] L. Yu, H. Fu, B. Wu, N. Clinton, and P. Gong, "Exploring the potential role of feature selection in global land-cover mapping," Int. J. Remote Sens., vol. 37, no. 23, pp. 5491–5504, 2016.

[15] Z. Kiala, O. Mutanga, J. Odindi, and K. Peerbhay, "Feature selection on sentinel-2 multispectral imagery for mapping a landscape infested by parthenium weed," Remote Sens., vol. 11, no. 16, p. 1892, 2019.

[16] X. Fan et al., "Sentinel-2 Images Based Modeling of Grassland Above-Ground Biomass Using Random Forest Algorithm: A Case Study on the Tibetan Plateau," Remote Sens., vol. 14, no. 21, p. 5321, 2022.

[17] R. R. Pullanagari, G. Kereszturi, and I. Yule, "Integrating airborne hyperspectral, topographic, and soil data for estimating pasture quality using recursive feature elimination with random forest regression," Remote Sens., vol. 10, no. 7, p. 1117, 2018.

[18] G. Ewa, F. David, and O. Katarzyna, "Evaluation of machine learning algorithms for forest stand species mapping using Sentinel-2 imagery and environmental data in the Polish Carpathians," Remote Sens. Environ., vol. 251, no. 112103, p. 7, 2020.

[19] H. Xie, Y. Zhang, Z. Wu, and T. Lv, "A bibliometric analysis on land degradation: Current status, development, and future directions," Land, vol. 9, no. 1, p. 28, 2020.

[20] M. Lin and C. Chen, "Application of fuzzy models for the monitoring of ecologically sensitive ecosystems in a dynamic semi-arid landscape from satellite imagery," Eng. Comput., vol. 27, no. 1, pp. 5–19, Jan. 2010, doi: 10.1108/02644401011008504.

[21] M. Drusch et al., "Sentinel-2: ESA's optical high-resolution mission for GMES operational services," Remote Sens. Environ., vol. 120, pp. 25–36, 2012.

[22] A. F. Alqurashi, L. Kumar, and P. Sinha, "Urban land cover change modelling using time-series satellite images: A case study of urban growth in five cities of Saudi Arabia," Remote Sens., vol. 8, no. 10, p. 838, 2016.

[23] M. Rahman and R. Planning, "Land use and land cover changes and urban sprawl in Riyadh, Saudi Arabia: An analysis using multi-temporal Landsat data and Shannon's Entropy Index," Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 41, pp. 1017–1021, 2016.

[24] C. Li, J. Wang, L. Wang, L. Hu, and P. Gong, "Comparison of classification algorithms and training sample sizes in urban land classification with Landsat thematic mapper imagery," Remote Sens., vol. 6, no. 2, pp. 964–983, 2014.

[25] P. Ettehadi Osgouei, S. Kaya, E. Sertel, and U. Alganci, "Separating built-up areas from bare land in mediterranean cities using Sentinel-2A imagery," Remote Sens., vol. 11, no. 3, p. 345, 2019.

[26] G. Kuc and J. Chormański, "Sentinel-2 imagery for mapping and monitoring imperviousness in urban areas," Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. 42, pp. 43–47, 2019.

[27] B. Kavhu, Z. E. Mashimbye, and L. Luvuno, "Climate-based regionalization and inclusion of spectral indices for enhancing transboundary land-use/cover classification using deep learning and machine learning," Remote Sens., vol. 13, no. 24, p. 5054, 2021.

[28] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," presented at the 2014 science and information conference, 2014, pp. 372–378.

[29] M. Kganyago, J. Odindi, C. Adjorlolo, and P. Mhangara, "Selecting a subset of spectral bands for mapping invasive alien plants: a case of discriminating Parthenium hysterophorus using field spectroscopy data," Int. J. Remote Sens., vol. 38, no. 20, pp. 5608–5625, 2017.

[30] P. Gong and P. Howarth, "An assessment of some factors influencing multispectral land-cover classification," Photogramm. Eng. Remote Sens., vol. 56, no. 5, pp. 597–603, 1990.

[31] H. Fei et al., "Cotton classification method at the county scale based on multi-features and Random Forest feature selection algorithm and classifier," Remote Sens., vol. 14, no. 4, p. 829, 2022.

An Add-on CNN based Model for the Detection of Tuberculosis using Chest X-ray Images

Roopa N K[1], Mamatha G S[2]
Research Scholar, Dept. of Computer Science & Engineering, Sri Siddhartha Institute of Technology, Tumakuru, India[1]
Dept. of Information Science and Engineering, RV College of Engineering, Bangalore, India[2]

Abstract—**Machine Learning has been potentially contributing towards smart diagnosis in the medical domain for more than a decade with a target towards achieving higher accuracy in detection and classification. However, from the perspective of medical image processing, the contribution of machine learning towards segmentation is not been much to find in recent times. The proposed study considers a use case of Tuberculosis detection and classification from chest x-rays where a unique machine learning approach of Convolution Neural Network is adopted for segmentation of lung images from CXR. A computational framework is developed that performs segmentation, feature extraction, detection, and classification. The proposed system's study outcome is analyzed with and without segmentation over existing machine learning models to exhibit 99.85% accuracy, which is the highest score to date in contrast to existing approaches found in the literature. The study outcome based on the comparative analysis exhibits the effectiveness of the proposed system.**

Keywords—Chest X-Ray; machine learning; convolution neural network; segmentation; detection; classification

I. Introduction

Tuberculosis (TB) is a disease caused due to the bacteria Mycobacterium tuberculosis [1]. Most often, this affects the lungs and causes tissue damage in them, while its common symptom is found to be cough. These bacteria can spread through the air; hence they will spread from person to person when the infected person coughs, sneezes, or spits [2]. According to WHO, one-fourth of the world's population is already infected with these bacteria, but they are not ill and cannot transmit it further [3]. People infected with TB have a 5-10% lifetime risk of falling ill due to it. When the person falls ill due to TB, it shows immediate symptoms like cough, blood in the cough, high fever, chest pain, and in some cases, mental illness. In order to treat this, finding out the extent of damage caused to the lungs is most important. Early diagnosis is crucial for treatment.

TB is also most commonly confused with lung cancer due to all similar symptoms. When doctors have such confusion, they will go for a TB skin test and blood test; however, such a test doesn't exhibit the criticality of lung damage. Hence, Chest X-Ray (CXR) is commonly adopted for the identification of stages of the criticality of TB [4]. This is where the domain of medical image processing comes into play, where various algorithms and mechanisms are constructed in order to find all the indicative symptoms of abnormalities in the lung region when the subject is infected by TB. Some of the possible problems in this stage of acquisition of CXR image are that there are higher possibilities of artifacts in that acquired CXR image [5]. This could be due to lower or fluctuating illumination conditions, absence of some significant region of lungs, movement of the subject during taking X-rays, etc. The conclusive remarks to state that a subject is confirmed of TB completely depends upon a manual analysis of the doctor. Because of this manual assessment, it's humanly impossible for any doctor to diagnose the disease for a large number of patients per day. Therefore, there is a serious need to make this identification system smarter and more intelligent, which can diagnose the disease without human intervention, which is manually not possible for humans. This is where artificial intelligence and machine learning come into the picture [6]. Currently, much research is being carried out toward adopting machine learning in diagnosing critical diseases [7]-[9]. There are also some significant studies where common diseases and diseases associated with COVID-19 have been assessed in recent times [10].

However, there are not many recent studies investigating TB. It has also been noticed that detection and classification completely depend upon the efficiency of processing input and extracted features. One such technique is known as segmentation, which differentiates the background image from the foreground image, offering more clearance to the system or a physician to make a conclusive decision about the diagnosis.

This paper introduces a novel computational model to perform segmentation and thereby contribute towards detecting and classifying CXR in TB patients. The contribution of the study is i) an augmented U-Net model is presented to carry out segmentation, ii) a computationally efficient training model is introduced, iii) a simplified preprocessing is carried out to leverage accuracy, and iv) accomplishing higher accuracy of classification till date. The organization of the manuscript is as follows: Discussion of existing literature where Machine learning is used for identification and classification is carried out in Section II followed by exclusive highlights of the research problem in Section III; Section IV showcases the research methodology, while Section V discusses system implementation. Section VI discusses the results obtained from the study. Finally, Section VII makes conclusive remarks about the study's contribution.

II. Related Work

This Section discusses the different mechanisms carried out in the existing system toward analyzing CXR images. The prime emphasis is given to the latest publication with different techniques of detection and classification of CXR. Although the proposed study targets the detection and classification of Tuberculosis, this section mainly studies all potential approaches toward screening any form of significant deformities in CXR. This is meant to assist more information that could be assistive towards the detection and classification of Tuberculosis too.

The recent work carried out by Lin et al. [11] has presented a detection of COVID-19 symptoms right from CXR images using an adaptive attention network using ResNet. Although this work aims to extract contextual information for lesion detection in CXR, its limitation is that it is quite an iterative process in the course of learning the network. Another recent work by Wu et al. [12] has used a unique classification technique using the fractional order of convolution approach. The method also makes use of a radial Bayesian network in order to identify complex structures. However, the limitation of the model is that it involves a higher number of layered operations to perform classification. A study towards segmentation has been reported by Eslami et al. [13] that uses an adversarial network with conditional generative attributes where a network is constructed for all the pixels. Although the model claims support for multitasking applications, its limitation remains in complex architecture design over many pixels. A study towards adopting an adversarial network is also witnessed in work carried out by An et al. [14]. The study has developed a multi-appearance model for carrying out the extraction of significant features associated with COVID-19.

Further, the model uses a design of an adaptive network for a multi-scale adversarial domain for targeting better accuracy. The study's limitation is using a complex structure of an adaptive network irrespective of an accuracy of 98.83%. Londono et al. [15] present a unique evaluation-based model where a Convolution Neural Network (CNN) is used for training the dataset of CXR with COVID-19 symptoms. The study also performs preprocessing of data to deal with variability issues during analysis. The limitation of the study is lowered accuracy score of 91.5% while ignoring the process of masking the images. The problems associated with detecting lesion in CXR is carried out by Li et al. [16] where an amplitude modulation scheme has been presented to extract deformation features within CXR with deformable convolution. The study also uses the loss function of regression in order to carry out optimization; however, a lowered precision score of 0.914 is obtained as a limitation. A unique form of machine learning approach called a deep zoom neural network is presented by Wang et al. [17] that targets optimizing the training process using CXR images for thoracic diseases. The study has used U-Net in order to carry out segmentation followed by using attention heatmap, the regions of lesions are extracted. The limitation of the study is that it is more inclined toward detection and possesses fewer conditional constraints for supporting classification. The study of tuberculosis detection is presented by Rahman et al. [18] using deep CNN for classification. The study's limitation is its accuracy of 98.6% without much emphasis on the masking process. Further study towards segmentation of CXR images is carried out by Munawar et al. [19], focusing on using the generative adversarial network. This study carries out segmentation masking while the model training is carried out using multiple discriminators. The limitation of the study is its lower accuracy, irrespective of using the sophisticated adversarial network. It is also observed that most of the existing system focuses more or less on detection or classification; however, not much emphasis is offered to outlier detection, which is essential to confirm its accuracy. A study in such a direction is carried out by Kim et al. [20] using artificial intelligence. The study also uses a Recurrent Neural network to perform learning operations. The limitation of the model is the inclusion of higher convolution layers; still, extraction of potential features doesn't carry any objective function.

Paluru et al. [21] have presented a study of chest CT images, especially focusing on the segmentation process using a unique form of CNN called as Anam-Net. The study claims to include lower parameters to show its lightweight features. The limitation of the model is that it still has a dependency on high-end resources in order to make it functional. Synthesis is one of the essential operations for analyzing CXR, while a study in this direction is carried out by Salehinejad et al. [22]. The study has used CNN, whose outcome shows the supportability of five classes of CXR. The limitation of this model is that it lacks potential preprocessing and planning while taking input for real- images. Catala et al. [23] have investigated CXR images for the identification of Pneumonia with more emphasis on the dataset. Work carried out by Zhang et al. [24] has carried out a segmentation process for CXR using deep learning, which is capable of generating information about infection over the lungs. The limitation of the study is its lowered accuracy score of 95.9%. The work carried out by Zhou et al. [25] used ResNet and Support Vector Machine (SVM) in order to detect abnormalities in CXR images. The study also uses image regrouping, where encoders of deep networks extract the features. The limitation of the study is its lowered accuracy of 93% only. The work carried out by Zaidi et al. [26] has used a tailored make CNN model to perform lung segmentation. The limitation of the study is its inclusion of a higher number of iterations in order to achieve below-average accuracy. Similarly, there are various other works in the same problem domain carried out by Wu et al. [27], Yan et al. [28], Fan et al. [29], and Lian et al. [30]. Therefore, various detection and classification mechanisms exist in current times towards CXR images with reported claims of accuracy outcome and limitation. The next section highlights identified research problems.

III. RESEARCH PROBLEM

After reviewing the existing system to analyze CXR images, the following are the open-end research problems being identified:

- Existing models for the detection of lung abnormalities have mainly found lung anatomy, which is preferentially applicable to low-level image processing. However, such methods often produce inferior segmentation when certain areas of the lung are malformed or missing from CXR.

- Machine learning techniques are mainly used for classification, mainly for feature extraction. CNN is one such dominant machine learning model. However, existing studies using this technique were generally found to be computationally slow and expensive due to higher iterations and resource involvement. It also does not help extract information on proper thresholding, which could be a potential problem during training operations. Other than that, there is not much emphasis on consistency and scalability since the uncertainty increases when the segmented regions change to different resolutions. Beyond that, not many techniques have been used to classify specific regions of the lung and its connected organs. From this perspective, research on segmentation is rather sparse.

- Existing studies have adopted the distortion model for the investigation of CXR. They were primarily used for the segmentation of lung fields in CXR; however, their performance is not up to the mark for large networks or high numbers of training images. There is less evidence to be verified at the same time, which reduces applicability in a practical world scenario. Furthermore, the mechanism to initialize the lung model during Tuberculosis has also been found to be manual and error-prone.

- There is no denying the fact that adopting CNN or other machine learning approaches gives good classification performance. However, the participation of machine learning approaches to perform autonomous segmentation is much lower than in existing systems. Without a proper segmentation considering all possible constraints of the CXR images, an improved form of identification and classification cannot be made. Furthermore, most machine learning approaches applied to CXR images have achieved low accuracy scores, while there is still scope to optimize them. Therefore, existing studies are found to be more inclined to implement complex architectures on machine learning rather than trying simplified modeling of it.

Therefore, the problem statement of the proposed study is "Optimized usage of machine learning towards identification of abnormalities in CXR images with more focus on segmentation approach is quite a challenging task".

IV. Proposed System

The research work reported in this paper aims to develop an efficient and robust computational model that can accurately identify and classify TB disease using chest X-ray (CXR) images. As it has been identified based on review of literature that there is wide adoption of CNN for medical image analysis such as detecting tuberculosis from chest X-ray images. However, due to the complexity of CXR images, which include detailed information about the shoulder bones, rib cage, and outer body of the person, direct usage of CNN on these images may not yield very accurate results. In this regard, the proposed work presents a highly integrated system with more optimized computing operation in automating the task of TB diagnosis. The schematic architecture of the proposed system is illustrated in Fig. 1.

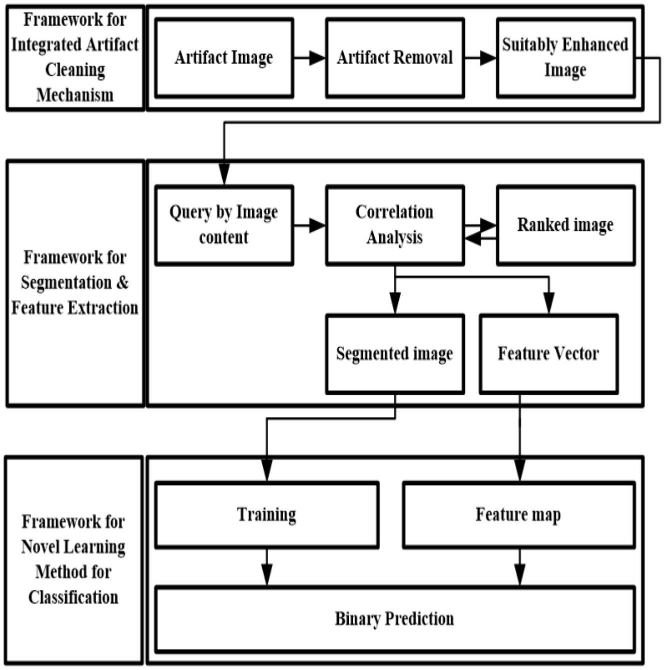


Fig. 1. Block based Architecture of the propsoed system.

The proposed system aims to accurately diagnose TB by using a combination of exploratory data analysis, image segmentation, and machine learning. Firstly, the system executes exploratory data analysis to understand the data and determine the preprocessing requirements to make the input data suitable for learning models. This includes data cleaning, artifact removal normalization, and feature engineering.

Secondly, the system performs image segmentation to extract the region of interest from the input CXR image. This is done using a customized and enhanced version of the UNet Model, which accurately locates the lungs and masks out residual regions. This segmentation process reduces the surface of computational complexity, improves the feature extraction process, and increases accuracy by extracting only the region of interest. The generated masks from the proposed UNet learning model accurately represent the boundaries of the relevant structures in the images.

Finally, the generated masks serve as input to a CNN classifier to diagnose or classify TB. The CNN is trained on a large and diverse dataset of CXR images with and without TB, and is designed to accurately classify the input image as TB positive or TB negative.

Overall, this system has the potential to significantly improve the accuracy and efficiency of TB diagnosis, which is crucial for effective treatment and management of the disease. The performance of proposed system will be thoroughly evaluated on various metrics, and compared with existing methods in the literature.

A. Introduction to CXR based Diagnosis of TB

To understand the research flow, it is essential to comprehend the working mechanism of chest X-ray (CXR) and how it can be used to diagnose tuberculosis (TB). It is important to note that there are several techniques available to diagnose TB in patients, which range from invasive to non-invasive methods. Radiological diagnosis is considered a non-invasive method as it does not require any surgical tools to be inserted inside the patient's body, thereby posing a lower risk. However, CXR diagnosis is not always reliable as it requires an expert pulmonologist to interpret the results accurately. Fig. 2 to 4 illustrate how an expert pulmonologist makes a diagnosis using CXR.

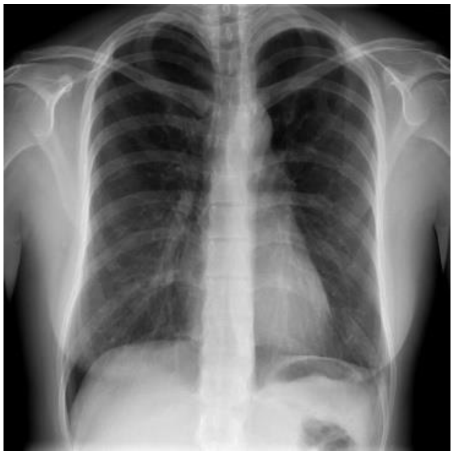


Fig. 2. Illustration of normal lungs.

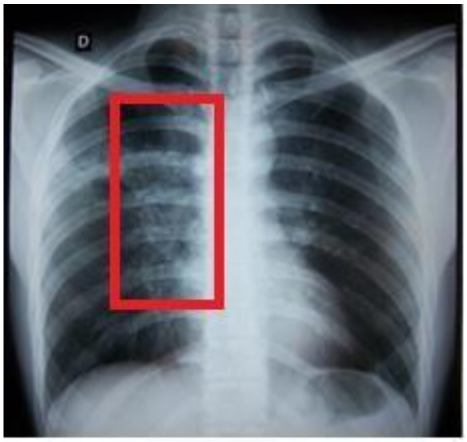


Fig. 3. Illustration of TB lungs.

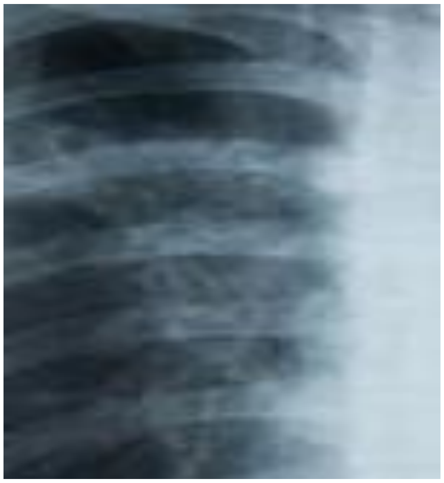


Fig. 4. Illustration of damaged lung tissues due to TB.

In CXR, an image of the chest is captured using an X-ray machine, which produces a black and white image of the chest cavity. The image contains detailed information about the structures inside the chest, including the lungs, heart, and ribs. A trained pulmonologist can use this image to identify abnormalities such as nodules, masses, and infiltrates that may indicate the presence of TB. The process of CXR diagnosis involves a thorough examination of the image by an expert pulmonologist who looks for specific patterns and abnormalities that may indicate TB.

The presence of TB in the lungs can cause damage to lung tissue, as shown in Fig. 3. However, it can be difficult for a layperson to distinguish tissue damage caused by TB from damage to the muscles of the lungs. To diagnose TB, doctors typically perform a TB skin test or a TB blood test. The TB skin test involves injecting a small amount of tuberculin under the skin and observing whether a blister forms at the injection site after 72 hours. However, this test is painful and only indicates the presence of TB bacteria on the skin. Similarly, the TB blood test only indicates the presence of TB bacteria in the blood. Neither of these tests can reveal the extent of lung damage caused by TB, which is critical for accurate diagnosis. The most definitive way to assess lung damage is through a lung biopsy, but this invasive procedure carries a risk of secondary lung infections. To avoid this risk, expert doctors rely on two non-invasive methods: listening to the sound of the lymph glands with a stethoscope while the patient breathes, or using chest X-rays to diagnose tissue damage in the lungs.

The manual process requires specialized knowledge and experience, which can make it challenging to diagnose TB accurately. Therefore, an effective computational model is developed that can accurately diagnose TB using CXR images.

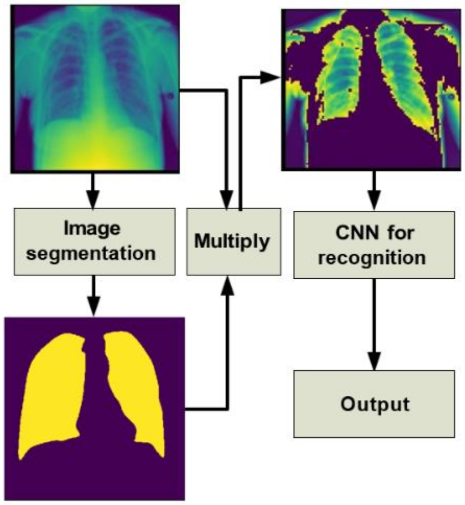


Fig. 5. Outline of the proposed computational model for TB diagnosis.

The illustration of computational model is shown in Fig. 5, which uses image processing techniques and machine learning algorithms to automate the process and improve the accuracy of TB diagnosis. The proposed computational model, as illustrated in Fig. 5, consists of three main components: a CXR image, its segmentation using a customized UNet model, and TB recognition using a CNN with the obtained mask. The target accuracy for proposed system is a minimum of 90%, because an accuracy of 90% is the bare minimum in the medical system to be acceptable to the reliability of a biopsy. Additionally, our proposed method is non-invasive, providing a desirable alternative to invasive biopsy for TB diagnosis. One of the major advantages of our model is the adoption of novel segmentation techniques using machine learning, which makes the entire process autonomous and independent of human intervention. This feature not only reduces the possibility of errors due to human intervention but also saves time and resources.

V. System Implementation

As mentioned in the previous section, the proposed system aims at identifying and classifying tuberculosis from chest radiographs (CXR). This section discusses the implementation procedure adopted in the proposed system design system. The entire discussion is carried out in systematic manner following preprocessing operation and Add on CNN based TB detection. Here Add on means applying proposed customized UNet model for precise mask generation for supervised learning.

A. Pre-processing

The first step of the research methodology is to collect a large dataset of CXR images that includes both normal and TB-infected images. The dataset considered is diverse enough to cover normal and TB infections images. In the next step, the CXR dataset is then subjected to extensive preprocessing operation to enhance the image quality and remove any artifacts. This implementation of preprocessing phase includes checking size of image and converting to grayscale, resizing, and removing any artifacts or noise, and then splits them into training and testing datasets using an 80:20 ratio. The preprocessing over input CXR images is done using a Python library OpenCV's function. For example, the signature used for resizing image to 512×512 is shown as follows:

$$\text{Resized\_CXR} = \text{cv2.resize}(\text{CXR}, (512, 512)).$$

The enhancement of the each CXR images in the dataset are carried out using adaptive histogram-based equalization approach. The implementation step is discussed as follows:

1) Let an input CXR image be denoted by $I(x,y)$, where x and y represent the spatial coordinates of the image.

2) Divide the input image $I(x,y)$ into non-overlapping tiles of size $N\ x\ N$.

3) For each tile, compute the histogram $H(i,j,k)$, where i and j represent the pixel coordinates within the tile and k represents the intensity levels ranging in 0 to $L-1$, where L is the number of gray levels.

4) Apply the Contrast Enhancement Function (CEF) on each tile, which maps the original pixel values to new enhanced values, denoted by $E(x,y)$. The CEF is numerically expressed as follows:

$$E(i,j) = \frac{fr((CDF(I(i,j))-CDFmin)}{(1-CDFmin) \times (L-1))} \quad (1)$$

where CDF is the Cumulative Distribution Function, CDFmin is the minimum CDF value in the tile, and round function rounds the values to nearest integer.

5) Replace the pixels within each tile with the enhanced values $\mathrm{E(x,y)}$.

6) Reconstruct the output image by combining the enhanced tiles.

The output image is denoted by $\mathrm{I'(x,y)}$, as enhanced version of the input CXR image $I(x,y)$. Similar operation is executed for ∀ CXR ∈ Dataset. Afterwards, dataset is split into training and testing sets with an 80:20 ratio using Python's train_test_split ( ) function from the scikit-learn library. Overall, this phase applies a basic pre-processing scheme to make CXR images suitable for segmentation process.

B. Segmentation and Mask Generation

In this phase, a customized UNet model (i.e., UNet with bi-ConvGRU model) is developed and trained to segment the lung region from the input CXR images. Once the lung region is segmented, a mask of the region of interest (ROI) is generated to focus on the segmented lungs only. The output of the proposed UNet model is a binary mask that shows the segmented lung region.

The customization in UNET is done by integrating it with bi-ConvGRU layer to each of the encoder and decoder blocks in the UNET model. Basically, the convolutional layers in the encoder and decoder are replaced with bi-ConvGRU which consist of a set of updates and reset gates that control the flow of information through the network, and a hidden state that stores the current state of the network. The bi-convGRU layers learn to selectively update and forget information based on the input image and the previous state of the network. This layers also have attention mechanisms that allow the network to focus on specific parts of the input image when making predictions. The attention mechanisms use learned weights to selectively weight the input image features at different spatial locations. The mathematical model for the U-Net architecture with bi-ConvGRU can be described as follows:

Input: An enhanced image of size $W \times H \times C$, where W is width, H is height, and C is the number of channels.

Output: A segmentation map or a mask of size $W \times H$, where each pixel represents the class of the corresponding pixel in the input image.

Encoder: The input image is passed through a series of convolutional layers with filters of increasing size. Here, the encoder takes the input X and generates a set of feature maps E such that:

$$E = \{E1, E2, E3, E4, E5\} \quad (2)$$

Each convolutional layer is followed by batch normalization and activation function (ReLU). The output feature maps are downsampled using max-pooling such that:

$$\mathrm{E_i} = \mathrm{Pool(ReLU(Conv(E\_\{i-1\})))} \quad (3)$$

The output of the ith encoder block is denoted as U_i, where i denotes the depth of the block. The output feature map U_i has a spatial size of $W_i \times H_i$ and contains C_i channels.

Decoder: The decoder upsamples the feature maps from the encoder, restoring the original image size. The ith decoder block takes the output from the ith encoder block and concatenates it with the feature map from the corresponding level of the encoder, which helps to capture more detailed information. The output of the ith decoder block is denoted as D_i, where i denotes the depth of the block. The output feature map D_i has a spatial size of $W_i \times H_i$ and contains C_i channels.

Bi-ConvGRU: The bi-ConvGRU processes the feature maps from the encoder and decoder. It enhances the network's ability to capture long-range dependencies. The bi-ConvGRU is implemented as follows:

- A convolutional layer with 1x1 kernel size is applied to the feature maps from the encoder and decoder to reduce the number of channels.
- The output feature maps are then passed through two layers of bi-directional ConvGRU, which processes the feature maps in both forward and backward directions.
- The output of the bi-ConvGRU is then upsampled to the original image size.

The output of the bi-ConvGRU is denoted as G and has a spatial size of W x H and a single channel.

Final layer: The output of the bi-ConvGRU is passed through a convolutional layer with 1x1 kernel size to obtain the final segmentation map. Fig. 6 shows schematic architecture of the implemented customized UNet model.

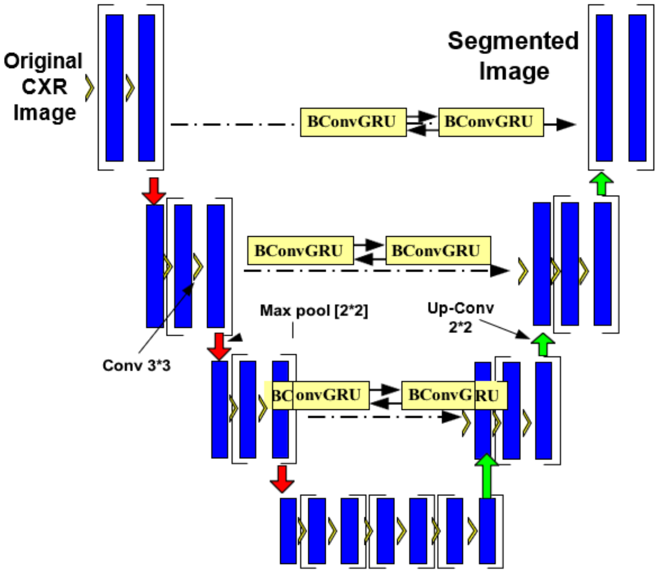


Fig. 6. Proposed customized U-Net model for segmentation and mask generation.

For ∀ pixel at location (i,j) in the feature map at layer l, the encoding process applies a convolution operation to the corresponding receptive field in the previous feature map at layer $l-1$, adds a bias term, and applies an activation function to obtain the output feature map at layer l. This process is repeated for all pixels in the feature map to obtain the full output feature map at layer l, expressed as follows:

$$h_{i,j}^{l} = f(W_l \times h_{i,j}^{l-1} + b_l) \quad (4)$$

where:

- $h_{i,j}^{l-1}$ represents the feature map at layer $l-1$ of the previous convolutional layer,
- $h_{i,j}^{l}$ represents the feature map at layer l,
- W_l is the weight matrix for the convolutional operation at layer l,
- b_l is the bias vector at layer l, and
- $f(\,)$ is the activation function applied element-wise to the output of the convolution operation.

Specifically, the customized UNet is used to learn the mapping between the input image and the output segmentation masks. The learned model parameters can then be used to minimize the energy function, which consists of a data fitting term and a regularizing term, to generate the optimized segmentation. Eq. (5) represents the image segmentation function with the addition of transform domain analysis to enhance its performance. It includes a term that measures the curvature of the segmentation contours in the transform domain, which helps to improve the smoothness of the segmentation.

$$argmin_{u,}|k| + \mu \int_{c} |\Delta u|^{2}\, dx + \int (u-f)^{2}\, dx \quad (5)$$

In Eq. (5), the variable f represents the original image pixel values, u represents the average pixel strength in a segment, Δu represents the variance of pixel intensities in a segment, and k represents a constant that is optimized by Bi-Conv-GRU.

C. CNN based TB Diagnosis

In the previous section, the image segmentation is performed using a customized UNet model, which generates a binary mask of the regions of interest (ROI) in the input image. The trained UNet model is then used to generate masks from other CXR datasets by processing the input CXR images through the trained UNet model. The output of the UNet model is a binary mask that shows the segmented lung region. The generated mask is then used to train a CNN model for TB detection. The input to the CNN model is the generated mask, and the output is the binary classification of TB or non-TB. The CNN model is trained using the preprocessed training dataset, and its performance is evaluated using the preprocessed validation dataset. The TB diagnosis model involved implementation of the CNN architecture which includes convolutional layers, pooling layers, and fully connected layers with activation functions such as ReLU and sigmoid. The loss function used could be binary cross-entropy, and the optimization algorithm is used as Adam optimizer. Mathematically, this process can be represented as follows:

Let I be the input CXR image, and M be the binary mask generated by the UNet model. The trained UNet model can be represented as a function F, which takes the input image I and generates the binary mask M such that:

$$M = F(I) \quad (6)$$

The binary mask M is then fed into a CNN model, which takes it as input and generates a binary classification output, indicating whether the image contains TB or not. The CNN model can be represented as a function G, which takes the binary mask M as input and generates the binary classification output y:

$$y = G(M) \tag{7}$$

Thus, the entire process of image segmentation and mask generation using UNet, and TB detection using a CNN can be represented as a composite function H, which is the composition of F and G, given as follows:

$$H(I) = G(F(I)) \tag{8}$$

To train the CNN model, we use the binary mask M generated by the UNet model as the input, and its corresponding label (0 or 1, indicating whether the image contains TB or not) as the output. The CNN model is trained using a binary cross-entropy loss function, given as follows:

$$L(y_{true}, y_{pred}) = -\alpha + (\beta \times \gamma) \tag{9}$$

where:

$$\alpha = (y_{\text{true}} \times \log(y_{\text{pred}})$$

$$\beta = (1 - \text{y\_true})$$

$$\gamma = \log(1 - \text{y\_pred})$$

In Eq. (9), the variable y_{true} is the true label, y_{pred} is the predicted output from the CNN model, and log represents the natural logarithm. The training process involves minimizing the loss function over a set of training images and their corresponding labels, using an optimizer, Adam. The negative sign in the beginning of the right-hand side of the equation is used to indicate that we want to minimize the loss function. During training, the objective is to minimize the difference between the true labels and the predicted labels. By adding a negative sign to the equation, we can use optimization algorithms that are designed to minimize a function, rather than maximize it. In other words, by minimizing the negative of the log likelihood loss, we are maximizing the likelihood of the predicted labels given the true labels. Once the CNN model is trained, it can be used to classify new CXR images as either TB positive or negative, by first generating the binary mask using the UNet model and then passing it through the CNN model. The proposed learning model can be used as a tool for early diagnosis of damage caused by TB in the lungs. To be more specific, this device will help in the detection of tuberculous pneumonia or tuberculous pneumothorax.

VI. Experimental Analysis

The design and development of the proposed system is done using Python executed in Anaconda distribution. The proposed work encompasses two main contributions Viz. i) development of novel customized UNet learning model which is trained to generate binary mask as segmented lung region from the given chest X-ray image; ii) implementing and training CNN with generated mask to diagnose Tuberculosis from the given mask generated. Both models are integrated and their response are synchronized to carry out detection of Tuberculosis. This section presents the outcome and performance analysis of the proposed learning models.

A. Performance Analysis of Segmentaion Model

The dataset used to train the proposed customized UNet model for image segmentation is highlighted in Table I.

TABLE I. Dataset Used for Training UNet for Image Segmentation

Dataset	Total images	Training images	Testing ratio
Lung image segmentation [31]	704	563	141

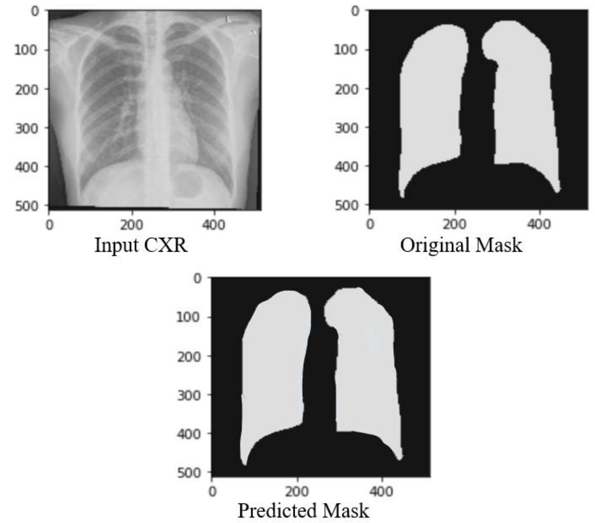
As shown in Table I, the dataset obtained from Kaggle which consists of total 704 chest X-rays with corresponding masks. The dataset is further split in training and testing set with a ratio of 80:20. The parameters used to train the models is also highlighted in Table II.

TABLE II. Training Parameters Used for Image Segmentation

No. Epochs	Batch size	Loss function	Optimizer
50	32	Binary crossentropy	Adam

The training of the proposed augmented UNet model for generating mask in form of segmented image is carried out with total 563 images. The model is trained for 50 epochs, followed by batch size equals 32. In addition, binary cross entropy and Adam is used as loss function and optimized to calibrate learning of the model. The trained UNet model is further applied to different chest-Xray to generate mask.

The interpretation of the Fig. 7 reveals that the predicted mask is mostly the same as the original mask with only minor differences, it indicates that the UNET-based segmentation model is performing well on the given CXR image. The smooth edge border in the predicted mask could be due to the model's ability to capture fine details and edges in the image. A good performance is achieved by the customized based segmentation model indicating that it is able to accurately generate mask with lung areas. This would be useful in a TB detection system, as it can provide a good scope for an accurate localization of the disease for further analysis and diagnosis.



Input CXR

Original Mask

Predicted Mask

Fig. 7. Depiction of mask predicted by the proposed customized U-Net model with input CXR and original mask.

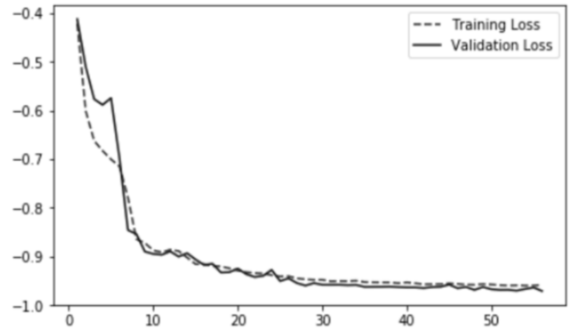


Fig. 8. Model training and validation loss curve analysis.

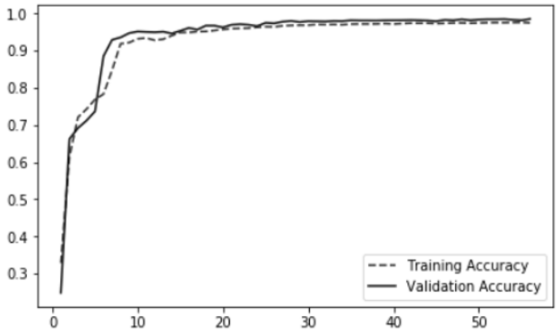


Fig. 9. Model training and validation accuracy curve analysis.

The training and validation loss curve in Fig. 8 shows the evolution of the loss (or cost) function over time as the model is trained on the training data. It can be seen from the graph trend that the loss decreases over time or epochs as the model learns to make better predictions. Similarly, the graph trend of training and validation accuracy curve shown in Fig. 9 accuracy curves follow a stable and linear trend towards 0.9696 accuracy on 50 epochs suggests that the model has converged and is performing well on the given task. Although, there is initial fluctuations in the training and validation accuracy curves which are common and can be attributed to the random initialization of weights in the model and the stochastic nature of the optimization algorithm. As the training progresses, the model learns more features from the data and the accuracy of the model on both the training and validation data gets improved. This also indicates that the model is not overfitting to the training data and generalizes well to new, unseen data.

B. *Performance Analysis of CNN based TB Detection*

The dataset used for generating mask using trained UNet model and training CNN model for detecting TB is highlighted in Table III.

A synthetic dataset is used in the phase of disease (TB) detection. Many chest X-rays images including normal and Tuberculosis are collected from different sources to build a dataset. The dataset consists of 7000 images, where 3500 images are belongs to normal CXR and reaming 3500 CXR images belongs to TB. The images are processed with the trained UNet model to generate a mask for all the input images towards performing efficient supervised learning for disease detection. After generating mask, dataset is split into training and testing set with ratio of 80:20. Table IV shows the training parameters used for training the CNN for TB detection.

The classification model is trained over 5600 masks for 50 epoch, and batch size considered equals to 32. Binary cross entropy loss function and Adam optimizer is used to improve the learning of the CNN learning model. The performance of trained CNN is assessed with testing dataset. The analysis is conducted using confusion matrix and classification performance indicators such as accuracy, precision, recall rate and F1-score. The accuracy is a general metric that measures the overall performance of the model, while precision and recall are more specific metrics that measure the model's ability to correctly classify positive samples. The F1-score is a balanced metric that considers both precision and recall. The confusion matrix for TB detection model is shown in Figure 10.

The confusion matrix shown in Fig. 10 reveals that the trained CNN model using the generated masks from the proposed customized UNet model has performed well in detecting both normal and TB images. The confusion matrix shows that out of 700 normal images, 693 have been correctly predicted as normal, which gives a precision of 1.00 and a recall of 0.99. Similarly, out of 700 TB images, all have been correctly predicted as TB, giving a precision of 0.99, recall and F1-score of 1.00 as shown in Table V.

TABLE III. DATASET USED FOR GENERATING MASK AND TRAINING CNN FOR TB DETECTION

Dataset	Total images	Training images	Testing ratio
TB lungs classificationdataset	7000	5600	1400

TABLE IV. TRAINING PARAMETERS FOR TB DETECTION USING CNN

No. Epochs	Batch size	Loss function	Optimizer
50	32	Binary crossentropy	Adam

TABLE V. QUANTIFIED OUTCOMES FOR TB DETECTION

	Precision	Recall	F1-score	Support
Normal Lung	0.99	0.99	0.99	700
TB Lung	1.00	1.00	1.00	700
micro avg	0.99	0.99	0.99	1400
macro avg	1.00	0.99	0.99	1400
weighted avg	1.00	0.99	0.99	1400

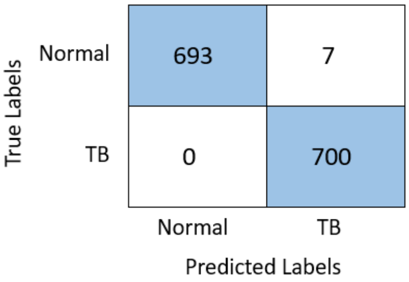


Fig. 10. Confusion plot for assessing performance of trained CNN for TB detection.

The quantified values shown in Table V represent the performance metrics of a binary classification model for the classes "Normal" and "TB". The precision for "Normal CXR" is 1.00, meaning that all the samples predicted as "Normal" were correctly classified, and for "TB" it is 0.99, indicating that out of all samples predicted as "TB", 99% were actually "TB". The recall for "Normal" is 0.99, meaning that out of all the actual "Normal" samples, the model correctly identified 99%, and for "TB" it is 1.00, indicating that the model correctly identified all the actual "TB" samples.

The micro avg. represents the performance metrics computed globally by counting the total true positives, false negatives, and false positives, and the macro avg represents the average of the performance metrics calculated for each class. The weighted avg. takes the weighted average of the performance metrics, where the weights are the number of samples in each class. Overall, the model has high precision, recall, and F1-score for both classes, indicating a good performance in the binary classification task.

C. Extensive Analysis with Different Versions of CNN

This section presents an extensive analysis of the proposed work by evaluation proposed system with different version of existing CNN models. Moreover, entire analysis is carried in two scenarios viz. i) performance analysis with proposed segmentation scheme using customized UNet and ii) performance analysis without segmentation (i.e., models are trained directly on the input images with any pre-processing and segmentation).

For an effective analysis, the proposed system is subjected to a comparative analysis. However, this is a different form of comparative analysis, where the outcome of proposed system (performance parameters with segmentation) is further subjected to 7 version of neural network viz. Resnet 18 [32], Resnet 50 [33], Resnet101 [34], VGG19 [35], Densenet201 [36], Squeezenet [37], and Mobilenet [38].

The prime justification behind this mode of comparative analysis is to assess whether the performance of proposed classification model with segmentation using augmented U-Net model can be further improved upon using above mentioned seven variants of CNN model with different layers, The idea is to perform an assessment with respect to standard performance parameters of Accuracy, Precision, Sensitivity, F1-Score, and Specificity using dual combination of with proposed system (i.e., with segmentation) and without proposed system (i.e., without segmentation).

As shown in Fig. 11, the accuracy of learning models with proposed segmentation scheme ranges between 99-100, whereas that of without segmentation ranges between 95-97 for different types of CNNs architecture or model. Nearly similar trend is also observed for precision (Fig. 12), sensitivity (Fig. 13), F1 score (Fig. 14) and specificity (Fig. 15).

The F1 score with segmentation is found in range of 97-98 whereas that of without segmentation is found between 95 and 96 mainly. Out of different variants of CNN, the outcome is found to be better for Squeezenet and Mobilenet for accuracy followed by Resnet 18 in Fig. 11. The outcome of precision from Fig. 12 showcases Resnet50, Resnet101, and Squeezenet to be performing well compared to other variants of CNN model. The outcome of sensitivity from Fig. 13 highlights better performance of Densenet201 followed by Mobilenet and Resnet101. Similar performance trend can be also seen with respect to higher F1 score from Fig. 14 compared to others versions of CNN. The performance of Resnet50, Resnet101, Densenet201, and Squeezenet offers nearly similar performance for specificity shown in Fig. 15.

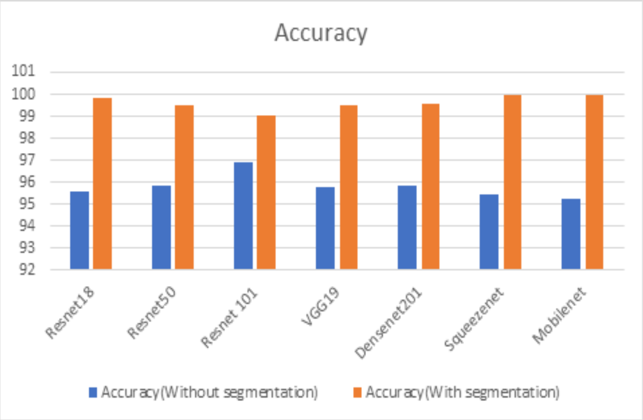


Fig. 11. Comparative analysis of accuracy (%).

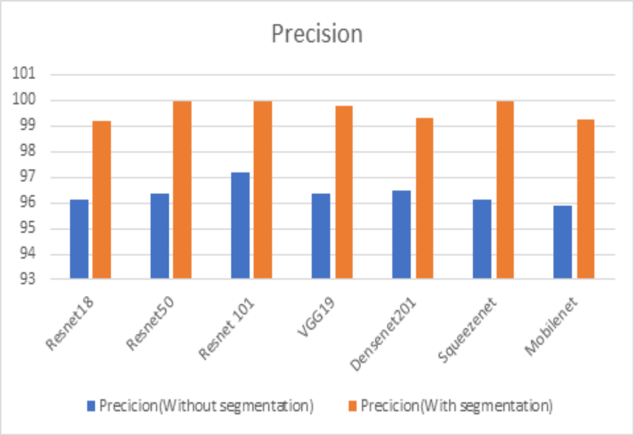


Fig. 12. Comparative analysis of precision (%).

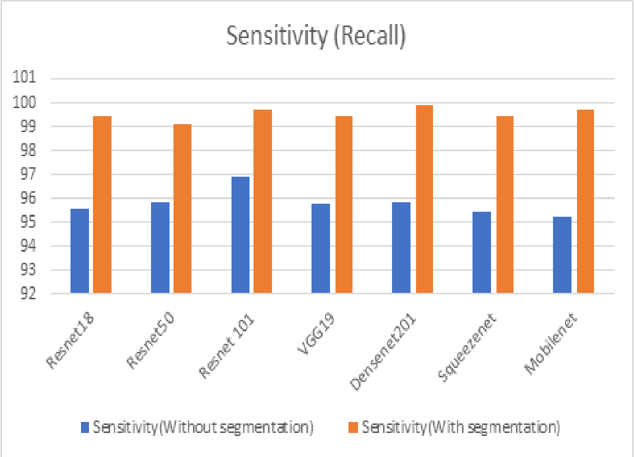


Fig. 13. Comparative analysis of sensitivity (%).

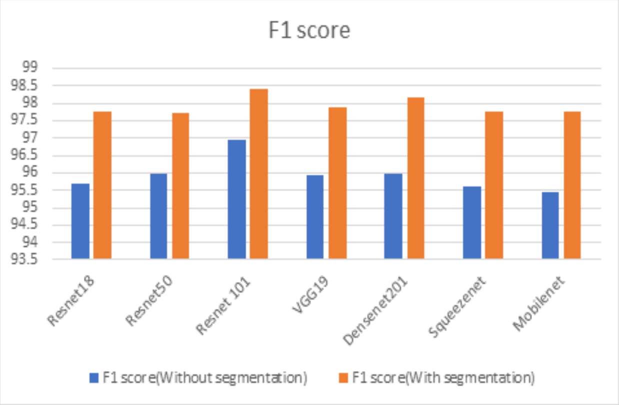


Fig. 14. Comparative analysis of f1 score (%).

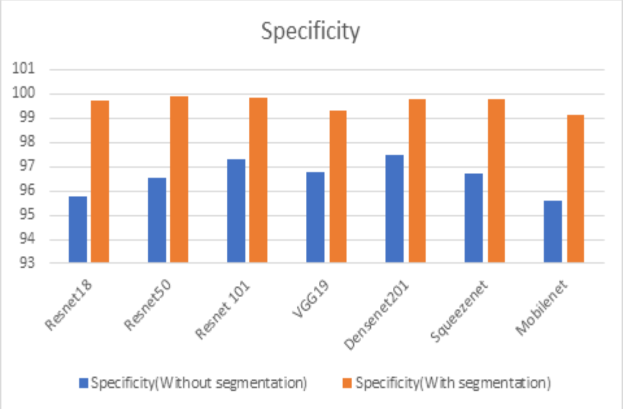


Fig. 15. Comparative analysis of specificity (%).

Based on the analysis and performance statistics, it can be concluded that the proposed scheme has provided a better performance with strong potential in TB diagnosis. The customized UNet model is able to accurately capture the important features in the input CXR images and generate precise masks, which are then used to train the CNN model. This results in a more robust and accurate CNN model, which is able to accurately classify the CXR images as normal or TB. Therefore, the customized UNet model acts as an important pre-processing step, which enhances the performance of the CNN model. Overall, the combination of exploratory data analysis, image segmentation, and appropriate training of CNN provided a more accurate and efficient approach to TB diagnosis, which is crucial for effective treatment and management of the disease.

VII. Conclusion

In this paper, the research work has suggested a novel automated disease diagnosis scheme based on the Add-on CNN learning model which benefited using augmented U-Net based segmentation algorithm. The implementation of the proposed system follows a systematic procedure which included exploratory data analysis, image segmentation, and CNN to accurately diagnose TB. The proposed system effectively preprocesses the input data, locates the lungs, and extracts the region of interest from the CXR images, using customized UNet model which is augmented with Bi-Conv-GRU layers to carry out precise and optimized form of segmentation. This procedure greatly improves the accuracy and efficiency of the subsequent classification process. The generated masks from the proposed UNet Model accurately represent the boundaries of the relevant structures in the images, which improves the feature extraction process and increases the accuracy of the CNN classifier. Our evaluation on various metrics with extensive analysis demonstrates the effectiveness and potential of the proposed system in TB diagnosis. In the future, we plan to further improve the proposed system using self-exploration like reinforcement learning and apply it to larger datasets and different types of medical images. Also, the future study focuses on security aspect of medical imaging in telemedicine application.

References

[1] Y. Shah, Molecular Epidemiology Study of Mycobacterium Tuberculosis Complex, IntechOpen, ISBN: 9781839680991, 1839680997, 2021.

[2] B.R. Bloom, K.K. Holmes, P. Jha, S. Bertozzi, Major Infectious Diseases, ISBN: 9781464805240, 1464805245, 2017.

[3] World Health Organization, Global Tuberculosis Report 2019, ISBN: 9789241565714, 9241565713, 2019.

[4] G.B. Migliori, M. C. Raviglione, Essential Tuberculosis, ISBN: 9783030667030, 3030667030, 2021.

[5] S. Bhalla, S. Martínez-Jiménez, Chest Imaging, Oxford University Press, ISBN: 9780199858064, 0199858063, 2019.

[6] K.C. Santosh, N. Das, S. Ghosh, Deep Learning Models for Medical Imaging, ISBN: 9780128236505, 0128236507, Elsevier Science, 2021.

[7] J. Latif, C. Xiao, S. Tu, S. U. Rehman, A. Imran and A. Bilal, "Implementation and Use of Disease Diagnosis Systems for Electronic Medical Records Based on Machine Learning: A Complete Review," in IEEE Access, vol. 8, pp. 150489-150513, 2020, doi: 10.1109/ACCESS.2020.3016782.

[8] M. R. Ahmed, Y. Zhang, Z. Feng, B. Lo, O. T. Inan and H. Liao, "Neuroimaging and Machine Learning for Dementia Diagnosis: Recent Advancements and Future Prospects," in IEEE Reviews in Biomedical Engineering, vol. 12, pp. 19-33, 2019, doi: 10.1109/RBME.2018.2886237.

[9] G. Joo, Y. Song, H. Im and J. Park, "Clinical Implication of Machine Learning in Predicting the Occurrence of Cardiovascular Disease Using Big Data (Nationwide Cohort Data in Korea)," in IEEE Access, vol. 8, pp. 157643-157653, 2020, doi: 10.1109/ACCESS.2020.3015757.

[10] G. Joo, Y. Song, H. Im and J. Park, "Clinical Implication of Machine Learning in Predicting the Occurrence of Cardiovascular Disease Using Big Data (Nationwide Cohort Data in Korea)," in IEEE Access, vol. 8, pp. 157643-157653, 2020, doi: 10.1109/ACCESS.2020.3015757.

[11] Z. Lin et al., "AANet: Adaptive Attention Network for COVID-19 Detection From Chest X-Ray Images," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 4781-4792, Nov. 2021, doi: 10.1109/TNNLS.2021.3114747.

[12] J. -X. Wu, P. -Y. Chen, C. -M. Li, Y. -C. Kuo, N. -S. Pai and C. -H. Lin, "Multilayer Fractional-Order Machine Vision Classifier for Rapid Typical Lung Diseases Screening on Digital Chest X-Ray Images," in IEEE Access, vol. 8, pp. 105886-105902, 2020, doi: 10.1109/ACCESS.2020.3000186.

[13] M. Eslami, S. Tabarestani, S. Albarqouni, E. Adeli, N. Navab and M. Adjouadi, "Image-to-Images Translation for Multitask Organ Segmentation and Bone Suppression in Chest X-Ray Radiography," in IEEE Transactions on Medical Imaging, vol. 39, no. 7, pp. 2553-2565, July 2020, doi: 10.1109/TMI.2020.2974159.

[14] J. An, Q. Cai, Z. Qu and Z. Gao, "COVID-19 Screening in Chest X-Ray Images Using Lung Region Priors," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 11, pp. 4119-4127, Nov. 2021, doi: 10.1109/JBHI.2021.3104629.

[15] J. D. Arias-Londoño, J. A. Gómez-García, L. Moro-Velázquez and J. I. Godino-Llorente, "Artificial Intelligence Applied to Chest X-Ray

Images for the Automatic Detection of COVID-19. A Thoughtful Evaluation Approach," in IEEE Access, vol. 8, pp. 226811-226827, 2020, doi: 10.1109/ACCESS.2020.3044858.

[16] C. Li, D. Zhang, S. Du and Z. Tian, "Deformation and Refined Features Based Lesion Detection on Chest X-Ray," in IEEE Access, vol. 8, pp. 14675-14689, 2020, doi: 10.1109/ACCESS.2020.2963926.

[17] K. Wang, X. Zhang, S. Huang, F. Chen, X. Zhang and L. Huangfu, "Learning to Recognize Thoracic Disease in Chest X-Rays With Knowledge-Guided Deep Zoom Neural Networks," in IEEE Access, vol. 8, pp. 159790-159805, 2020, doi: 10.1109/ACCESS.2020.3020579.

[18] T. Rahman et al., "Reliable Tuberculosis Detection Using Chest X-Ray With Deep Learning, Segmentation and Visualization," in IEEE Access, vol. 8, pp. 191586-191601, 2020, doi: 10.1109/ACCESS.2020.3031384.

[19] F. Munawar, S. Azmat, T. Iqbal, C. Grönlund and H. Ali, "Segmentation of Lungs in Chest X-Ray Image Using Generative Adversarial Networks," in IEEE Access, vol. 8, pp. 153535-153545, 2020, doi: 10.1109/ACCESS.2020.3017915.

[20] C. -M. Kim, E. J. Hong and R. C. Park, "Chest X-Ray Outlier Detection Model Using Dimension Reduction and Edge Detection," in IEEE Access, vol. 9, pp. 86096-86106, 2021, doi: 10.1109/ACCESS.2021.3086103.

[21] N. Paluru et al., "Anam-Net: Anamorphic Depth Embedding-Based Lightweight CNN for Segmentation of Anomalies in COVID-19 Chest CT Images," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 3, pp. 932-946, March 2021, doi: 10.1109/TNNLS.2021.3054746.

[22] H. Salehinejad, E. Colak, T. Dowdell, J. Barfett and S. Valaee, "Synthesizing Chest X-Ray Pathology for Training Deep Convolutional Neural Networks," in IEEE Transactions on Medical Imaging, vol. 38, no. 5, pp. 1197-1206, May 2019, doi: 10.1109/TMI.2018.2881415.

[23] O. D. T. Catalá et al., "Bias Analysis on Public X-Ray Image Datasets of Pneumonia and COVID-19 Patients," in IEEE Access, vol. 9, pp. 42370-42383, 2021, doi: 10.1109/ACCESS.2021.3065456.

[24] P. Zhang, Y. Zhong, Y. Deng, X. Tang and X. Li, "Drr4covid: Learning Automated COVID-19 Infection Segmentation From Digitally Reconstructed Radiographs," in IEEE Access, vol. 8, pp. 207736-207757, 2020, doi: 10.1109/ACCESS.2020.3038279.

[25] C. Zhou, J. Song, S. Zhou, Z. Zhang and J. Xing, "COVID-19 Detection Based on Image Regrouping and Resnet-SVM Using Chest X-Ray Images," in IEEE Access, vol. 9, pp. 81902-81912, 2021, doi: 10.1109/ACCESS.2021.3086229.

[26] S. Z. Y. Zaidi, M. U. Akram, A. Jameel and N. S. Alghamdi, "Lung Segmentation-Based Pulmonary Disease Classification Using Deep Neural Networks," in IEEE Access, vol. 9, pp. 125202-125214, 2021, doi: 10.1109/ACCESS.2021.3110904.

[27] Y. -H. Wu et al., "JCS: An Explainable COVID-19 Diagnosis System by Joint Classification and Segmentation," in IEEE Transactions on Image Processing, vol. 30, pp. 3113-3126, 2021, doi: 10.1109/TIP.2021. 3058783.

[28] F. Yan, X. Huang, Y. Yao, M. Lu and M. Li, "Combining LSTM and DenseNet for Automatic Annotation and Classification of Chest X-Ray Images," in IEEE Access, vol. 7, pp. 74181-74189, 2019, doi: 10.1109/ACCESS.2019.2920397.

[29] D. -P. Fan et al., "Inf-Net: Automatic COVID-19 Lung Infection Segmentation From CT Images," in IEEE Transactions on Medical Imaging, vol. 39, no. 8, pp. 2626-2637, Aug. 2020, doi: 10.1109/TMI.2020.2996645.

[30] J. Lian et al., "A Structure-Aware Relation Network for Thoracic Diseases Detection and Segmentation," in IEEE Transactions on Medical Imaging, vol. 40, no. 8, pp. 2042-2052, Aug. 2021, doi: 10.1109/TMI.2021.3070847.

[31] Kaggle: Your Home for Data Science. (n.d.). https://www.kaggle.com/ nikhilpandey360/lung-segmentation-from-+chest-x-ray-dataset.

[32] X. Ou et al., "Moving Object Detection Method via ResNet-18 With Encoder–Decoder Structure in Complex Scenes," in IEEE Access, vol. 7, pp. 108152-108160, 2019, doi: 10.1109/ACCESS.2019.2931922.

[33] X. Yu, C. Kang, D. S. Guttery, S. Kadry, Y. Chen and Y. -D. Zhang, "ResNet-SCDA-50 for Breast Abnormality Classification," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 1, pp. 94-102, 1 Jan.-Feb. 2021, doi: 10.1109/TCBB.2020.2986544.

[34] Zhang, Q. A novel ResNet101 model based on dense dilated convolution for image classification. SN Appl. Sci. 4, 9 (2022). https://doi.org/10.1007/s42452-021-04897-7.

[35] A. F. Ibrahim, S. P. Ristiawanto, C. Setianingsih and B. Irawan, "Micro-Expression Recognition Using VGG19 Convolutional Neural Network Architecture And Random Forest," 2021 4th International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), 2021, pp. 150- 156, doi: 10.1109/ISAMSR53229.2021.9567872.

[36] K. Adam, I. I. Mohamed and Y. Ibrahim, "A Selective Mitigation Technique of Soft Errors for DNN Models Used in Healthcare Applications: DenseNet201 Case Study," in IEEE Access, vol. 9, pp. 65803-65823, 2021, doi: 10.1109/ACCESS.2021.3076716.

[37] B. Qiang et al., "SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition," in IEEE Access, vol. 9, pp. 77661-77674, 2021, doi: 10.1109/ACCESS.2021.3079337.

[38] K. Kadam, S. Ahirrao, K. Kotecha and S. Sahu, "Detection and Localization of Multiple Image Splicing Using MobileNet V1," in IEEE Access, vol. 9, pp. 162499-162519, 2021, doi: 10.1109/ACCESS.2021.3130342.

Brightness and Contrast Enhancement Method for Color Images via Pairing Adaptive Gamma Correction and Histogram Equalization

Bilal Bataineh
Information Systems Department-College of Computers and Information Systems,
Umm Al-Qura University, Mecca 24382, Saudi Arabia

***Abstract*—For enhanced adaptability to poor light enhancement whilst achieving high image contrast, a new method for color image correction based on the advantages of non-linear function in grey transformation and histogram equalization techniques is proposed in this work. Firstly, the original red, green and blue (RGB) image is converted into the HSV color space, and the V channel is used for enhancement. An adaptive gamma generator is proposed to adaptively calculate gamma parameters in accordance with dark, medium, or bright image conditions. The computed gamma parameters are used to propose a cumulative distribution function that produces an optimized curve for illumination values. Next, a second modified equalization is performed to evenly correct the offset of the illumination curve values on the basis of the equal probability of the available values only. Finally, the processed V channel replaces the original V channel, and the new HSV model returns to the RGB color space. Experiments show that the proposed method can significantly improve the low contrast and poor illumination of the color image whilst preserving the color and details of the original image. Results from benchmark data sets and measurements indicate that the proposed method outperforms other state-of-the-art methods.**

Keywords—Color image; gamma correction; histogram equalization; image contrast; image enhancement

I. INTRODUCTION

These days, digital images are used widely because of the rapid development in capturing machines and computer vision technology. Usually, we deal with low-quality images with low contrast and poor illumination caused by various capturing conditions [1]–[3]. Low-quality visual images pose challenges to human perception, as well as image processing and computer vision applications [2], [4], [5]. Therefore, the quality of images should be improved before they are used to make images highly appropriate to the human visual perception and be analyzed easily by machines and improve computer vision applications [1]–[3], [6], [7].

Although no standards exist for correct levels of illumination and contrast, any images captured in condition lighting that does not correspond to normal levels of light are often poor images [2], [8], [9]. In accordance with reviews in previous studies, low contrast and poor illumination are classic challenges in image processing, computer vision and other applications, such as medical technologies, military applications, satellite imaging, video monitoring, traffic technologies, industrial production and underwater image enhancement [2]–[4], [10], [11].

In reality, color images are commonly used, and research is concerned about improving the quality of images [4], [10], [12]. In general, the visual quality of images is enhanced by improving their illumination and contrast. The main enhancement techniques are histogram equalization (HE) [3], grey transformation (GT) [13], defogging model [14], image fusion [8], Retinex theory [15], frequency-domain [11] and machine learning [16].

HE methods remap the histogram of pixel values to corresponding new values to improve the poor contrast of images [2], [3]. They are widely used for their simplicity, swiftness, and effectiveness. However, the output quality varies with the characteristics of the input image, which may cause uneven brightness, lost details and color degradation. GT methods calculate new values of pixels by using linear/non-linear mathematical functions [2], [17]. They are simple, but they could require manual input parameters. This leads to unsatisfactory results, and some details may be lost in the enhanced images.

Image defogging methods remove atmospheric dispersion, such as fog, from images to restore the contrast [2], [5]. Defogging methods show good performance. Nevertheless, they have complex computation, require prior knowledge of the scene and show the over-enhancement problem. Image fusion methods use multiple images of the same scene under different conditions to extract information for improving the image quality [2], [8]. They are difficult to use in many applications, and the process time is long.

Retinex theory-based methods deal with illumination conditions that vary spatially with color and intensity [2], [18], [19]. Retinex methods are simple and enhance contrast, brightness, and colors. However, they may damage edges and produce over-enhancement and uneven brightness levels. Frequency-domain methods use a Fourier transform of an image to be multiplied by a high-frequency modulation filter and then perform inverse transformation to produce an enhanced image [2], [20], [21]. Such methods can improve image details and noise but require complex calculations and manual input parameters. Recently, machine learning methods have been a trend in image enhancement for their excellent performance [22]–[27]. Nonetheless, they require large data sets for training, benchmark data sets on poor contrast and illumination are still lacking and the complexity of the models increases the cost in time and hardware required.

Almost all the above-mentioned techniques still have low contrast within weak illumination images, as well as missing detail in enhanced images. The available methods also suffer from various drawbacks. However, GT and HE methods are mostly used for their simplicity and fewer disadvantages.

The objective of this paper is to propose a novel global enhancement method to overcome the disadvantages of previous methods by combining the advantages of HE and GT methodologies in one method. The proposed method is simple and fast, and it offers excellent performance to improve low contrast and poor illumination in color images. It does not cause color distortion and increases information preservation as much as possible compared with existing methods. A set of experiments is conducted using many measurements on several benchmark data set images. The results show that the proposed method significantly improves the contrast and illumination, as well as preserves the colors and details of the benchmark data sets better than existing methods.

The rest of this paper is organized as follows. In Section II, the state of the art is presented. Section III describes the proposed method in detail. Section IV shows the experiments, results and discussion. Section V presents the conclusion.

II. Related Work

In accordance with literature reviews provided over the years, several methods for improving images' contrast and illumination have been proposed. Previous survey studies, such [2]–[4], [6], [7], [10], [11], [17], [20], [28]–[31], reported that HE and GT are the most involved techniques in proposed contrast and illumination enhancement methods given their simplicity, fast processing time and high performance.

Many GT methods use linear functions to calculate current values of pixels. However, bright regions may become saturated after values are reset, causing lost details in processed images. These methods also require manual intervention and some experience from the user to find optimal enhancement [32]–[35]. To avoid this problem, GT methods then adopt non-linear functions to perform the enhancement. For instance, authors of [1] proposed an enhancement method for color images of the retina. This method improves the detail by using a bilateral filter. Next, non-linear gamma and logarithmic functions are used to correct the illumination. Finally, the corrected grey values replace the V channel in the HSV color space. This method has achieved great performance, but it shows over-enhancement in some cases. In [8], an enhancement method for low-light images based on a pair of complementary gamma functions through image fusion is introduced. Experiments show that this method can enhance the detail and improve the contrast of low-light images. Nevertheless, it exhibits over-enhancement and loses details in some cases.

In [9], a method based on Retinex theory for improving image illumination is proposed. This method uses a recursive filter to estimate layers and a 2D visual gamma transform function to remap pixels. The results show that this method improves visibility in local areas, but it loses detail in images. Authors of [13] recommended a non-linear transformation with a bat optimization algorithm. The bat algorithm is employed to solve the problem of manual selection of control parameters. It is used for automated selection of control parameters in non-linear transformation, but this method presents slow time processing.

In [36], a method to enhance low-light color images based on the fusion strategy is proposed, Gaussian function, and non-linear function parameters. This method improves brightness and contrast. However, it is slow, so it cannot be used to improve video images. Authors of [37] introduced a method that uses a grey-value histogram to adaptively obtain a non-linear function to enhance dark regions whilst preventing bright regions. Nonetheless, the performance is strongly limited. Authors of [38] proposed a contrast and brightness enhancement method based on adaptive gamma correction non-linear function to enhance dimmed regions. This method can enhance low-brightness images and is only effective for such images.

Previous GT methods mainly used logarithmic or gamma as transformation functions, and GT was adopted widely as a step in other image enhancement methods, especially in low-light images. The use of non-linear functions may require complex calculations, in addition to the traditional problems of over-enhancement and unsatisfactory results.

HE is another common technique to improve illumination and contrast and prevent saturation. It has become a popular technique for its high ability to enhance contrast, and several improvements, such as improving the brightness, have been proposed to overcome its limitations. The basic HE was proposed in [39]; this method used a PDF for different CDFs to calculate the new distributions of values. In [40], the output by applying preprocessing steps to the original is improved. However, the method produces unwanted side effects in a wide type of images.

To reduce the side effects of using grey values of the entire image to estimate the optimal enhancement for local different areas in the image, many methods have adopted the local approach, which divides the histogram or image into independent parts to be enhanced individually [41]–[42]. In general, the results of these methods were better than those of the global approach. Nevertheless, these methods are complex and slow, as well as produce noise and over-enhancement challenges.

The mean and variance-based sub-image HE method [43] was proposed to enhance the contrast and preserve the illumination of images. This method exhibits high performance in many types of images. However, it may lose some details from images with a narrow range of grey values. Authors of [42] suggested an HE method for color images. In this method, a clipped histogram based on the HSV color space is separated into two parts by a thresholding method, then the parts are equalized independently. The result of this method has many problems.

Some methods have adopted different techniques instead of the usual equalization to find corresponding new pixel values. Authors of [44] used the min, max, mean and median of a histogram to clip extreme values. Then, several methods were applied to remap new histograms. Authors of [21] utilized a set of filters and the binary tree structure to remap histogram values. In [45], a region-based thresholding method and the entropy of grey level is adopted to reshape histograms. These methods are complex to apply and depend on the accuracy of the set of involved methods. In addition, results are unsatisfactory in many cases. Previous HE methods could effectively enhance light, improve contrast, and increase the visibility of image details. Nonetheless, HE methods may produce noise, lose information, and destroy color fidelity.

In sum, GT mainly deals with illumination issues but does not improve contrast. It may produce problems on contrast in many cases. Meanwhile, HE methods mainly enhance the contrast between pixel values in images, but they do not improve the illumination perfectly or even produce a potential challenge in low illumination levels.

To utilize both advantages of HE and GT, some proposed algorithms combine HE and GT as steps in one method. For example, authors of [12] proposed a method for improving image contrast, adaptive gamma correction with weighted histogram distribution are used, and then the gamma function in the illumination channel of the HSI color space. This method improves contrast and preserves the color of images. However, noise is created for the outputs. In [46], a method for improving the contrast of images via HE and the gamma function is introduced. This method improves images, but the results are unsatisfactory. In [47], a method to improve dark images is proposed. This method uses both the non-linear gamma function and the local HE method as steps in the algorithm. It enhances dark images; nevertheless, its output is a contrast that depends on the state of the images.

Methods that combine the advantages of both GT and HE technologies should be proposed, especially for large sets of images that suffer from low contrast and poor illumination, such as underwater and medical images. Moreover, the traditional challenges mentioned before must be addressed. Wang et al. [2] claimed that research on image enhancement should focus on the following issues: reduction of processing time and complexity to meet practical application needs and improvement of the effectiveness and adaptive capabilities of proposed methods.

III. Proposed Method

The objective of this work is to propose a method for improving poor illumination and low contrast, as well as preserving the natural color and details of original color images. The flow chart of the proposed method is shown in Fig. 1, whilst each step is described in detail below.

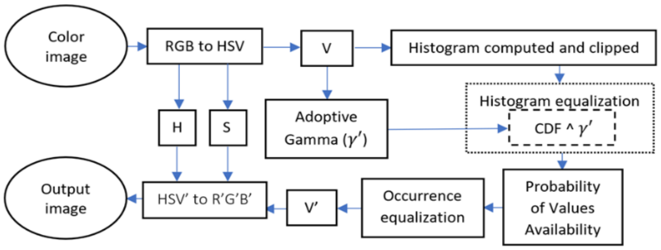


Fig. 1. The flowchart of proposed method.

A. Color Space Conversion

In human eyes, cones consist of three different photopigments that enable color perception under normal lighting conditions. When light levels drop to darkness, the human eye becomes highly sensitive to light and loses the ability to distinguish color. Therefore, illumination correction is of critical importance for enhancing images obtained under inaccurate lighting conditions. In general, RGB is not the most intuitive color model for humans; it is tied on screen hardware to display colors. By contrast, humans usually introduce illumination and color as two different related things. Several color space systems present color and illumination similar to the human visual system. HSV is one of the easy-to-use systems to identify colors and illumination and closest to how humans perceive and compare colors.

The hue (H), saturation (S) and brightness (V) in the HSV color space are independent of one another. Hue (H) is the rainbow side of color, showing where a particular color is in the visible light spectrum. Saturation (S) describes how pure a color is. Value (V) is the brightness; it is the amount of light that comes from the color. The H and S channels are both related to color. Any modification will distort the color, such that both are ignored. Enhancement of the value of brightness (V) does not affect the color information of images. The channel (V) that presents the obtained brightness matrix of an image (Fig. 2 (d)) is separated from the two channels, hue (H) and saturation (S), to be used in the proposed method individually.

The expression for converting RGB space to HSV space is as follows:

$$H = \begin{cases} 60 \times (G - B)/(V - \ min(R,G,B)) & if\ V = R \\ 120 + 60 \times (B - R)/(V - \ min(R,G,B)) & if\ V = G. \\ 240 + 60 \times (R - G)/(V - \ min(R,G,B)) & if\ V = B \end{cases} \quad (1)$$

$$S = 1 - min(R,G,B)/V \quad (2)$$

$$V = max(R,G,B) \quad (3)$$

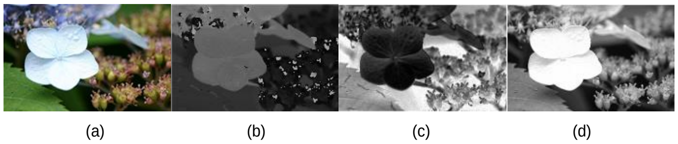


Fig. 2. (a) Original images in RGB format, (b) H channel, (c) S channel, and (d) V channel (brightness matrix) of HSV.

B. Adaptive Gamma

To effectively enhance the illumination of images, the range of low-level values should be increased significantly, the moderate-level values should be increased slightly, and the high-level values should be maintained. To achieve that, a non-linear function is adopted in this work to transform the brightness level values. The gamma function is a common non-linear transformation for illumination level curve and is expressed as:

$$G(x,y) = N(x,y)^{\gamma} \qquad (4)$$

where N refers to the normalized pixel illumination values in the range [0,1]; G(x, y) denotes the new illumination values of x, y pixels; and γ is a constant of the gamma correction parameter.

Different transformation curves can be obtained by changing the constant parameter γ . When $\gamma > 1$, the transformation will stretch to low illumination values, making images look darker. Conversely, when $\gamma < 1$, the transformation will extend to high illumination values to make images look brighter. Meanwhile, when y = 1, no transformation occurs. Despite the advantages of gamma transformation, it does not consider the global grey distribution of images, and its adaptability is poor.

To overcome these disadvantages, an adaptive gamma generator, in which the gamma parameter is acquired adaptively in accordance with the global illumination condition of an image (dark, medium or bright image), is proposed. Firstly, the mean of values in the brightness matrix (V) is calculated (Eq. (5)) to compute the brightness level of the image. Larger values indicate brighter images, whereas smaller values indicate darker images.

$$Mean = \frac{1}{X \times Y} \sum_{x,y=0}^{x,y=X-1,Y-1} V(x,y) \quad (5)$$

where X and Y refer to the size of the brightness matrix (V), and V(x, y) is the value at x, y points. Then, the adaptive gamma parameter (γ') value is calculated using the following proposed equation:

$$\gamma' = \frac{Mean}{L \times 0.33} + C \qquad (6)$$

where L is the maximum grey level of the images; here, $L = 255$, and it is multiplied by 0.33 to adapt γ' value to one of the three global states, which are dark, medium or bright, for the images. C is a bias constant to overcome the problem of $\gamma' = 0$; here, $C = 0.1$. The extracted gamma parameters produce a high transformation to brightness, which is required to unify the characteristics of different images, regardless of their original state (dark, medium, or bright), and the subsequent steps in this work.

C. Histogram Computation and Clipping

A histogram is a cumulative representation of the distribution of brightness values and displays the frequency of each value in V. Some values have higher frequency than others (Fig. 3 (c)), which produces the over-enhancement problem after HE process. To prevent over-enhancement, the histogram is clipped to reduce the high enhancement rate of higher-frequency values by using an adaptive threshold value based on the mean of histogram values. Any value that exceeds the threshold will regard the threshold as a new value, as shown in Fig. 3 (d). The calculation is performed using the following equations:

$$T_{clip} = \frac{1}{L} \sum_{l=0}^{l=L} H(l) \qquad (7)$$

$$H_{clip}(l) = T_{clip}: for\ H(l) > T_{clip} \quad (8)$$

where T_{clip} is the threshold value, H_{clip} is the clipped histogram, and l is a point in the histogram from 0 to 255.

D. Equalization Process

In this stage, HE with a proposed significant modification is performed to improve image contrast and prepare the illumination for subsequent stages. A proposed CDF function is also used to greatly improve the illumination. Firstly, the PDF of the clipped histogram is computed as follows:

$$pdf(l) = \frac{H_{clip}(l)}{\sum H_{clip}(l)} \qquad (9)$$

The computed PDF is used to calculate the CDF by using the following equation:

$$cdf(l) = \sum_{l=0}^{l=L} pdf(l) \qquad (10)$$

However, the CDF displays a curve wherein its values are within [0, 1], as shown in Fig. 3 (e); they can be assumed as normalized pixel values. This condition can be exploited to turn the CDF curve into an illuminated curve of the images. The CDF curve can be further transformed to clarify the details and improve the illumination of the images. To do this, a proposed CDF (cdf') is used to produce a new curve (Fig. 3 (f)) by using the proposed adaptive gamma parameter of Eq. (11) via the following equation:

$$cdf'(l) = \left(\sum_{l=0}^{l=L} pdf(l)\right)^{\gamma'} \qquad (11)$$

This proposed CDF function aims to improve the illumination and contrast adaptively to unify the visual properties of any image, ranging from considerably light to dark. The subsequent steps affected by any type of illumination problem are then unified. Next, the HE processes is performed. New values of brightness matrix are calculated using the following transformation function:

$$V_{MID}(x,y) = cdf'(V(x,y)) \times 255 \qquad (12)$$

where V_{MID} is the new brightness matrix that will be used in the last enhancement stage. V_{MID} has an improved contrast but has extreme high displacement towards high-brightness values, as shown in Fig. 3 (g), and new histograms, as shown in Fig. 3 (h). The histogram values shift to high-brightness regions for all image types compared with the original histograms shown in Fig. 3 (c). Therefore, the next step is required to solve this displacement.

E. Probability of Value Availability

After V_{MID} matrix is calculated, a second equalization process, which aims to modify the displacement of values equally, is applied. Here, an equal probability of density is applied instead of a normal pdf. The equal probability is calculated based on the available values only by using the following equation:

$$V_{\text{available}}\ (l) = \begin{cases} 1:\ V_{MID} \neq 0 \\ 0:\ V_{MID} = 0 \end{cases} \qquad (13)$$

$$pdf_{available}\ (l) = \frac{V_{available}}{\sum V_{\text{available}}} \qquad (14)$$

where $V_{\text{available}}$ is a list of size (L=255). Each cell has 1 if the index value is available in V_{MID}; $else, it\ has$ 0, as shown in Figure 4 (a). $pdf_{\text{available}}$ is the equal probability of each available value in V_{MID} matrix.

F. Value Equalization

Subsequently, the distribution of displaced values of V_{MID} matrix is corrected by rearranging equally to correct the illumination levels and the contrast of the processed images. Therefore, the cdf of $pdf_{available}(l)$ is calculated using Eq. (15), as shown in Fig. 4 (b), and equalized using the transformation function of Eq. (16).

$$cdf_{available}(l) = \sum_{l=0}^{l=L} pdf_{available}(l) \qquad (15)$$

$$V'(x,y) = cdf_{available}(V_{MID}(x,y)) \times 255 \quad (16)$$

From Fig. 4 (c), the result is a processed brightness matrix (V') that presents improved contrast and illumination properties of the original V channel (Fig. 3 (b)). Their new histograms in Fig. 4 (e) show balanced and arranged distribution of values that are better than the original histograms (Fig. 3 (c)).

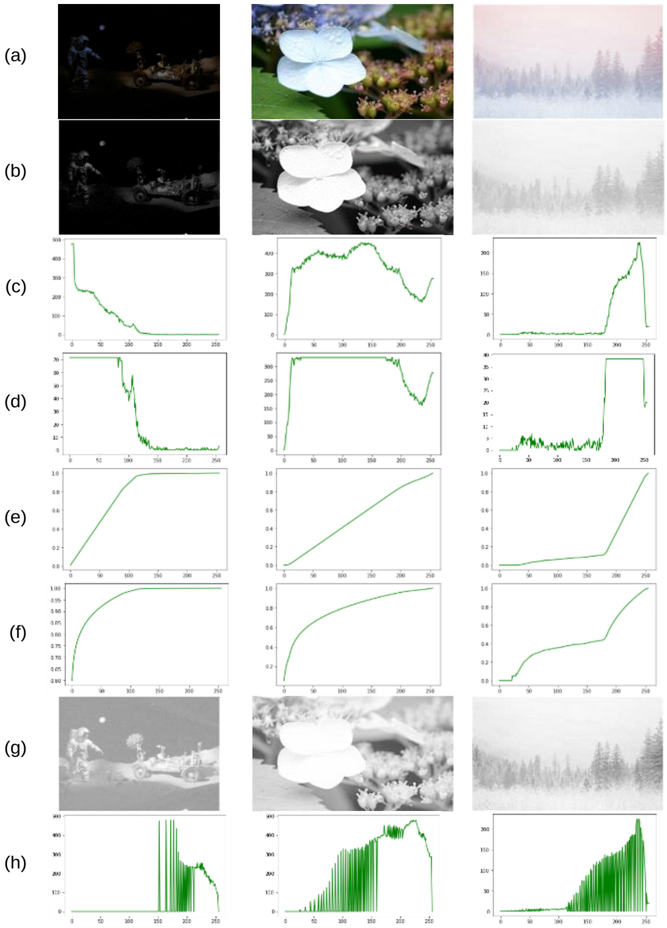


Fig. 3. (a) Original images in RGB format: dark (left), normal (middle) and overly bright (right); (b) V channel of HSV (the brightness matrix); (c) brightness matrix histogram of each image; (d) histogram of brightness matrix values; (d) clipped histogram; (e) CDF of each histogram; (f) proposed CDF (cdf') of each associated histogram; (g) new brightness matrix (V_{MID}) of each image; and (f) histogram distribution of each new brightness matrix (V_{MID}).

G. HSV Modification and Conversion into RGB

After the contrast and illumination of V' matrix have been improved in the previous stages, it is used to produce brilliant color images. V' matrix is the processed V channel, which will replace the original V channel and be recombined with the H and S channels of the original image. The new HSV' model is then returned to the RGB color space. The conversion from HSV to RGB is shown using the following formats.

$$A = V' \times S \qquad (17)$$

$$m = V' - C \qquad (18)$$

$$X = A \times (1 - \left|\left(\frac{H}{60^{\circ}}\right) mod2 - 1\right|) \qquad (19)$$

$$(r, g, b) \begin{cases} (A,X,0), & 0 \le H < 60 \\ (X,A,0), & 60 \le H < 120 \\ (0,A,X), & 120 \le H < 180 \\ (0,X,A), & 180 \le H < 240 \\ (X,0,A), & 240 \le H < 300 \\ (A,0,X), & 300 \le H, < 360 \end{cases}, \qquad (20)$$

$$R' = (r \times m) \times 255, G' = (g \times m) \times 255, B' = (b \times m) \times 255. \qquad (21)$$

where $R'G'B'$ denotes the equivalent red, green and blue channels of HSV' for each pixel in the modified image. The values of S and V' illumination-corrected channels range within (0 to 1), whilst 0º ≤ H < 360º. The results present a fine colorful image with improved illumination and contrast properties.

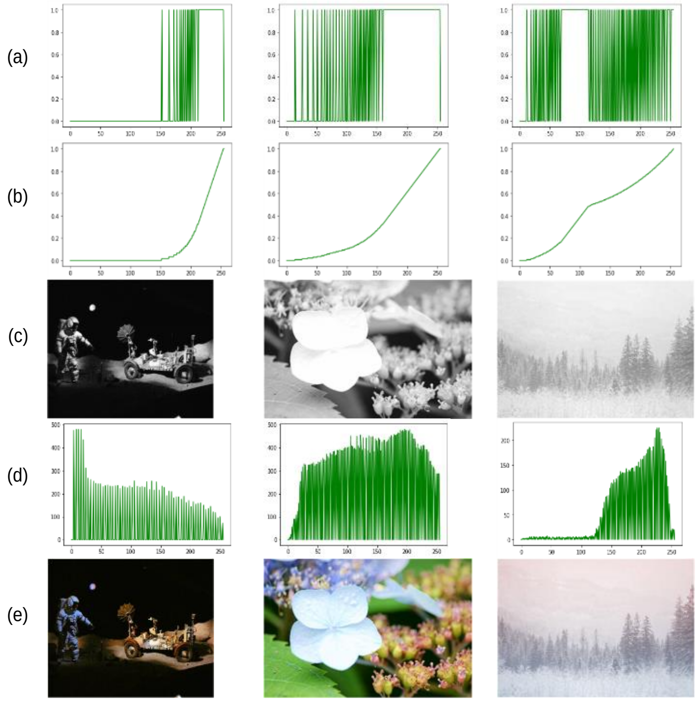


Fig. 4. (a) Pdf of occurrences availably of the new brightness matrix (V_{MID}) (Figure 4(e)), (b) CDF of each pdf, (c) Final processed brightness matrix (V'), (d) Histogram distribution of each final brightness matrix, and (e) Results images.

IV. Experiments and Results

In this section, two types of experiments are performed to evaluate the proposed method: visual and statistical experiments. A comparative analysis is also conducted with a set of well-known recent methods. In accordance with the state of the art, the bio-inspired multi-exposure (BIME) method [48], multi-scale Retinex with color restoration (MSRCR) [49], naturalness preserved enhancement (NPE) algorithm [50], simultaneous reflection and illumination estimation (SRIE) [51], multi-deviation fusion (MF) method [52], low-light image enhancement (LIME) [52] and a fusion-based enhancement method for weakly illuminated images (Dong) [53] are chosen to evaluate the performance of the proposed method.

To obtain significant results showing valid performance of each of the contributing methods, approximately 180 color images from many benchmark data sets for poor illumination and contrast challenges are used for the experiments. Specifically, the DICM data set contains **69 color** indoor and outdoor images in varying degrees of dark to high-brightness lighting conditions captured from commercial digital cameras[54]. The LIME-data data set contains **10** indoor and outdoor low-light color images[55]. The MEF data set contains **17** high-quality dark indoor and outdoor color images, including natural scene [56]. The NPE [50] data set contains **85** low-light outdoor images collected from the Internet [50]. They present outdoor natural scenes, including cloudy, daytime, daybreak, nightfall, and night-time scenes.

A. Visual Experiments

Visual experiments measure image quality based on the subjective perception of human vision. These experiments can only qualitatively assess image quality, which provide a clear idea of the image quality for humans. They are conducted on selected images from the above data sets. The most interesting cases are selected and presented in Fig. 5 to 9. The results show that all selected images are visually enhanced differently depending on the method used. In general, all images become better than original images, which is expected because the involved methods are some of the most interesting in accordance with the state of the art.

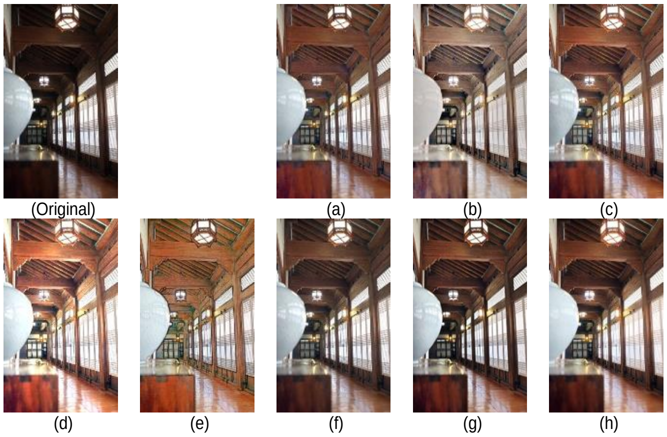


Fig. 5. Experimental results of image (59) from DICM dataset, where the original image, and its results by (a) NPE, (b) MSRCR, (c) MF, (d) LIME, (e) Dong, (f) BIME, (g) SRIE and (h) Proposed methods.

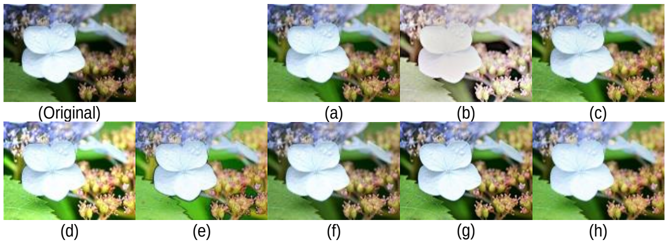


Fig. 6. Experimental results of image (50) from DICM dataset, where the original image, and its results by (a) NPE, (b) MSRCR, (c) MF, (d) LIME, (e) Dong, (f) BIME, (g) SRIE and (h) Proposed methods.

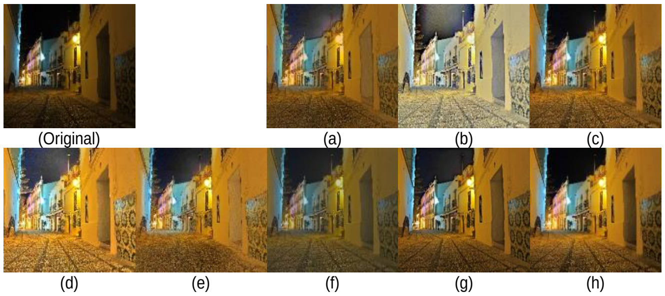


Fig. 7. Experimental results of image (5) from LIME dataset, where the original image, and its results by (a) NPE, (b) MSRCR, (c) MF, (d) LIME, (e) Dong, (f) BIME, (g) SRIE and (h) Proposed methods.

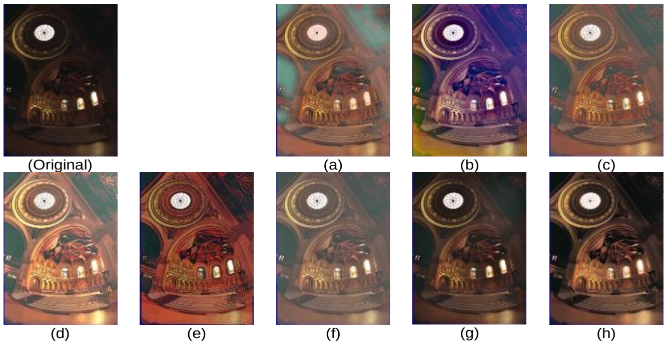


Fig. 8. Experimental results of image (Memorial) from MEF dataset, where the original image, and its results by (a) NPE, (b) MSRCR, (c) MF, (d) LIME, (e) Dong, (f) BIME, (g) SRIE and (h) Proposed methods.

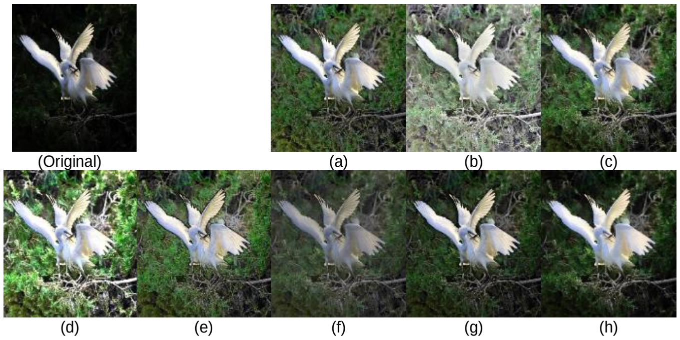


Fig. 9. Experimental results of image (birds) from NPE dataset, where the original image, and its results by (a) NPE, (b) MSRCR, (c) MF, (d) LIME, (e) Dong, (f) BIME, (g) SRIE and (h) Proposed methods.

In detail, the MSRCR [49], LIME [52] and Dong [53] methods produce images with high brightness levels. The MSRCR [49], Dong [53] and NPE [50] methods show excessive levels of enhancement, and edges appear aggressive. MSRCR [49] and LIME [52] present color fading and uneven brightness levels in some cases. The Dong [53] method produces low contrast levels caused by the effect of oil paintings on images. The BIME [48] method exhibits low brightness and contrast levels. It is the worst method in case of brightness. The SRIE [51], MF [52] and proposed methods preserve the colors of original images, but MF [52] shows faded colors in some cases. The SRIE [51] and proposed methods produce constant brightness levels with all cases of images. They preserve color effects for all involved images. In conclusion, the LIME [52] method achieves higher brightness with accepted preservation for color and details, but SIRI and the proposed method achieve higher performance in the perception of human vision.

B. Statistical Experiments

The visual experiment methodology is simple and evaluates the visual quality of images in an understandable way for humans. However, this type of assessment lacks stability and observed properties of image structures and details. In addition, visual experiments are influenced by the background, visual capability, and visual properties of images; experimental conditions; and the emotional state and motivation of observers.

To overcome the disadvantages of visual experiments, analytical statistical evaluation methods, which use objective criteria that rely on benchmark measures, are adopted. The measurement methods used in this work are mean value (MV), the standard deviation (STD) for contrast, entropy, peak signal-to-noise ratio (PSNR), structural similarity index metric (SSIM), absolute mean brightness error (AMBE) and contrast-to-noise ratio (CNR).

- **Mean** of grey values of images expresses their brightness. Larger values indicate brighter images, whereas smaller values indicate darker images.

$$Mean = \frac{1}{X\times Y}\sum_{x,y=0}^{x,y=X-1,Y-1} I(x,y) \quad (22)$$

where *X* and *Y* refer to the size of image *I*, and *I(x, y)* is the value at x, y points.

- **STD,** in greyscale values of images, measures the contrast of images. Larger values show more information and better visual properties in images

$$STD = \sqrt{\frac{\sum_{x,y=0}^{x,y=X-1,Y-1} I(x,y)-(I(x,y)-Mean)}{X\times Y}} \quad (23)$$

- **AMBE** is a common method for detecting brightness change. It evaluates the similarity between the brightness of original and processed images. It presents the absolute delta between the MVs of original and processed images. Smaller values indicate that the brightness of the processed image is closer to that of the original image.

$$AMBE = |Original_{\ Mean} - Processed_{Mean}| \quad (24)$$

- **SSIM** is popular for assessing image quality because it simulates human visual perception about the structure of images. SSIM analyses the correlations between pixels by comparing luminance, contrast and structure between original and processed images. A value closer to 1 indicates more similarity between the two images.

$$SIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (25)$$

where L is the dynamic range of the pixel values; and μ_x, μ_y are the means of x and y, respectively. σ_x^2, σ_y^2 are the variances of x and y, respectively; and σ_{xy} is the covariance of x and y.

- **CNR** simulates human perception to evaluate contrast resolution in images. It shows a visual quantitative evaluation of the detectability of defects. Smaller values are better.

$$CRN = \frac{|S_A - S_B|}{\sigma_n} \quad (26)$$

where S_A and S_B are the signal intensities (MVs) of A and B regions of interest, respectively; and σ_n is the STD of the background image noise.

- **PSNR** is a common method to measure the effect of denoising on an image. Larger values show a smaller difference between the original and processed images.

$$PSNR = 10\, log_{10}\left(\frac{255^2}{MSE}\right) \quad (27)$$

$$MSE = \frac{1}{(X \times Y)} \sum_{x,y=0}^{x,y=X-1,Y-1} (Org_{image}(x,y) - Pro_{image}(x,y))^2 \quad (28)$$

- **Entropy** is used in evaluating image quality by measuring the amount of information in images. A larger value shows more details in an image.

$$Entropy = \sum_{V=0}^{V=255} e(V) = -\sum_{V=0}^{V=255} p(V)\, log_2 p(V) (29)$$

where $p(V)$ is the probability of the grey value V in the image.

Here, the result of each measurement method could not be meaningful individually. For example, high mean and AMBE show high levels of brightness. Nonetheless, they may lead to missing information in the processed image if the accompanying STD, SSIM or PSNR values are small. Considerable difference between original and processed images and entropy indicates great effect, but they are not good with high CNR. With much side effects, excessive noise and unwanted details are produced in processed images. Therefore, all previous measurements should not be used individually to estimate the quality of output images. The relationships and interactions between all measurement values must be analyzed to evaluate performance.

Tables I to IV show the evaluation results of the involved methods. The analytical observation indicates that the SRIE [51] and proposed methods are more stable than the other methods, regardless of the type of data sets. The other methods show a large variation in their performance in accordance with the data set used. The SRIE [51] and proposed methods are the most adaptive methods for different types of images. However, the proposed method produces enhanced images that have higher brightness levels, better contrast and better preservation for color and details than the SRIE [51] method.

TABLE I. RESULTS OF SRIE, NPE, MSRCR, MF, LIME, DONG, BIME, AND PROPOSED METHODS ON DICM DATASET BY MEAN, STD, AMBE, SSIM, CNR, PSNR, AND ENTROPY MEASUREMENTS

	Mean	**STD**	**AMBE**	**SSIM**	**CNR**	**PSNR**	**Ent.**
Prop.	108.6	62.31	27.23	0.76	0.36	28.73	10.93
BIME	112.8	55.49	31.37	0.69	0.45	29.13	11.15
Dong	122.7	53.52	41.27	0.58	0.59	28.45	11.65
LIME	143.9	62.24	62.54	0.53	0.78	28.03	11.79
MF	109.8	54.52	28.46	0.69	0.41	28.5	11.5
MSRCR	153.8	53.45	73.94	0.52	1.01	27.99	12.07
NPE	108.1	55.21	27.23	0.68	0.40	28.45	11.28
SRIE	102.8	57.77	21.4	0.78	0.32	28.7	11.06
Original	81.38	56.6	0.00	1.00	0.00	100	10.10

TABLE II. RESULTS OF SRIE, NPE, MSRCR, MF, LIME, DONG, BIME, AND PROPOSED METHODS ON LIME DATASET BY MEAN, STD, AMBE, SSIM, CNR, PSNR, AND ENTROPY MEASUREMENTS

	Mean	**STD**	**AMBE**	**SSIM**	**CNR**	**PSNR**	**Ent.**
Prop.	65.97	50.5	30.47	0.62	0.54	28.37	11.08
BIME	74.01	34.3	38.52	0.52	0.85	27.55	11.07
Dong	81.94	47.5	46.45	0.41	0.83	28.08	12.05
LIME	108.9	59.2	73.41	0.31	1.07	27.81	12.65
MF	75.91	45.6	40.42	0.48	0.74	27.84	11.8
MSRCR	142.49	51.5	107	0.27	1.74	27.98	12.74
NPE	76.69	40.6	41.19	0.47	0.81	27.75	11.64
SRIE	60.36	42.7	24.87	0.65	0.48	28.22	11.15
Original	35.49	38.2	0.00	1.00	0.00	100	9.44

TABLE III. RESULTS OF SRIE, NPE, MSRCR, MF, LIME, DONG, BIME, AND PROPOSED METHODS ON MEF DATASET BY MEAN, STD, AMBE, SSIM, CNR, PSNR, AND ENTROPY MEASUREMENTS

	Mean	**STD**	**AMBE**	**SSIM**	**CNR**	**PSNR**	**Ent.**
Prop.	70.28	63.98	30.93	0.61	0.4	28.3	10.67
BIME	85.77	52.45	46.42	0.46	0.72	27.62	11.04
Dong	86.58	54.91	47.23	0.38	0.67	27.74	11.94
LIME	112.35	67.47	73	0.31	0.91	27.77	12.17
MF	77.12	53.8	37.77	0.48	0.57	27.94	11.46
MSRCR	135.19	55.75	95.84	0.3	1.33	27.97	12.41
NPE	82.18	50.55	42.83	0.44	0.67	27.78	11.44
SRIE	61.82	54.18	22.47	0.65	0.33	28.14	10.8
Original	39.35	47.59	0.00	1.00	0.00	100	8.96

TABLE IV. RESULTS OF SRIE, NPE, MSRCR, MF, LIME, DONG, BIME, AND PROPOSED METHODS ON NPE DATASET BY MEAN, STD, AMBE, SSIM, CNR, PSNR, AND ENTROPY MEASUREMENTS

	Mean	STD	AMBE	SSIM	CNR	PSNR	Ent.
Prop.	116.29	66.27	34.42	0.77	0.44	27.97	11.56
BIME	114.46	52.84	33.41	0.74	0.51	27.73	11.49
Dong	132.57	57.36	50.71	0.58	0.71	28.12	12.31
LIME	157.15	62.02	75.29	0.53	0.96	27.91	12.22
MF	116.49	55.23	34.62	0.7	0.52	28.3	12.01
MSRCR	162.74	52.5	80.87	0.52	1.08	27.97	11.99
NPE	115.11	56.05	33.35	0.72	0.5	28.05	11.83
SRIE	106.99	61.95	25.12	0.82	0.38	28.3	11.52
Original	81.87	60.96	0.00	1.00	0.00	100	10.81

C. Analysis and Discussion

The objective of the conducted experiments is to evaluate the performance of the respective methods in accordance with their ability to enhance image quality by improving contrast and brightness, as well as preserving the details and color of processed images similar to the original ones. From the literature review, SRIE [51], NPE [50], MSCRR, MF [52], LIME [52], Dong [53] and BIME [48] are popular methods that deal with many challenges of low contrast and brightness. Therefore, high performance is expected using these methods to compare with the proposed method. On the basis of the results, the following conclusions are drawn:

- The **MSRCR** [49] method shows the highest levels of brightness by recording the highest mean and AMBE values. However, it destroys the contrast in many cases, which is evident because its STD values are lower than those of the original images in the DICM and NPE data sets (Tables I to IV). In addition, it shows worse SSIM and CNR results than the other methods. It also presents worse MVs of PSNR (specifically, it is the worst in the DICM data set) and the farthest entropy value of the original images. The results of SSIM, CNR, PSNR and entropy show that this method is the worst for preserving the color and natural details of processed images.

- The **LIME** [52] method has the second-highest mean and AMBE values after the MSRCR method. It increases brightness more than the methods mentioned. This correlates with the highest STD values compared with the other methods. It is one of the best methods to improve the contrast of enhanced images. Nevertheless, its SSIM, CNR and entropy are the worst after the MSRCR method, in addition to the varying PSNR values in accordance with the applied data set. This method is unstable based on the applied images, and it is one of the worst methods used to preserve the details of the processed images.

- The **Dong** [53] method shows average brightness levels by scoring average mean and AMBE values. However, this method reduces the contrast of images in many cases with STD values that are lower than those of the original images in some data sets. Moreover, it shows some of the worst results for SSIM, CNR, PSNR and entropy compared with other methods. The results indicate that the performance of this method varies in accordance with input images, and it is not good at preserving the color and natural details of processed images.

- The **BIME** [48] method shows average brightness levels by scoring average mean and AMBE values. Nonetheless, it significantly reduces the contrast of images. It shows the lowest STD values, and its STD is smaller than that of the original images in most cases. This method scores an average value for SSIM, CNR, PSNR and entropy compared with other methods. Therefore, its ability to preserve colors and details is intermediate amongst the participating methods.

- The **MF** [52] method shows average brightness with average values of mean and AMBE, but it reduces the contrast of processed images compared with that of the original ones. It shows the lowest STD in most cases. The ability of this method to preserve colors and details is intermediate but unstable amongst the involved methods by scoring varying average values for SSIM, CNR, PSNR and entropy.

- The **NPE** [50] method gives average values of mean and AMBE and average brightness levels in processed images. However, its STD values are less than those of the original images in most cases. It reduces the contrast of processed images. It scores an average value for SSIM, CNR, PSNR and entropy. Therefore, its ability to preserve colors and details is intermediate amongst the involved methods.

- The **SRIE** [51] method shows average but the lowest brightness levels compared with the involved methods by scoring the lowest mean and AMBE values in all data sets. Nevertheless, it improves the contrast of resulting images by scoring higher STD values than those of the original images in all image cases. In addition, the **SRIE** [51] method exhibits the best SSIM and CNR results in all data sets, as well as better-than-average PSNR and entropy. Therefore, it is one of the best methods to preserve colors and details of processed images.

- The **proposed** method shows average brightness levels by scoring average values in mean and AMBE. The STD shows that it is the best method to improve the contrast of outputs. The proposed method scores higher STD in two data sets and presents the best performance amongst the involved methods overall. Moreover, the proposed method shows the second-best SSIM and CNR results in all data sets, next to the SRIE [51] method. It scores the best or better-than-average PSNR values in all data sets. Therefore, it is one of the best methods to preserve colors and details of processed images. In case of entropy, it achieves the closest values to the original images, which means its results are the closest to the original images.

Based on the above observations, the performance of the methods varies between the brightness levels and the accuracy of preserving the properties of the resulting images. The LIME [52] and MSRCR [49] methods achieve higher brightness levels amongst the respective methods but fail to preserve the color and information of the original images. Whilst MSRCR performs the worst, the LIME [52] method can be accepted for human vision applications. However, it loses considerable information, which makes it inefficient for computer vision applications.

On the contrary, the performance of the remaining methods, such as SRIE [51], NPE [50], MF [52], LIME [52], Dong [53], BIME [48] and the proposed method, varies in brightness from average to better than average. Nonetheless, many of these methods reduce the contrast and fail to preserve the color and detail of the images, such as the NPE [50], MF [52], LIME [53], Dong [53] and BIME [48] methods. They show variance and over-enhancement performance for many image situations. Therefore, they can be useful in specific cases of images. However, in general, they cannot adapt to all situations of human vision and computer vision applications.

In cases of preserving image properties, such as color, contrast and detail, only the SRIE [51] and proposed methods achieve the best performance. In addition to stable performance in enhancing brightness levels, they improve contrast, produce fewer side effects and show better color and detail preservation similar to original images. SRIE [51] and the proposed method are the most suitable in computer vision applications. Nevertheless, comparison of the two methods shows that the proposed method is better at improving contrast and keeping details closer to the original images. In conclusion, the proposed method is the most adaptive and stable regardless of the type of image applied. In future work, we aim to conserve details in images with extremely varied levels to highlight all the details in the image and prevent any loss of information.

V. Conclusion

In this work, a new method for color image brightness and contrast correction based on the advantages of non-linear function in gray transformation and histogram equalization techniques is proposed. The proposed method consists of set stages: the original red, green and blue (RGB) image is converted into the HSV color space, and the V channel is used for enhancement. Next, an adaptive gamma generator is proposed to calculate gamma parameters in accordance to dark, medium, or bright image conditions. This parameter is used to propose a cumulative distribution function that produces an optimized curve for illumination values. Then, a second modified equalization is performed to evenly correct the offset of the illumination curve values based on the equal probability of the available values only. Finally, the processed V channel replaces the original V channel, and the new HSV model returns to the RGB color space. The experiments show that the proposed method significantly improves the low contrast and poor illumination of the color image while preserving the color and details of the original image. It is the most adaptive and stable method, regardless of the type of image applied, compared to other state-of-the-art methods.

Acknowledgment

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4361009DSR03).

References

[1] B. Bataineh and K. H. Almotairi, “Enhancement Method for Color Retinal Fundus Images Based on Structural Details and Illumination Improvements,” Arab. J. Sci. Eng., vol. 46, no. 9, pp. 8121–8135, 2021, doi: 10.1007/s13369-021-05429-6.

[2] W. Wang, X. Wu, X. Yuan, and Z. Gao, “An Experiment-Based Review of Low-Light Image Enhancement Methods,” IEEE Access, vol. 8, pp. 87884–87917, 2020, doi: 10.1109/ACCESS.2020.2992749.

[3] K. G. Dhal, A. Das, S. Ray, J. Gálvez, and S. Das, “Histogram Equalization Variants as Optimization Problems: A Review,” Arch. Comput. Methods Eng., vol. 28, no. 3, pp. 1471–1496, 2021, doi: 10.1007/s11831-020-09425-1.

[4] M. Jian, X. Liu, H. Luo, X. Lu, H. Yu, and J. Dong, “Underwater image processing and analysis: A review,” Signal Process. Image Commun., vol. 91, p. 116088, 2021, doi: 10.1016/j.image.2020.116088.

[5] P. Li, F. Wang, Y. Liang, and X. Zhang, “Single Image Defogging Method Based on Adaptive Modified Dark Channel Value,” vol. 91, no. Msbda, pp. 148–153, 2019, doi: 10.2991/msbda-19.2019.23.

[6] W. A. Mustafa and M. M. M. Abdul Kader, “Contrast Enhancement Based on Fusion Method: A Review,” J. Phys. Conf. Ser., vol. 1019, no. 1, 2018, doi: 10.1088/1742-6596/1019/1/012025.

[7] V. S. Padmavathy and R. Priya, “Image contrast enhancement techniques-a survey,” Int. J. Eng. Technol., vol. 7, no. 2.33 Special Issue 33, pp. 466–469, 2018, doi: 10.14419/ijet.v7i1.1.10146.

[8] C. Li, S. Tang, J. Yan, and T. Zhou, “Low-light image enhancement via pair of complementary gamma functions by fusion,” IEEE Access, vol. 8, pp. 169887–169896, 2020, doi: 10.1109/ACCESS.2020.3023485.

[9] W. Kim, R. Lee, M. Park, and S. H. Lee, “Low-Light Image Enhancement Based on Maximal Diffusion Values,” IEEE Access, vol. 7, pp. 129150–129163, 2019, doi: 10.1109/ACCESS.2019.2940452.

[10] R. R. Hussein, Y. I. Hamodi, and R. A. Sabri, “Retinex theory for color image enhancement: A systematic review,” Int. J. Electr. Comput. Eng., vol. 9, no. 6, pp. 5560–5569, 2019, doi: 10.11591/ijece.v9i6.pp5560-5569.

[11] J. Dabass and R. Vig, “Biomedical image enhancement using different techniques - A comparative study,” Commun. Comput. Inf. Sci., vol. 799, pp. 260–286, 2018, doi: 10.1007/978-981-10-8527-7_22.

[12] M. Veluchamy and B. Subramani, “Image contrast and color enhancement using adaptive gamma correction and histogram equalization,” Optik (Stuttg)., vol. 183, pp. 329–337, 2019, doi: 10.1016/j.ijleo.2019.02.054.

[13] A. Asokan, D. E. Popescu, J. Anitha, and D. J. Hemanth, “Bat algorithm based non-linear contrast stretching for satellite image enhancement,” Geosci., vol. 10, no. 2, pp. 1–12, 2020, doi: 10.3390/geosciences10020078.

[14] Q. Liu et al., “Single Image Defogging Method Based on Image Patch Decomposition and Multi-Exposure Image Fusion,” Front. Neurorobot., vol. 15, no. July, pp. 1–14, 2021, doi: 10.3389/fnbot.2021.700483.

[15] J. Yoon and Y. Choe, “Retinex based image enhancement via general dictionary convolutional sparse coding,” Appl. Sci., vol. 10, no. 12, pp. 1–18, 2020, doi: 10.3390/app10124395.

[16] R. Wang, Q. Zhang, C. W. Fu, X. Shen, W. S. Zheng, and J. Jia, “Underexposed photo enhancement using deep illumination estimation,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 6842–6850, 2019, doi: 10.1109/CVPR.2019.00701.

[17] N. Dey, “Uneven illumination correction of digital images: A survey of the state-of-the-art,” Optik (Stuttg)., vol. 183, no. February, pp. 483–495, 2019, doi: 10.1016/j.ijleo.2019.02.118.

[18] Y. Huang, Y. Li, and Y. Zhang, “A Retinex image enhancement based on L channel illumination estimation and gamma function,” vol. 137, no. Jiaet, pp. 312–317, 2018, doi: 10.2991/jiaet-18.2018.55.

[19] E. H. Land and J. J. McCann, “Lightness and Retinex Theory,” J. Opt. Soc. Am., vol. 61, no. 1, pp. 1–11, 1971, doi: 10.1364/JOSA.61.000001.

[20] K. Akila, S. Chitrakala, and S. Vaishnavi, “Survey on illumination condition of video/image under heterogeneous environments for enhancement,” ICACCS 2016 - 3rd Int. Conf. Adv. Comput. Commun. Syst. Bringing to Table, Futur. Technol. from Arround Globe, 2016, doi: 10.1109/ICACCS.2016.7586389.

[21] J. Xiong et al., “Application of Histogram Equalization for Image Enhancement in Corrosion Areas,” Shock Vib., vol. 2021, 2021, doi: 10.1155/2021/8883571.

[22] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, “Underexposed photo enhancement using deep illumination estimation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6849–6857.

[23] S. Park, S. Yu, M. Kim, K. Park, and J. Paik, “Dual autoencoder network for retinex-based low-light image enhancement,” IEEE Access, vol. 6, pp. 22084–22093, 2018.

[24] H. Singh, A. Kumar, L. K. Balyan, and G. K. Singh, “Swarm intelligence optimized piecewise gamma corrected histogram equalization for dark image enhancement,” Comput. Electr. Eng., vol. 70, pp. 462–475, 2018.

[25] W. Ren et al., “Low-light image enhancement via a deep hybrid network,” IEEE Trans. Image Process., vol. 28, no. 9, pp. 4364–4375, 2019.

[26] Y. Deng, C. C. Loy, and X. Tang, “Aesthetic-driven image enhancement by adversarial learning,” in Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 870–878.

[27] Y.-S. Chen, Y.-C. Wang, M.-H. Kao, and Y.-Y. Chuang, “Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6306–6314.

[28] M. N. Aziz, T. W. Purboyo, and A. L. Prasasti, “A survey on the implementation of image enhancement,” Int. J. Appl. Eng. Res., vol. 12, no. 21, pp. 11451–11459, 2017.

[29] C. R. Nithyananda, A. C. Ramachandra, and Preethi, “Review on Histogram Equalization based Image Enhancement Techniques,” Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016, pp. 2512–2517, 2016, doi: 10.1109/ICEEOT.2016.7755145.

[30] E. Irmak and A. H. Ertas, “A review of robust image enhancement algorithms and their applications,” 2016 4th IEEE Int. Conf. Smart Energy Grid Eng. SEGE 2016, no. December, pp. 371–375, 2016, doi: 10.1109/SEGE.2016.7589554.

[31] J. P. Greco et al., “Illuminating Low Surface Brightness Galaxies with the Hyper Suprime-Cam Survey,” Astrophys. J., vol. 857, no. 2, p. 104, 2018, doi: 10.3847/1538-4357/aab842.

[32] Q. Xu, H. Jiang, R. Scopigno, and M. Sbert, “A novel approach for enhancing very dark image sequences,” Signal Processing, vol. 103, pp. 309–330, 2014, doi: 10.1016/j.sigpro.2014.02.013.

[33] Z. Feng and S. Hao, “Low-Light Image Enhancement by Refining Illumination Map with Self-Guided Filtering,” Proc. - 2017 IEEE Int. Conf. Big Knowledge, ICBK 2017, pp. 183–187, 2017, doi: 10.1109/ICBK.2017.37.

[34] L. Florea, C. Florea, and C. Ionascu, “Avoiding the Deconvolution: Framework Oriented Color Transfer for Enhancing Low-Light Images,” IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., pp. 936–944, 2016, doi: 10.1109/CVPRW.2016.121.

[35] Z. Zhou, N. Sang, and X. Hu, “Global brightness and local contrast adaptive enhancement for low illumination color image,” Optik (Stuttg)., vol. 125, no. 6, pp. 1795–1799, 2014, doi: 10.1016/j.ijleo.2013.09.051.

[36] W. Wang, Z. Chen, X. Yuan, and X. Wu, “Adaptive image enhancement method for correcting low-illumination images,” Inf. Sci. (Ny)., vol. 496, no. July, pp. 25–41, 2019, doi: 10.1016/j.ins.2019.05.015.

[37] L. Tao and V. Asari, “An integrated neighborhood dependent approach for nonlinear enhancement of color images,” Int. Conf. Inf. Technol. Coding Comput. ITCC, vol. 2, no. 1, pp. 138–139, 2004, doi: 10.1109/itcc.2004.1286612.

[38] G. Cao, L. Huang, H. Tian, X. Huang, Y. Wang, and R. Zhi, “Contrast enhancement of brightness-distorted images by improved adaptive gamma correction,” Comput. Electr. Eng., vol. 66, pp. 569–582, 2018, doi: 10.1016/j.compeleceng.2017.09.012.

[39] R. C. Gonzalez and R. E. Woods, “Digital image processing.” Prentice hall Upper Saddle River, NJ, 2002.

[40] M. Yang, G. Tang, X. Liu, L. Wang, Z. Cui, and S. Luo, “Low-light image enhancement based on Retinex theory and dual-tree complex wavelet transform,” Optoelectron. Lett., vol. 14, no. 6, pp. 470–475, 2018.

[41] H. Yoon, Y. Han, and H. Hahn, “Image contrast enhancement based sub-histogram equalization technique without over-equalization noise,” World Acad. Sci. Eng. Technol., vol. 50, p. 2009, 2009.

[42] K. Kapoor and S. Arora, “Colour Image Enhancement based on Histogram Equalization,” Electr. Comput. Eng. An Int. J., vol. 4, no. 3, pp. 73–82, 2015, doi: 10.14810/ecij.2015.4306.

[43] L. Zhuang and Y. Guan, “Image Enhancement via Subimage Histogram Equalization Based on Mean and Variance,” Comput. Intell. Neurosci., vol. 2017, 2017, doi: 10.1155/2017/6029892.

[44] P. Kandhway, A. K. Bhandari, and A. Singh, “A novel reformed histogram equalization based medical image contrast enhancement using krill herd optimization,” Biomed. Signal Process. Control, vol. 56, p. 101677, 2020, doi: 10.1016/j.bspc.2019.101677.

[45] S. F. Tan and N. A. M. Isa, “Exposure based multi-histogram equalization contrast enhancement for non-uniform illumination images,” IEEE Access, vol. 7, pp. 70842–70861, 2019.

[46] S. C. Huang, F. C. Cheng, and Y. S. Chiu, “Efficient contrast enhancement using adaptive gamma correction with weighting distribution,” IEEE Trans. Image Process., vol. 22, no. 3, pp. 1032–1041, 2013, doi: 10.1109/TIP.2012.2226047.

[47] H. Singh, A. Kumar, L. K. Balyan, and H.-N. Lee, “Optimally sectioned and successively reconstructed histogram sub-equalization based gamma correction for satellite image enhancement,” Multimed. Tools Appl., vol. 78, no. 14, pp. 20431–20463, 2019.

[48] Z. Ying, G. Li, and W. Gao, “A Bio-Inspired Multi-Exposure Fusion Framework for Low-light Image Enhancement,” vol. 14, no. 8, pp. 1–10, 2017, [Online]. Available: http://arxiv.org/abs/1711.00591.

[49] A. B. Petro, C. Sbert, and J.-M. Morel, “Multiscale Retinex,” Image Process. Line, vol. 4, pp. 71–88, 2014, doi: 10.5201/ipol.2014.107.

[50] S. Wang, J. Zheng, H. M. Hu, and B. Li, “Naturalness preserved enhancement algorithm for non-uniform illumination images,” IEEE Trans. Image Process., vol. 22, no. 9, pp. 3538–3548, 2013, doi: 10.1109/TIP.2013.2261309.

[51] X. Fu, D. Zeng, Y. Huang, X. P. Zhang, and X. Ding, “A Weighted Variational Model for Simultaneous Reflectance and Illumination Estimation,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2782–2790, 2016, doi: 10.1109/CVPR.2016.304.

[52] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. Paisley, “A fusion-based enhancing method for weakly illuminated images,” Signal Processing, vol. 129, pp. 82–96, 2016, doi: 10.1016/j.sigpro.2016.05.031.

[53] X. Dong, Y. Pang, and J. Wen, “Fast efficient algorithm for enhancement of low lighting video,” ACM SIGGRAPH 2010 Posters, SIGGRAPH ’10, p. 8811, 2010, doi: 10.1145/1836845.1836920.

[54] C. Lee, C. Lee, and C.-S. Kim, “Contrast enhancement based on layered difference representation,” in 2012 19th IEEE international conference on image processing, 2012, pp. 965–968.

[55] X. Guo, Y. Li, and H. Ling, “LIME: Low-light image enhancement via illumination map estimation,” IEEE Trans. Image Process., vol. 26, no. 2, pp. 982–993, 2017, doi: 10.1109/TIP.2016.2639450.

[56] K. Ma, K. Zeng, and Z. Wang, “Perceptual quality assessment for multi-exposure image fusion,” IEEE Trans. Image Process., vol. 24, no. 11, pp. 3345–3356, 2015.

A Study on Institution Improvement Plans for the National Supercomputing Joint Utilization System in South Korea

Hyungwook Shim, Myungju Ko, Yungeun Choe, Jaegyoon Hahm*
National Supercomputer Department, Korea Institute of Science and Technology Information,
Daejeon, Republic of Korea

***Abstract*—The purpose of this paper is to discover institutional gaps in the supercomputing joint utilization system that the government is actively promoting as an alternative to the problem of shortage of domestic supercomputing resources. The institutional gaps were discovered by examining the current status of laws, top-level plans, and operating guidelines related to the current joint utilization system and matching them with problems or issues that need to be resolved socially. The improvement plan for the institutional gaps was derived at a level that can be reflected in the operating guidelines of the Specialized Center and Unit Center so that the performing subject constituting the joint utilization system can directly participate and solve it. In the future, for the effective operation of the joint utilization system, we plan to promote the domestic market through the diffusion of research results and secure external technological competitiveness by reflecting the contents of institution improvement.**

Keywords*—Component; supercomputer; joint utilization system; institutional gap; national supercomputing center; specialized center

I. Introduction

Looking at major policies and technology development trends of major supercomputer countries, there is a tendency to integrate and link resources for efficient construction and utilization of computing resources. Leading countries such as XSEDE and ACCESS in the U.S., EuroHPC JU and PRACE in Europe, and Flagship 2020 strategy in Japan collect and intensively support supercomputer resources for key strategic areas at the national level, and also provide large-scale resources to various academic and industrial users as needed. In South Korea, as demand for AI-based supercomputer calculations is gradually increasing, the need and urgency for a joint utilization system for efficient use of resources has increased. Therefore, the government is making efforts to prepare a joint utilization system as an alternative to securing insufficient supercomputing resources. The joint utilization system refers to a system that is jointly utilized at the national level by linking domestic supercomputer resources, and converts from the existing governance system of a single national center to an ecosystem composed of national-specialized-unit centers. In 2021, with the designation of the 1st specialized center (7 fields, 7 institutions) and the beginning of the establishment of the joint utilization system in Korea, institutional support for smooth system establishment and non-discriminatory and fair participation of companies is essential. In particular, considering the nature of the structure of the supercomputer industry, where small and medium-sized venture companies participate and the number of experts is limited, influx of new companies and mid- to long-term development are important. Therefore, this paper analyzes the current status of the joint utilization system, support system that has been overhauled, focuses on participating organizations, discovers blank areas related to issues that are currently socially required to be resolved, and proposes institutional improvement measures [1-3].

This paper consists of a total of six sections. In the introduction of Section I, it was explained by presenting the background and necessity of the research, and the purpose. The Section II and III introduce the supercomputer joint utilization system, explain its functions and compositional system. Also, previous studies related to supercomputing joint utilization system are investigated, and academic values such as differentiation and novelty from this paper are reviewed. Section IV analyzes the current institutional status of the joint utilization system in Korea and discovers gaps. Section V analyzes the causes of the institutional vacuum and presents improvement plans. In the conclusion of Section VI, the analysis results were summarized and implications and future utilization plans were mentioned.

II. Theoretical Background

The supercomputing joint utilization system refers to a system in which the supercomputing resources installed in the domestic government, companies, research institutes, and schools in accordance with Article 17 of the (Supercomputer Act) is interlocked with joint utilization resources and used as needed at the national level. The main body of the joint utilization system is composed of a National Center, a Specialized Center, and a Unit Center according to the (Supercomputer Innovation Strategy) (referred to as Innovation Strategy). Currently, KISTI as the National Center and seven institutions, Korea Meteorological Administration, GIST (Gwangju Institute of Science and Technology) as the Specialized Center, has been selected. In the future, Specialized Centers and Unit Centers will continue to expand considering the needs of each field and computing demand. Jointly utilized resources are used for research areas where there is no Specialized Center or research that has a Specialized

Center but requires other types of resources, and there are plans to flexibly respond to various demands such as emergency demand support such as COVID-19. In the joint utilization system, resource linkage and service provision are made through the joint utilization platform. The platform closely links common resources with different resource characteristics and operating environments and improves user accessibility by providing cloud-based integrated services. Therefore, the following ripple effects can be obtained for users who use common resources [4-5].

- Establish a system that can integrate and utilize jointly utilized resources by National Center, Specialized Centers, Unit Centers to expand national supercomputing service resources and utilization base
- Improved scalability so that users can utilize resources as much as they need by integrating by applying cloud technology without being tied to physical environment of specific facilities and specific public institutions
- Improved user accessibility and convenience so that first-time users and advanced users who require various specialized languages and execution environments can use supercomputing services without restrictions and inconveniences.

III. Literature Review

Lee (2018) performed an economic analysis on the effect of joint utilization of supercomputers. A cost-benefit analysis method was used as an economic feasibility analysis method, and the economic effect was analyzed by comparing the cost required for the establishment and operation of the existing centralized system. Economic effect means cost reduction effect. Hardware, software construction and purchase costs, operating manpower and maintenance costs were selected as cost reduction factors [6]. Kim (2015) analyzed trends on supercomputer innovation policies in the manufacturing sector and drew implications. Manufacturing innovation plans of advanced manufacturing powerhouses in the United States, Japan, and Germany were analyzed to investigate the use of supercomputers in manufacturing innovation plans and the promotion trends of related organizations and projects. In order to expand the use of supercomputers in the domestic manufacturing sector, it is concluded that participation of small and medium-sized venture companies is necessary, and efforts to prepare related policies are urged [7]. Nam (2016) introduces a plan to introduce and operate a supercomputer system led by a National Center and secure self-development capabilities as a way to secure national competitiveness of supercomputers. As the main contents, it describes the demand forecast of domestic resources, development goals and strategies for the introduction of supercomputers, and analyzes the operating environment for the establishment of a supercomputer ecosystem in the long term, and presents future plans and expected effects [8]. Huh (2021) suggested legal and institutional improvements to secure technological competitiveness in the future supercomputer market. Problems were identified through analysis of relevant legal systems and ecosystem conditions, and improvements were suggested, such as strengthening the role of the government, National Centers, and Specialized Centers, supporting industries, and promoting commercialization. As for the problems with the (supercomputer law), the highest law related to supercomputers, the factors that impede the linkage between national R&D achievements and industries, and the factors that are ineffective in enforcement were analyzed with an emphasis [9].

Summarizing the content of previous studies, it can be divided into studies that analyze economic effects on the subject of joint use of supercomputers, and studies that present legal and institutional improvements or plans for supercomputers. In the case of the supercomputer joint utilization system, since it is being promoted starting with the designation of a Specialized Center in 2021, there has been no research dealing with the legal and institutional aspects yet. Although many studies on supercomputers and related laws and top-level planning units have been published, there are no studies dealing with subsystems such as operating guidelines related to the joint utilization system. In addition, this paper derives legal and institutional improvements related to the joint utilization system, so it is novel compared to previous studies. As for the improvement plan, the status of problems and issues in the domestic supercomputer industry was investigated, and through the analysis of the OS matrix with the solution plan, inventiveness was secured and academic value was enhanced in presenting the improvement plan by focusing on the institutional gaps.

IV. Joint Utilization System Institutional Status Analysis

A. Identification of Problems and Issues

Institutional problems and issues of the joint utilization system were investigated through individual interviews with supercomputer forum members, etc., and a number of problems and issues presented were prioritized in order of frequency and selected. Problems and issues including the lack of supercomputer computational resources that can be used jointly, the absence of source technology related to supercomputer manufacturing, the insignificant participation of SMEs and the lack of professional manpower due to the ecosystem centered on government-funded research institutes, schools, and large corporations has been derived

B. Institutional Support Status

The status of institutional support for the derived problems and issues was derived based on the possibility of solving problems and issues and the level of association as an alternative, with reference to the legal system related to the joint utilization system. First, as a solution to the lack of infrastructure, two alternatives can be suggested: expansion of domestic supercomputer infrastructure and utilization of overseas infrastructure. The legal basis for the expansion of domestic infrastructure can be found in Articles 11 and 17 of the (Supercomputer Act), and it is stated that the government must provide support for infrastructure expansion. Second, as an alternative to the absence of source technology, direct technology development and introduction of foreign technology can be presented as alternatives, and institutional grounds exist for the expansion of investment in R&D projects and technology exchange through international cooperation. There are two alternatives to the problem of insignificant private participation: inducing participation of private companies and direct promotion. By law, the government must provide preferential support to technology-intensive SMEs related to supercomputing and the establishment of a joint utilization system in which companies, institutions, organizations, and universities participate. Lastly, there are alternatives to the lack of professional manpower, such as fostering professional manpower, developing training programs, and utilizing overseas professional manpower. In accordance with Articles 9 and 12 of the (Supercomputer Act), the government and the National Center must establish a plan for fostering and supplying specialized manpower, and make efforts to introduce advanced technology through attracting excellent foreign manpower and international cooperation.

As a result of the current status analysis shown in Table I, it can be considered that the response to industrial issues related to the joint utilization system is institutionally prepared at a certain level or higher. However, most of the institutional support specified is centering on the government and National Centers, and there are relatively few details on Specialized Centers and Unit Centers. Therefore, it is necessary to look at the support institution for the sub-participating organizations based on the performance system of the joint utilization system.

TABLE I. INSTITUTIONAL SUPPORT STATUS ANALYSIS RESULT

Problems and issues	Alternative solution	Relevant basis(Supercomputer Act)
Lack of infrastructure	Infrastructure expansion	Articles 11 and 17 of the Act
	Utilization of overseas infrastructure	-
Absence of source technology	Core/original technology development	Articles 10 and 16 of the Act
	Introduction of overseas core/original technology	Article 19 of the Act
Lack of private participation	Encourage participation of participating private companies (provision of incentives)	Article 17 of the Act
	Nurturing private companies	Article 18 of the Act, Enforcement decree of the Act 15-2
Lack of professional manpower	Enforcement Decree Establishment of curriculum (graduate school)	Article 12 of the Act, Enforcement decree of the Act 13
	Utilization of overseas experts	Article 19 of the Act
	Education training and program development	Article 9 of the Act

C. Discover Institutional Gaps

The analysis of the current status of institutional support in the joint utilization system was conducted targeting the Specialized Centers currently. As for the analysis method, a 2 by 2 matrix analysis method was used for the function of the Specialized Centers and the support for function execution. The function of the Specialized Center targets the main functions specified in the (Supercomputer Act). The contents of the support matched the contents of the “Innovation Strategy” the top-level supercomputer plan, and the “Operation Guidelines” of the Specialized Center, the subsystem, for budget, technology, and network, which are essential factors of the joint utilization system. First, the support contents of the “Innovation Strategy” and the joint utilization system of the “Operation Guidelines” were coded as Table II. In the “Innovation Strategy”, related institution specify services including service providers and users, educational programs, data management systems, etc., and “Operation Guidelines” support costs, technologies, and domestic and international networks for service operation and management.

TABLE II. CODING RESULT

[Innovation Strategy]	[Specialized Center Operation Guidelines]
AA - Designation and development of 10 Specialized Centers for 30 years AB - Training and dispatch to Specialized Centers, provision of technical education programs(strengthening manpower capacity) AC - Operation of a joint utilization consultative body(Establishment of resource distribution standards and establishment of an operating system) AD - Provides cloud-based platform service to link common resources and improve user accessibility AE - Establishment of Data Hub - Establishment of integrated data storage and preparation of data standard system AF - Expand integrated sharing of data in connection with domestic and foreign data centers (data dam: bio, material, transportation, disaster, environment, etc.) and platforms	BA - Linked technology support BB - Resource building budget BC - Providing opportunities to participate in R&D projects BD - Training and technical exchange support BE - Operating expenses BF - Network BG - International cooperation, technology exchange, etc.

The matrix analysis results are shown in Table III. In the function of the Specialized Center for building resources, institutional support was established in all factors of budget, technology, and network. The contents of support for operation and service provision and basic adaptation research are also presented in the “Innovation Strategy” and “Operation Guidelines”. In terms of data management and operation support and manpower training, the grounds for support are meticulously prepared without institutional gaps. On the other hand, in the part related to the dissemination of research results, the budget part belongs to the realm of the institutional gap. Dissemination of research results refers to all cases in which research results related to the joint utilization system are transferred to the private sector, schools, research institutes, etc. Support for the dissemination of research results refers to technology value evaluation costs, patent-related costs, standards-related costs, etc. under the “Act on Performance Evaluation and Performance Management of National R&D Projects”. In the case of other technology fields, various government-led financial support is provided. Therefore, although the current support institution for the joint utilization system is prepared at a compliant level, it can be analyzed as a situation where the financial support system is insufficient in terms of performance utilization.

TABLE III. MATRIX ANALYSIS

Specialized Center Functions	Support content		
	Budget	Technology	Network
Building resources	BB, AA	BA	AC
Operation and service provision	BE	BA	AC, AD
Basic and applied research	BC	BA, BC	BG
Dissemination of research results	-	BG	BD, BF
Data management and operation support	BE, AE	BA, AE, AF	AD, AF
manpower training	BD AB	BD, AB	BG

V. INSTITUTIONAL IMPROVEMENT PLAN

A. Cause Analysis

In order to prepare improvement plans for the joint utilization system support institution, it is necessary to analyze the cause of the gaps. For the analysis of causes, recent domestic supercomputer-related statistical data and reports were referred to, and the following four causes were derived by comprehensively considering the influence and solution potential among various causes.

1) Low participation of private enterprises: The budget for dissemination of research achievements is targeted at institutions that have research achievements. According to the National Center statistics, the utilization rate of national supercomputers by private companies in 2022 is very low at about 3.9% of the total. Given the computational capacity of the National Center, which supports about 140 institutions for a year, the number of private companies that commercialize research results is very small, considering the number of participating private companies. The participation of SMEs in the private sector is also very small. Even at KSC (Korea Supercomputing Conference), the largest conference in Korea, the committee is mostly composed of personnel from government-funded research institutes and schools, except for some employees of large foreign companies. Even in the case of representative research achievements of National Centers, all 26 are owned by schools or government-funded research institutes. Therefore, it is difficult to efficiently execute the budget as there is no beneficiary to support the spread of performance. The reasons for the limited participation of SMEs can be attributed to the investment cost and lack of professional manpower. Since supercomputers have high initial investment costs, SMEs have limitations in building and operating their own infrastructure. This is why the National Center is currently building infrastructure and providing services to SMEs. However, participation in services provided by National Centers is also limited. It is difficult to secure supercomputer experts, who are classified as relatively high-level workers, for the use of supercomputers, and there is a burden of labor costs for the experts.

2) Limitation of small market size: The global HPC market size is about $32 billion by 2021. However, in the case of South Korea, it is less than 980 million dollars, so effective economies of scale do not apply, and it is an unstable market in which some foreign companies may monopolize due to the absence of competitive domestic companies. Even if domestic companies commercialize technology, technical competition is impossible due to the technological gap with leading companies such as Intel, AMD, and NVIDIA, and it is difficult to respond in terms of price. In cases where institutions with excellent technologies, such as domestic conglomerates and government-funded research institutes, transfer technology and try to commercialize it, the results are sold back to large corporations and government-funded research institutes.

3) Inadequate start-up support system: In the field of supercomputers, there are few achievements in commercialization technology development for the dissemination of research achievements. In the statistical data of the National Center, more than 60% of creative research (usage, as of 2022), which is in the basic source stage, is higher than research related to application, development, and commercialization for start-ups. In other words, there are few research projects and achievements in the commercialization stage where start-ups can generate sales and invest in technology development and manpower recruitment. In addition, in the case of other fields, programs such as technology exchange, financial support, and environment creation are being prepared for corporate support. However, in the case of the supercomputer field, there is no program to support the growth and development of start-up companies, and there is a shortage of dedicated departments and manpower in charge of the government.

4) Few institutions that possess source technology and transfer cases: In the case of source technology related to supercomputers, major companies in some countries, such as the United States, China, and Japan, preoccupy it. As a result of a survey on the status of patent index by country for US registered patents, in South Korea, the patent impact index was 0.05, lower than that of the US (1.06), and the market security index was 1.75, lower than the US (2.63), leaving behind a lot in both technological competitiveness and quality. The lack of source technology and the need to promote commercialization and commercialization limited to applied technology also become an economic barrier for companies. In addition, because of few cases of technology transfer and success, it makes difficult for start-ups to enter the market.

B. Suggestions for Improvement

In order to prepare an effective and efficient budget support institution for the dissemination of research results, the following institutional improvement plans are proposed.

1) Establishment of a national center performance management organization utilizing the execution system and legal functions: First, the current National Center is designated as the KISTI, and the KISTI is designated as an institution dedicated to R&D performance management and distribution according to Article 33 of the Enforcement Decree of the (National R&D Innovation Act)". Based on the (Act on Performance Evaluation and Performance Management of National R&D Projects), etc., KISTI can receive support for all or part of the expenses necessary for the operation as a dedicated agency, and the dedicated agency can receive papers and patents from a number of research institutes to promote joint utilization by registering and depositing research results. Therefore, the National Center will be able to benchmark the existing research performance utilization and management system as a dedicated agency, establish its own research performance utilization and management system, and effectively operate the organizational system and professional manpower. Even in the transfer of achievements, the KISTI, in accordance with Article 11 of the (Act on the Promotion of Technology Transfer and Commercialization), transfers technology developed in public research institutes to the private sector. Since an organization dedicated to commercialization must be established, it is possible to quickly form an organization. In addition, since the NTIS (National Science & Technology Information Service) platform, which is a performance information integration service that enables performance status, statistical data, and performance search, is established, accessibility and diversity of performance information can be increased by providing services related to this platform. Therefore, it is possible to establish efficient governance for performance utilization through functional convergence and linkage with the organization of the National Center and the dedicated agency, and through this, it will be possible to lower the entry barrier for private companies and continuously secure excellence and transfer cases.

2) Support for the whole cycle of commercialization of SMEs using the venture center: One of the ways to grow the market is to nurture companies that directly participate in the market. Although costly, it is the most effective way to rapidly nurture companies and expand the market. In South Korea, there are excellent benchmarking cases. In the case of the green technology industry, a number of start-ups and SMEs are being introduced by setting up an Environmental Venture Center in an environmental research complex composed of a number of related government departments, research and policy institutions. Economic and technical support is provided to resident companies at a relatively low cost. It is supported with full-cycle demonstration facilities for commercialization, such as inspection and certification for sales, expert mentoring for technology transfer items from public institutions related to R&D, testing and analysis, and certification, performance test. To date, the Environmental Venture Center has generated about four trillion won in economic effects and created about 9,400 jobs for 15 years.

In the field of supercomputers, it is possible to build an efficient venture center through a joint utilization system. First, companies participating in the joint utilization system can be institutionally introduced by designating them as Specialized Centers and Unit Centers. In the case of Unit Centers, they will be expanded step by step after the designation of Specialized Centers, so relevant contents can be reflected when establishing guidelines for the operation of Unit Centers. In addition, Specialized Centers and Unit Centers are supported with all expenses necessary for infrastructure construction and operation. Therefore, when a Specialized Center and a Unit Center are integrated into a venture center, it is possible to effectively reduce infrastructure construction space and common operating costs. It is possible to develop various convergence technologies by physically gathering supercomputer companies by field, and companies, not the government, can discover jobs for future with high added value. Due to the nature of supercomputers, face-to-face training and mentoring are possible even for small businesses that have difficulties in accessing and using services, so the use of services in the private sector is expected to increase significantly. However, in the case of supercomputers, research results and technology transfer items are lacking. Unlike environmental venture centers, Specialized Centers and Unit Centers should prioritize technology development and self-implementation, and induce exchange of research results between centers.

3) Establishment of unit center certification system reflecting preferential treatment for small and medium-sized enterprises: The Specialized Center is currently designated by the Ministry of Science and ICT. Currently, seven institutions in seven fields have been designated, and there are plans to continuously expand Specialized Centers for each field through the development of supercomputer-related fields in the future. However, in the case of the Unit Center, the relevant information for designation is not institutionally prepared. Unit Centers are classified and designated by sector like Specialized Centers, and operation plans such as infrastructure establishment and technology development linked to Specialized Centers must be established in the form of sub-organizations of Specialized Centers. Therefore, the budget size and payment method should be specified institutionally so that the budgetary part can be considered in the operation plan establishment stage. In addition, it is necessary to determine the appropriate size of the Unit Center through a demand survey by field so that many start-ups and small and medium-sized enterprises can participate. Also regulations to receive preferential treatment in the designation examination must be prepared. There is a need for a recovery support system to alleviate the burden of failure. Considering the market environment where the commercialization success rate of SMEs does not exceed 21%, it is necessary to prepare a safety device that can continue the benefits as a Unit Center [10].

4) Support lowering the threshold for commercialization through the introduction of the preferential purchase institution: The institution that preferentially purchases the research results of companies participating in the joint utilization system, such as the public institution preferential purchase institution, is essential for startups and SMEs to cross the threshold of commercialization. In the case of related companies in the supercomputer field, the initial threshold is even higher. Based on the supercomputers built in South Korea, the domestic market share is very low. Therefore, voluntary market entry by domestic companies should be pursued from a long-term perspective, and the government should provide sufficient time and resources for companies to grow by first raising the total purchase ratio in consideration of the characteristics of the sector.

VI. Conclusion

At the beginning of the establishment of the supercomputer joint utilization system, institutional gaps were discovered and improvement plans were drawn. In order to effectively perform the functions of the Specialized Centers constituting the joint utilization system, most of the support items are included through government laws and plans, such as the （Supercomputer Law）, “Innovation Strategy” and “Operation Guidelines”, etc. However, it was confirmed that the support system for securing a budget for the dissemination of research results was insufficient. As improvement plans for this, establishment of a National Center performance management organization using the execution system and legal functions of an institution dedicated to research achievements, support for the entire life cycle of commercialization of SMEs using venture centers, and establishment of a Unit Center certification institution that reflects preferential treatment for SMEs. Finally, support for lowering the threshold for commercialization through the introduction of a preferential purchase institution was proposed.

The government plans to designate a total of 60 Unit Centers by 2030 and prepare “Operating Guidelines” within this year. Using the results of this study, a specific support plan will be prepared so that a number of private companies can participate in the competition for designating Unit Centers, and preferential treatment for SMEs to participate will be reflected in the evaluation plan. In addition, in consultation with related ministries, the plan for the preliminary feasibility study project will be promptly promoted so that sufficient budget can be secured in advance.

Acknowledgment

This research was carried out through major project (K-23-L02-C03-S01) support of Korea Institute of Science and Technology Information.

References

[1] Shim. H.W., Jung. Y.H., Hahm. J.G. (2023). A Study on the Designation Institution for Supercomputer Specialized Centers in Republic of Korea, International Journal of Advanced Computer Science and Applications, 14(1), 306-312.

[2] S.Mitsuhisa, K. Yuetsu, T. Miwako, O. Tesuya, "Co-Design and System for the Supercomputer “Fugaku”" IEEE micro, vol. 42(2), pp. 26-34, 2021.

[3] G. I. Savin, B.M. Shabanov, P. N.Telegin, A. V. Baranov, "Joint Supercomputer Center of the Russian Academy of Sciences: Present and Future", LOBACHEVSKII JOURNAL OF MATHEMATICS, vol. 40(11), pp. 1853-1862, 2019.

[4] Ministry of Science and ICT (2021), National Supercomputing Innovation Strategy.

[5] "Supercomputer Specialized Center Operation Guidelines", Republic of Korea, 2021.

[6] Lee. H.J., Choi. Y.K., Park. J.S.(2018), An Economic Analysis on the Operation Effect of Public Supercomputer, Journal of the Korea Industrial Information Systems Research, 23(4), 69-79.

[7] Kim. J.S., Seo. D.W., Kim. E.J.(2015), supercomputing industry support policy and business trends, CAD/CAM review, 21(1), 32-37.

[8] Nam. D.Y., Lee. P.W.(2016), Development and establishment of national supercomputing infrastructure to build future supercomputing ecosystem, Communications of the Korean Institute of Information Scientists and Engineers, 34(2), 22-30.

[9] Huh. T.S., Jung. Y.H., Ko. M.J.(2021), Legal Institutional Improvement for Activating National Supercomputing Ecosystem, The Journal of the Korea Contents Association, 21(2), 641-651.

[10] Kim. S.W., Kim. J.W.(2020), Innovative growth improvement plan for SME R&D support, Stepi insight, (249), 1-40

Efficient Handwritten Signatures Identification using Machine Learning

Ibraheem M. Alharbi
MIS Dept., College of Business, University of Jeddah, Jeddah,
Kingdom of Saudi Arabia (KSA)

***Abstract*—Any agreement or contract between two or more parties requires at least one party to employ a signature as evidence of the other parties' identities and as a means of establishing the parties' intent. As a result, more people are curious about Signature Recognition than other biometric methods like fingerprint scanning. Utilizing both Fourier Descriptors and histogram of oriented gradients (HOG) features, this paper presents an efficient algorithms for signature recognition. The use of Local binary patterns (LBP) features in a signature verification technique has been proposed. Using morphological techniques, the signature is encapsulated within a curve that is both symmetrical and a good match. Measured by the frequency with which incorrect patterns are confirmed by a given system, false acceptance rate (FAR) provides an indication of the effectiveness and precision of the proposed system. Using a local dataset of 60 test signature patterns, this investigation found that 10% were incorrectly accepted for FAR of 0.169. Experiments are conducted on signature photos from a local dataset. Verification of signatures has previously made use of KNN classifier. KNN classifier produced higher FARs and recognition accuracies than prior techniques.**

Keywords—K-nearest neighbor; histogram of oriented gradients; local binary patterns; false acceptance rate; Fourier descriptors

I. Introduction

Identifying and authenticating individuals has developed into a crucial step in the provision of services in commercial and government institutions, and is also an important aspect of maintaining law and order. Biometric data from the individual being verified. The two Greek words for life and measurement provide the basis of the English word "biometrics" (to measure). Because of their uniqueness, the biometric features listed in [1] can be used to confirm or authenticate an individual's identity. Because the distinguishing trait in question is intrinsic to the individual being identified, biometric identification is more fool proof. As a result, it's extremely difficult to give away, swap, or steal from another person.

Physiological features biometrics and behavioural characteristics biometrics are the two most common types of biometrics identification. Fingerprinting, retinal scans, and handprint scans are all examples of physiological biometrics. Checking a person's signature or voice is examples of behavioural biometrics. Biometric systems are "automatic methods of identifying or validating the identity of an individual based on physiological or behavioural features" [2]. A biometric system can function in verification mode or identification mode, depending on the requirements of the application. An individual claims their identity, and the system checks to see if that assertion is true. In order for a claim to be considered "genuine," there must be a significant degree of similarity between the user's input and the template of the claimed identity. If not, the user's claim will be denied and they will be labelled a "fraud." When performing Identification, the biometric system compares the user's input to all of the stored templates and returns the identification of the individual whose template is most similar to the user's input [3]. The system will typically return a refuse decision, indicating that the user presenting the input is not one of the enrolled users, if the highest similarity between the input and all the templates is less than a defined minimum level. Accordingly, the ratio of matches in an identifying system is 1 to N.

The need for a reliable method to verify and authenticate individual signatures arises from the widespread use of signatures in financial, economic, and legal operations. Static and dynamic digital handwritten signature authentications are the two most used methods. One of the most common types of static is a visual comparison between two scanned signatures or between a scanned signature and an ink signature [4]. The signer's signature is captured in the form of coordinate values from the signing device, which is subsequently used for dynamic digital handwritten signature authentication. One part of signature recognition and verification is determining whether or not the signature is authentic, and the other part is determining who the signature belongs to. Image processing and feature extraction methods are utilised throughout the static signature verification process [5].

In order to determine if an input signature is real or forged, matching relies on authentication (false signatures of an individual). Two stages make up this section: Verifying a person's identification using the first-signature part's database and other identifying information. Identification: Basically the input signature image for each subject is compared with entire of the database i.e. with samples from all subjects in the database. Verification: Which entails the comparing of the input signature image to samples of the same subject's signature. Confirming a person's identity is the primary focus of this procedure.

Signature authentication or verification approach may be writer dependent or writer independent. Following this process allows for a writer-independent approach. Mean distances between authentic and forgery classes, as well as forgery and known classes, are used to calculate prior parameter distributions for the respective means. Posterior class probabilities for the two classes, authentic and forged signatures for a given author, are calculated. Next, the probabilities of each group are compared and select the group with the higher probability based on a signature that is under scrutiny [6].

In case of writer dependent technique, an individual classifier is constructed for each user using his enrolment samples. During verification, only query signature samples are analysed by the classifier [7]. The success of these systems is obviously dependent on having a large enough sample size with which to train their classifiers.

A. Objective of the Research Work

By comparing the signature to other examples, the identity of the signer is determined. Learning-based signature recognition systems necessitate a sizable training set, ideally including examples from the vast majority of the intended users. Based on the published works, it is clear that several researchers have created their own databases to test the reliability of their signature verification or identification systems. Multiple static signature databases have also been made public for study. Signatures from a variety of countries and regions, including Malaysia, Spain, China, the Netherlands, Tunisia, etc., are used to test out the methodologies suggested in the literature. However, they don't accurately reflect the diversity of Arabic signatures. Also, most Arabic signatures are written in regional scripts other than English script, hence the approaches may not produce high recognition accuracy for these signatures. That's why it was important to create a regional database of signatures for the area and develop effective algorithms for checking the offline signatures so that they can be recognised with high precision.

II. Previous Work

Using global, directional, and grid aspects of signatures, authors [8, 9] suggested a solution for an off-line signature verification and identification system. The signatures were validated and categorised using a system called a Support Vector Machine (SVM). In [10], researchers analysed two popular Tablet PCs for use in signature verification tests. Using a database of 3000 signature photos, the authors report on experimental evaluations of authentication performance. Current best practises for authenticating digital signatures are summarised in [11]. With the help of Adaptive Feature Threshold, authors [12] have presented a person-dependent off-line technique for verifying signatures (AFT). This method improves on the practise of transforming a signature's basic feature into a binary feature vector in order to increase its representational resemblance to training signatures. They employed a hybrid of spatial pyramid and equi-mass sampling grids to enhance the representation of a signature dependent on gradient direction. They employed a DWT and a graph matching technique during the classification phase. Using graph matching to compare signatures and the Euclidean distance to determine how dissimilar they are, researchers. [13] present a Cross-Validation Technique for Graph Matching based Off-Line Signature Verification (CSMOSV). In [14], authors propose using image registration to identify and authenticate off-line Persian signatures.

As for the matching, they employed Euclidean distance and DWT to extract features. The approach, however, is language specific. In [15], authors offer an offline signature verification method that uses machine learning. Directional Gradient Density characteristics have been presented for competent forgery verification. A grid-based solution employing global characteristics for offline signature verification was reported by researchers in [16].

Authors [17] describe a system that requires fewer characteristics through the use of sub-pattern analysis, which in turn results in faster responses in real-time situations. A multilayer weighted fuzzy classifier that fuses match scores via selection priority has been developed to fully leverage the potential of two sets of characteristics. Multiple features for biometric recognition systems were proposed by researchers in [18]. Signature feature extraction, which takes in data from twelve various angles, was proposed using Rotated Complex Wavelet Filters (RCWF) and Dual Tree Complex Wavelet transform (DTCWT). In [19], authors proposed the issue of handwriting biometrics and presented a method for validating handwritten signatures with an ANN. In [20], researchers offer a scale- and rotation-invariant method for signature recognition based on the extraction of invariant rotation invariant texture features (sub-uniform local binary patterns) from each of an image's 12 blocks. Verification makes use of DCT coefficients. In [21], the state-of-the-art of offline signature recognition using Computer Vision is outlined. The authors go on some of the latest developments and areas for further study, including the creation of synthetic signatures, temporal drifting, identifying forgers and impostors, and dealing with scenarios involving more than one language. The Support Local Binary Pattern (SLBP) characteristics were proposed by authors in [22] for use in offline signature verification. Several writers use LBP variants in the context of signature verification.

To quickly and accurately verify signatures with minimal effort, authors demonstrate a technique that she has developed. For feature extraction, the Discrete Wavelet Transform (DWT) with Haar wavelets is primarily studied, both for global features and grid features, in [23]. Offline verification of signatures using a small number of basic geometric features was presented by authors in [24], the features Area, Euler's Number, Eccentricity, Standard deviation, Centroid, Skewness, Kurtosis, and orientation are employed. The artificial Immune Recognition System (AIRS) and ANN employed in the verification step are supported by a novel offline signature verification technique presented by researchers in [25].

From the reviewed literature, it is clear that many different approaches have been proposed for signature verification, and that experiments on both local datasets and standard datasets, such as MCYT have been used to provide verification results. It is important to note that many of the techniques described are language dependent, meaning that their effectiveness is limited to signatures written in those languages. Recognizing that offline signature verification is a challenging problem with room for further investigation, we have proposed the development of efficient verification systems that improve performance for signatures written in both local languages and other languages defined in a standard database of signature images [26].

It is well-documented that various feature extraction strategies have been developed for author-independent signature verification. Using a combination of multiple feature extraction, dichotomy transformation, and boosting feature selection, authors [27] present a writer-independent method. They used feature extraction methods that worked at various scales before applying the Dichotomy Transformation, which turned the problem into a two-class one. Finally, boosted feature selection is used to narrow down the training set to the most important features.

A writer-independent method for authenticating handwritten signatures was proposed by researchers in [28]. The primary segments' curvature was used to generate graphometric feature sets, and this was done virtually using Bezier curves. The robustness against forgeries was bolstered by employing an ensemble of classifiers.

Using the dichotomy transformation and an SVM writer-independent classifier, authors in [29] explored the usage of these deep convolution neural network (CNN) features for writer-independent offline signature verification. Experimental results on the Brazilian and GPDS datasets demonstrated that the proposed strategy outperformed its competitors [30-31].

III. The Proposed Work

If a separate model is trained for each user, the system is more robust in signature verification. Author-specific signature validations are more precise. During the training phase, authentic signatures from a given topic are considered positive examples, whereas signatures from other users are considered negative examples. Each user has their own binary classifier that they've been training. As the number of participants in the study grows, the complexity and cost of maintaining the system rises in tandem. Writer-independent signature verification techniques, on the other hand, can be used to categorise the signatures of any user in the dataset. When training a model in this setting, it is done so with all of the subjects combined into a single one. Offline and online automatic signature verification systems are available, depending on the preference of the author. Both cases involve training a classifier, in this case a supervised classifier, to verify signatures using a small subset of data drawn from a more complicated distribution than the whole. The classifier is taught using a dataset of authentic signatures that is large enough to be representative of all of the valid users that have registered for the verification service.

This paper proposes a method that allows writers to be creative while remaining untethered to any one particular platform. Extracting features involves estimating a continuous curve that best matches the signature based on Identification utilising Fourier Descriptors (FD). A closed boundary is drawn around the entire signature image, and then FDs are calculated as features of the boundary. KNN classifiers take these features as input and use them to determine whether or not a signature belongs to a certain person based on similarities between the input features and the features already stored in the database. Fig. 1 shows a general block diagram of the proposed method. The details of the method will be shown in the next subsections.

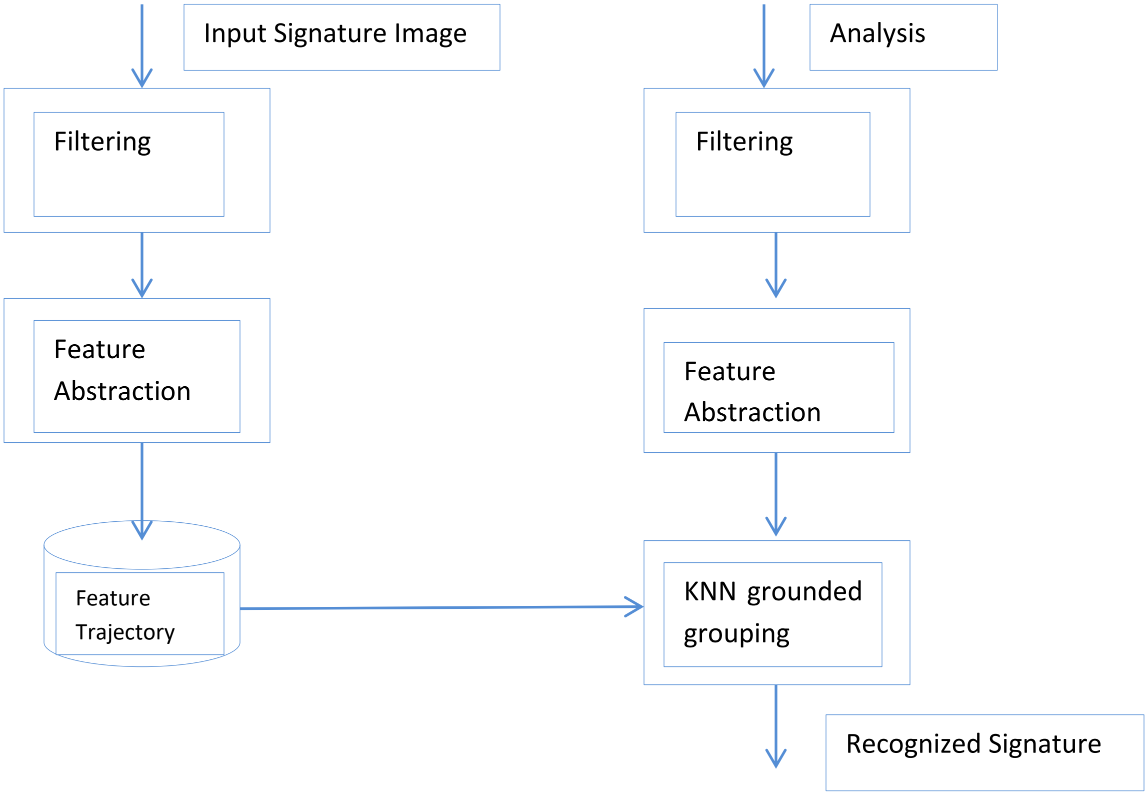


Fig. 1. The proposed block diagram.

A. Filtering

Some examples of pre-processing operations include noise reduction, binarization, rotation normalisation, resizing, and thinning. Scanning can create grey-scale signatures, although these may have unwanted extra dots. In order to get rid of the extraneous dots, median filters are applied to the gathered signature image (salt and pepper noise). Processing a grey-scale or colour image is more time-consuming than a binary one. The image is typically depicted in monochrome.

Pixels minimization is based on thresholding. Otsu's binarization strategy is suggested for use [31]. A signature's angular shifts over time are mitigated by rotation. The axes of inertia of all signatures are arranged horizontally. One approach to alignment involves finding the border of the signature with an edge detector, then thinning (or skeletonizing) it, applying the Radon transform, and determining the counter clockwise rotation angle. The signature's crookedness can be fixed by turning it around in a clockwise direction.

Signatures with a similar form but vastly different sizes have a low similarity score. This is because of the normalisation effect, which nullifies this. All normalised signatures provide comparison between reference (first-phase generated) and test samples (input signature picture classified as authentic or counterfeit). A bounding box is applied to the signature in order to normalise its proportions by erasing the surrounding space [32], [33].

The size of normalised images varies widely. Resolutions ranging from 40x40 to 200x300 were selected at random by the researchers. The width-to-height ratio (aspect ratio) is taken into account throughout the resizing process. The trademark image is reduced down to a single pixel in thickness. The operation remembers nothing beyond the bare minimum of the signature, even though this is by no means optional. It lessens the signature image should first have any unnecessary pixels removed before feature extraction can begin. The preliminary analysis is depicted in Fig. 2.

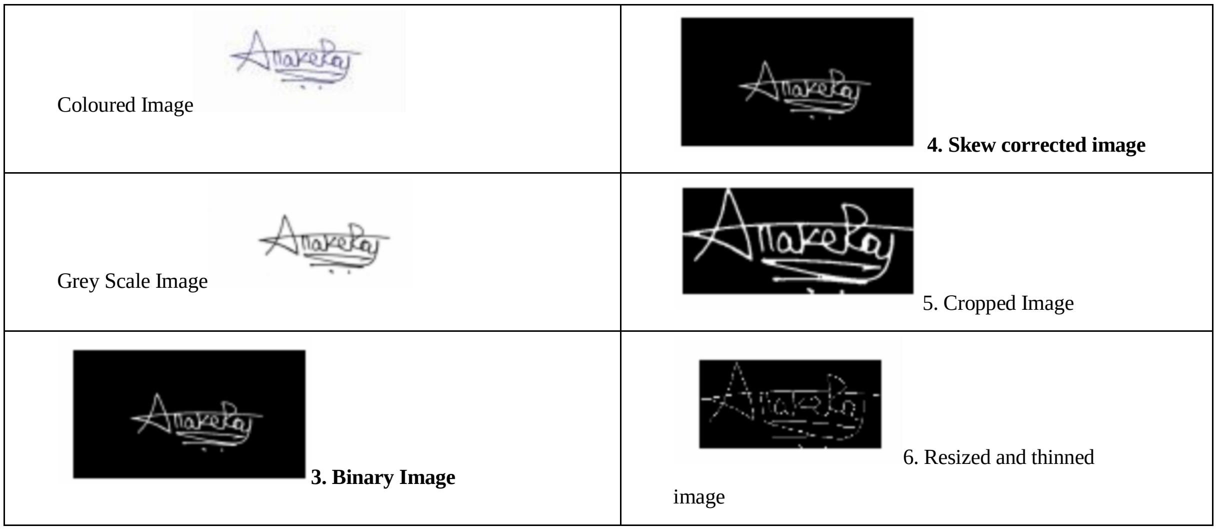


Fig. 2. Pre-processing of offline signature images.

B. Feature Abstraction

Shape analysis makes extensive use of Fourier Descriptors (FDs). Fourier descriptors of a shape are the coefficients of the Fourier transformation. The form of the object is reflected in these characterizations, which are expressed in terms of frequency. To begin, you'll need the N points that make up a region's boundary by sampling from the N pixels that make up the boundary. Just follow the perimeter around in a counter clockwise direction to get the job done. In the complex plane, the ordinate represents the imaginary axis and the abscissa represents the real axis, as seen in Fig. 3. Every point on the outline of the object has an x-coordinate pair of the form (Ak, Bk), where, 0 < k ≤ N-1.

The boundary is completely described by a set of coordinates. Coordinate series can then be used to describe the contour.

$$P(K) = [A(k), B(k)], \text{ for k=0, 1, 2…..N-1} \quad (1)$$

Where, $P(k) = A(k) + jB(k)$

The DFT of P(K) is given as:

$$X(u) = \frac{1}{N}\sum_{k=0}^{N-1} \quad P(k)e^{\frac{-j2\mu ku}{N}} \quad \text{for K= 0,1,2,….N-1} \quad (2)$$

The transform typically produces a large number of coefficients, but typically only a small subset of those coefficients are necessary to represent the essential properties of the shape. In this way, the FD's normalised magnitude can remove dependence on the size of the shape being examined. Information regarding the shape's finer intricacies can be found in the high frequency descriptors, while details about the shape's global or overall characteristics can be found in the low frequency descriptors. The size of the Fourier descriptors for these purposes must be increased accordingly.

Indexing forms have been drastically cut down. Since there are so many possible words, a figure out of the number of FD should be determined. Due to its position-only dependence, the DC component is useless for characterising shape and is therefore omitted. To normalise the scale, the second magnitude value is divided by each of the other descriptors' magnitude values.

In order to implement FD, the signature image will need to undergo a boundary tracing. Optimal FD for signature shape recognition are found through trial and error. Since a signature may include more than one part, we often encompass the whole thing within a closed curve that best represents it. The enclosed, closed curve is generated using morphological processes. The resulting curve is successfully computing FDs for form recognition, as they are signature-specific. Fig. 3 shows several example contour drawings.

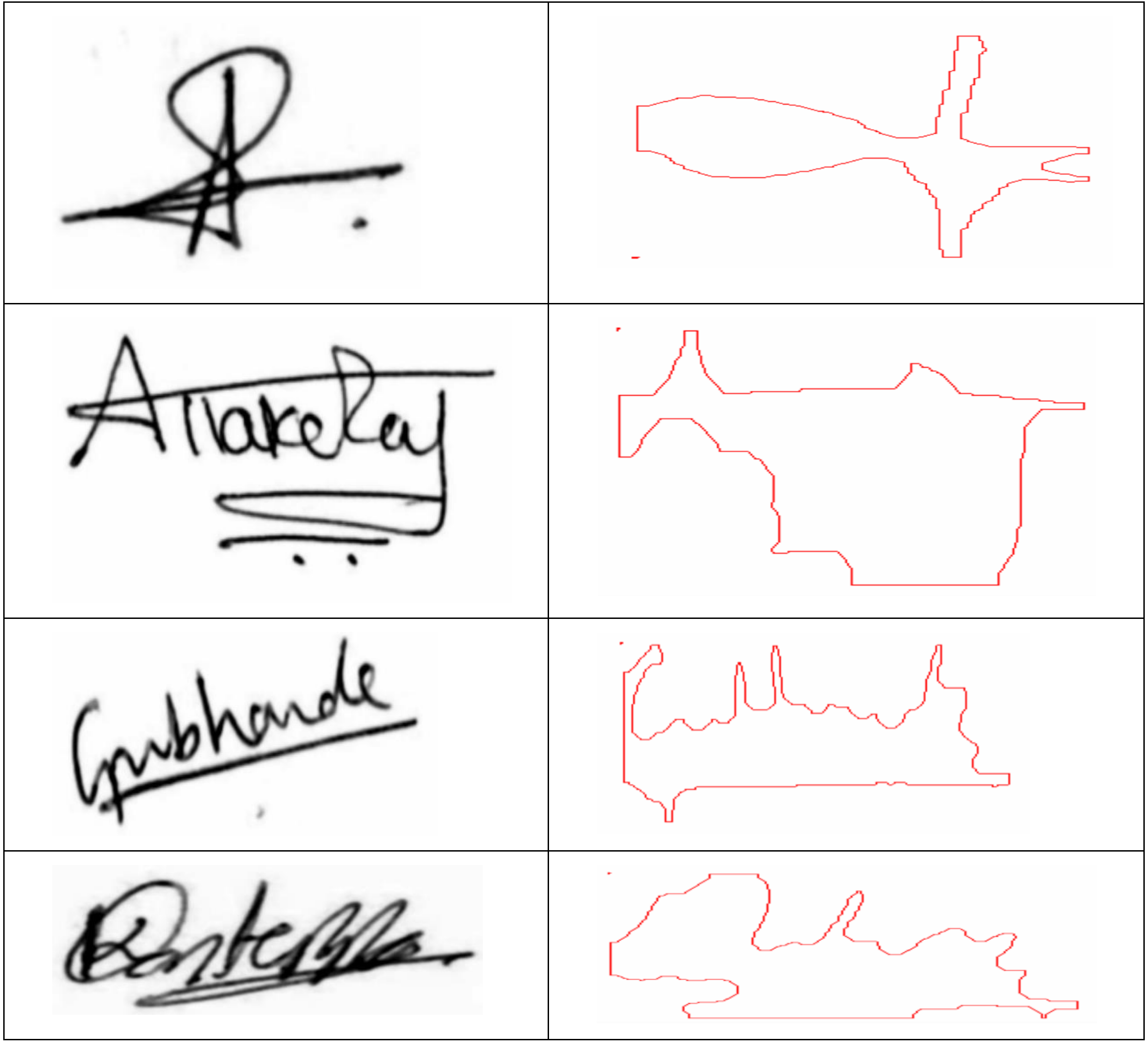


Fig. 3. Sample signature images and their enclosing curves.

C. Recognition

One common non-parametric classifier is the K-Nearest Neighbour (KNN) classifier. Classifier that uses the frequency of an unknown pattern's neighbours to estimate its posterior probability. When compared to other supervised statistical pattern recognition approaches, the KNN rule produces consistently good performance while making no a priori assumptions about the distributions from which the training examples are obtained.

To test FD's usefulness as feature vectors for signature image recognition, the KNN classifier has been selected. To do this, KNN classifier is used to compare the array of templates for the target signature to the arrays of all the other signatures in the database, with the goal of appropriately assigning the signature to a specific signature in the database based on the minimum distance attained. Specifically, the Euclidean distance is used for the calculation.

Following is a description of how to calculate the K-nearest neighbours.

Find the K value, which corresponds to the number of nearest neighbours. In most cases, an odd number (such as 1, 3, 5, etc.) is selected for K.

In order to do this, the distance between the query instance and all of the training samples is computed. A distance criterion, such as the Euclidean distance, can be used to calculate the distance between two points.

Find the closest neighbours by sorting the distances from closest to farthest.

Collect the neighbourhood's classification or tag. In most cases, labels are merely connected to the training sets.

In this method, the prediction value (label) of the query instance is determined by taking the simple majority of the category of the nearest neighbours.

IV. RESULT AND DISCUSSION

The goal here is to evaluate and contrast how various methods of approaching the problem fare. Method of identifying signatures is proposed. Ten participants from various occupations were selected at random. Variations, if any, in signatures were recorded by collecting them on white A4 paper at various intervals. Each participant signed 16 consent forms. A flatbed scanner is used to scan the signature pages at 300 dpi in grey-scale, and then a software is used to clip out each signer.

Profile work can be done in both the horizontal and vertical planes. From a total of 16, only 10 signatures were used for training and the remaining six were used for testing. This meant that there were 100 signatures in the training set and 60 in the test set. Training samples had 64-dimensional feature descriptors (FDs) generated for them and labelled with 10 topic labels for every 100 features.

Dimensions, each of which is a 64-dimensional vector of FDs. Label-free 64-dimensional FDs were calculated for each of the test samples. Using the generated FDs as features to describe the signature images, the training and test vectors were fed into a KNN classifier. To keep track of which feature vector xr from the signature picture corresponds to which reference signature Sr, a training phase is employed, when the system acquires its foundational knowledge. In recognition mode, the picture of the suspect signature Sq is displayed to the system, and the feature vector xq is fed into the KNN module alongside the reference set xr of signatures from users who have opted into the knowledge base.

In both cases, the outcomes are enhanced when K=1 is used. Using the MCYT database photos, the suggested technique performs better at K=1 than it does with images taken from the local dataset. The incorrect classification may be traced back to the fact that the enclosing boundary for that signature instance was different from the samples used for the rest of the image, which meant that the pre-processed image had gaps that weren't filled properly.

The classifier results are presented in Table I. The comparison of these subjects performance with different values of KNN classifier is plotted in Fig. 4.

TABLE I. RECOGNITION RESULTS USING KNN CLASSIFIER

Subjects	**No. of assessment**	**Recognition Local Database**		**MCYT Recognition**	
		K=1	**K=3**	**K=1**	**K=3**
1	10/6	6	6	6	5
2	10/6	6	5	5	6
3	10/6	5	6	6	5
4	10/6	6	6	5	5
5	10/6	5	5	5	5
6	10/6	5	4	5	6
7	10/6	4	2	4	5
8	10/6	5	4	5	5
9	10/6	4	5	6	5
10	10/6	6	6	6	6
Recognition accuracy in %		88.35	82.35	89.61	86.62

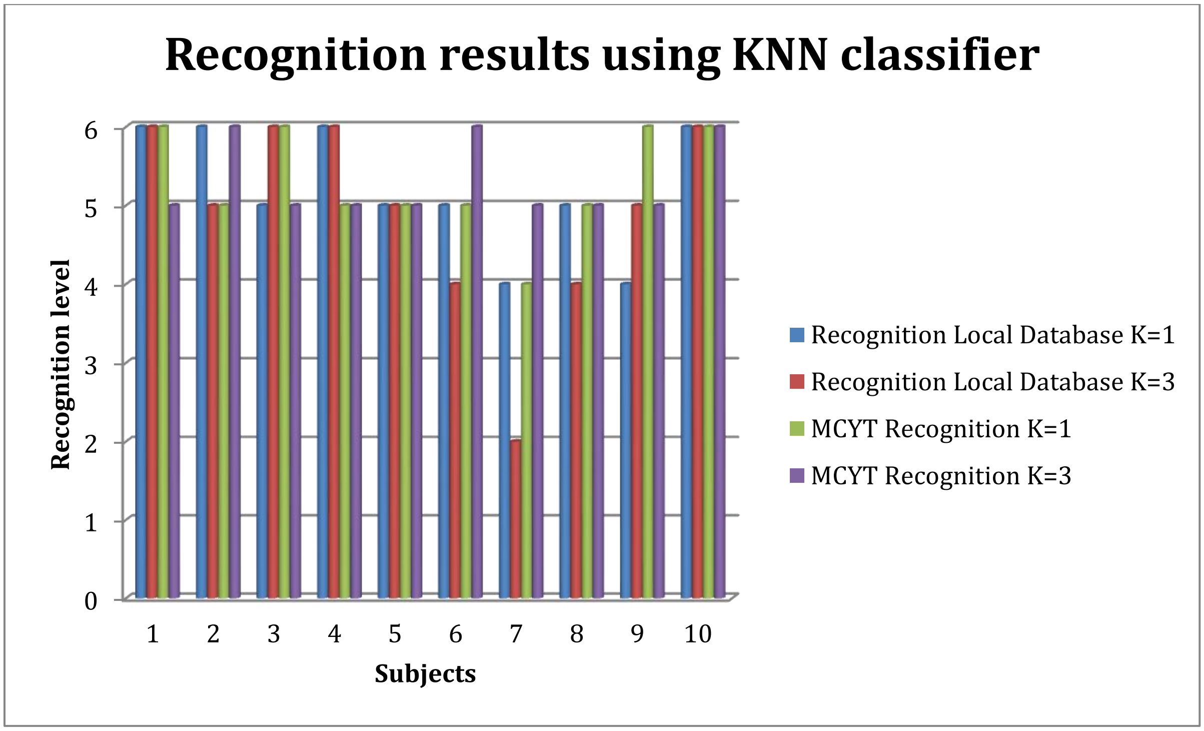


Fig. 4. Recognition results comparison using KNN classifier.

Form this comparison plot, we can conclude that KNN classifier with K=1 performs better and accuracy improved.

It is possible for a biometric security system to accept an access attempt from an unauthorised user; this possibility is quantified by the false acceptance rate (FAR). The false acceptance rate (FAR) of a system is sometimes described in terms of the fraction of identification efforts that result in a false positive.

$$false\ acceptance\ rate\ (FAR) = \frac{number\ of\ false\ acceptence}{number\ of\ identification\ attempts} \tag{3}$$

The accuracy and FAR for different data-set and for different value of K is presented and compared in Table II.

TABLE II. ACCURACY AND FAR VALUE OF CLASSIFIER

Data-set	Classifier	Accuracy	FAR
Local Data-set	KNN for K=1	88.35	0.1672
	KNN for K=3	82.35	0.2187
MCYT Data-set	KNN for K=1	89.61	0.1772
	KNN for K=3	86.62	0.2192

From this Table II, we can conclude that MCYT dataset and KNN classifier with K=1 perform best for signature identification.

V. CONCLUSION AND FUTURE SCOPE

A. Conclusion

The writer's characteristic patterns of behaviour during the signature-making process are retrieved as distinct features and then saved and compared in a Signature Recognition system. While signature verification seeks to confirm or reject a particular sample, signature recognition seeks to identify the author of that sample. Through the use of Fourier Descriptors and HOG features, we have presented efficient algorithms for signature recognition. Using LBP characteristics, a signature verification technique has been presented. An effective method for signature identification is given in this study. Features are extracted using FDs, and recognition is accomplished with KNN. The acquired findings validate the usefulness of the proposed method. In order to proceed with the recognition process, authentication must first take place. However, we found that there is a substantial class overlap in authentication when using the proposed technique, particularly between the confined boundaries of authentic signatures and forged ones are often confused. The challenge is deciding what level of confidence to assign to the recognised signature. Accuracy and false alarm rate (FAR) comparisons are shown with the experimental results. Functioning Area Ratio (FAR) is a tool for gauging and assessing the effectiveness reliability of a suggested system by counting the number of times incorrect patterns were confirmed using that system. Ten incorrectly accepted patterns out of 60 in the test signature dataset resulted in a FAR of 0.1672 and an accuracy of 88.35% for K=1.

B. Future Work

FDs and HOG characteristics, which are used in the signature identification method, broadened to include signature authentication. Signature images written in different Arabic and non-Arabic scripts can be studied using the methods provided in this experimental results have validated the effectiveness of the proposed system. While each of the proposed features may improve performance individually, they can be coupled for even greater gains. The performance can be enhanced by using an ensemble of classifiers.

REFERENCES

[1] lyani Mali, Samayita Bhattacharya,"Comparartive Study of Different Biometric Features",International Journal of Advanced in Computer and Cmmunication Engineering Vol.2, Issue 7, July 2013.

[2] Mary Lourde R and Dushyant Khosla," Fingerprint Identification in Biometric Security Systems", International Journal of Computer and Electrical Engineering, Vol. 2, No. 5, October, 2010.

[3] K P Tripathi, "International Journal of Computer Applications", (0975–8887) Volume 14– No.5, January 2011.

[4] K. R. Radhikas. V. Sheelag. N. Sekhar, "Off-Line Signature Authentication using Radial Basis Function", International Journal of Pattern Recognition and Artificial Intelligence, March 2011, Vol. 25, No. 02 : pp. 207-225.

[5] Dominique Rivard · Eric Granger, Robert Sabourin, "Multi-feature extraction and selection in writer-independent off-line signature verification", IJDAR (2013) 16:83–103, Springer-Verlag 2011.

[6] M.S.Shirdhonkar et.al. "Off-line Handwritten Signature Identification using Rotated Complex Wavelet Filters" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011 ISSN (Online): 1694-0814www.IJCSI.org 478.

[7] Supinder Singh, and Amandeep Kaur, "Off-Line Signature Verification using Sub uniform Local Binary Patterns and support vector machine ",Int'l Conf. on Chemical Engineering & Advanced Computational Technologies (ICCEACT'2014) Nov. 24-25, 2014 Pretoria (South Africa).

[8] Moises Diaz-Cabrera Aythami Morales Miguel A. Ferrer (2014) "Emerging Issues For Static Handwritten Signature Biometrics", Advances in Digital Handwritten Signature Processing: pp. 111-122

[9] P. Vickram, Dr. A. Sri Krishna and D.Swapna ,Offline Signature Verification using Support Local Binary Pattern, International Journal of Artificial Intelligence and Applications (IJAIA), Vol. 7, No. 6, November2016.

[10] Sumedha Tanajirao Panchal "Offline signature verification based on geometric feature extraction using artificial neural network", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 13, Issue 3, Ver.III (May. - June.2018), PP 53-59.

[11] Jasmeet Kaur, Dr. Reecha Sharma, "Offline Handwritten Signature Verification method based on Artificial Immune Recognition System and Artificial Neural Network", ISSN (e): 2250 – 3005 || Volume, 07 || Issue, 07|| June – 2017 || International Journal of Computational Engineering Research (IJCER).

[12] Victor L. F. Souza, Adriano L. I. Oliveira, Robert Sabourin, "A writer independent approach for offline signature verification using Deep Convolutional Neural Networks features", BRACIS 2018.

[13] Manato Hirabayashi, Shinpei Kato, Masato Edahiro, and Kazuya Takeda, Taiki Kawano and Seiichi Mita , "GPU Implementations of Object Detection using HOG Features and Deformable Models", IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), IEEE, 2013.

[14] Miguel A. Ferrer, J. Francisco Vargas, Aythami Morales, and Aaron Ordonez "robustness of offline signature verification based on gray level features", IEEE Transactions on Information Forensics and Security, Vol.7, No.3, June2012.

[15] Jagtap, A.B.; Sawat, D.D.; Hegadi, R.S.; Hegadi, R.S. Verification of genuine and forged offline signatures using Siamese Neural Network (SNN). Multimed. Tools Appl. 2020, 79, 35109–35123.

[16] Mshir, S.; Kaya, M. Signature recognition using machine learning. In Proceedings of the 2020 8th International Symposium on Digital Forensics and Security (ISDFS), Beirut, Lebanon, 1–2 June 2020; pp. 1–4.

[17] Poddar, J.; Parikh, V.; Bharti, S.K. Offline signature recognition and forgery detection using deep learning. Procedia Comput. Sci. 2020, 170, 610–617.

[18] Cruz, S.; Paulino, A.; Duraes, J.; Mendes, M. Real-Time Quality Control of Heat Sealed Bottles Using Thermal Images and Artificial Neural Network. J. Imaging 2021, 7, 24.

[19] Kao, H.H.; Wen, C.Y. An offline signature verification and forgery detection method based on a single known sample and an explainable deep learning approach. Appl. Sci. 2020, 10, 3716.

[20] Hirunyawanakul, A.; Bunrit, S.; Kerdprasop, N.; Kerdprasop, K. Deep learning technique for improving the recognition of handwritten signature. Int. J. Inform. Electron. Eng. 2019, 9, doi:10.18178/ijiee.2019.9.4.709.

[21] Anisimova, E.S.; Anikin, I.V. Finding a rational set of features for handwritten signature recognition. In Proceedings of the 2020 Dynamics of Systems, Mechanisms and Machines (Dynamics), Omsk, Russia, 10–12 November 2020; pp. 1–6.

[22] Zhou, P.; Feng, J.; Ma, C.; Xiong, C.; Hoi, S.C.H.; et al. Towards theoretically understanding why sgd generalizes better than adam in deep learning. Adv. Neural Inf. Process. Syst. 2020, 33, 21285–21296.

[23] You, K.; Long, M.; Wang, J.; Jordan, M.I. How does learning rate decay help modern neural networks? arXiv 2019, arXiv:1908.01878.

[24] Parviainen, P.; Tihinen, M.; Kääriäinen, J.; Teppola, S. Tackling the digitalization challenge: How to benefit from digitalization in practice. Int. J. Inf. Syst. Proj. Manag. 2017, 5, 63–77.

[25] Brennen, J.S.; Kreiss, D. Digitalization. InThe International Encyclopedia of Communication Theory and Philosophy; Wiley: Hoboken, NJ, USA, 2016; pp. 1–11.

[26] Lampert, C. Ramping up: Evaluating large-scale digitization potential with small-scale resources. Digit. Libr. Perspect. 2017, 34, 45–59.

[27] Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Offline handwritten signature verification—Literature review. In Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–8.

[28] Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognit. 2017, 70, 163–176.

[29] Shang, Y. A combinatorial necessary and sufficient condition for cluster consensus. Neurocomputing 2016, 216, 611–616.

[30] [30]. Bansal, A.; Nemmikanti, P.; Kumar, P. Offline signature verification using critical region matching. In Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia, Hainan, China, 13–15 December 2008; Volume 3, pp. 115–120.

[31] Daramola, S.A.; Ibiyemi, T.S. Offline signature recognition using hidden markov model (HMM). Int. J. Comput. Appl. 2010, 10, 17–22.

[32] Rajput G. G., Patil P. k.; Writer Independent Offline Signature Recognition based upon HOGs Features; IJEE, vol. 9, 01, 2017, pp.59-67.

[33] Khalajzadeh, H.; Mansouri, M.; Teshnehlab, M. Persian signature verification using convolutional neural networks. Int. J. Eng. Res. Technol. 2012, 1, 7–12.

[34] K. Kancharla, V. Kamble and M. Kapoor, "Handwritten Signature Recognition: A Convolutional Neural Network Approach," 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, 2018, pp. 1-5, doi: 10.1109/ICACAT.2018.8933575.

An AHP based Task Scheduling and Optimal Resource Allocation in Cloud Computing

Syed.Karimunnisa[1], Yellamma Pachipala[2]
Research Scholar[1], Associate Professor[2]
Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation Vaddesvaram,
AP, India-522502

***Abstract*—Cloud systems by virtue characterize ultimate resource utilization with ever evolving user requirements facilitating adaptivity. With a scope of enhancing the QoS needs of user applications, numerous factors are considered fortunning among which Task scheduling promises to grab focus. The Task Scheduling mechanism ascertains improvement by distributing the subtasks to specific set of resources pertaining to prevailing Quality models. The work emphasizes the need for effective task scheduling and optimizing resource allocation by modelling a modified AHP (Analytical Hierarchy Process) driven approach. The proposed method guarantees the functionality in two phases pertaining to Task ranking and pipelined with Optimized scheduling algorithms resulting in maximization of resource utilization. The former phase of task ranking is aided by improved AHP with substantial usage of fuzzy clustering followed by an enhanced CUCMCA (Chimp Updated and Cauchy Mutated Coot Algorithm) algorithm for optimal resource allocation of cloud applications. The contributed model promises leveraged performance of 32% for memory usage, 33.5% for execution time, 29% for makespan and 18% for communication cost over pre-existing conventional models considered.**

Keywords—Task scheduling; AHP; TS; QoS; optimization; CUCMCA

NOMENCLATURE

Abbreviation	Description
QoS	Quality of Service.
RM	Resource management.
CESS	Cross-Entropy based stochastic scheduling.
OP-MLB	Online VM Prediction based Multi-objective Load Balancing.
MESA	Migration enabled scheduling algorithm.
MINLP	Mixed integer non-linear programming.
LCS	Learning classifier systems.
SLA	Service level agreement.
VM	Virtual machine.
HASRA	Hotspot aware server relocation algorithm.
PM	Physical machine.
HAWDA	Hotspot adaptive workload deployment algorithm.
CSO	Cat swarm optimization.
RMFW	Resource Management Framework for multiple online Scientific workflows.
HGCSBAT	Hybrid cat swarm bat algorithm.
TS	Task scheduling.

I. INTRODUCTION

Cloud computing is a flexible strategy for exchanging distributed services and resources with the user wherever and anytime they require [1]. Because of its scalable user pool of services and resources, cloud computing has become more popular in recent years. This is because it allows users to freely control their consumption and only pay for the cloud resources they actually utilize [10] [28]. Google Compute Engine, Rackspace Cloud, and Amazon EC2 are just a few of the commercial cloud computing systems that have recently entered the market [2] [15]. Moreover, for hosting and delivering software solutions for many industrial applications, cloud computing has established itself as a dependable, affordable, and scalable service option [14] [12]. Still, a cloud must have sufficient capacity to meet peak user demand in order to uphold user expectations for QoS [19] [16].

Resource usage, server power consumption, and storage all play significant roles in the cloud environment, making resource management essential. While provisioning and allocating cloud resources, availability was frequently used as the determining factor without considering the other essential factors like resource utilization or the server's thermal properties [9]. A cloud system's ability to manage its resources autonomously and adaptively based on the workload changes is known as autonomous RM [20]. Elastic resource management encompasses a wide range of processes, including balanced virtual machine and application scheduling, server over/under-load control utilizing VM migration, etc. [3] [17]. One or more physical machine resources, including CPU, memory, I/O, and network bandwidth, may be overloaded due to the increased load of virtual machine operations [5] [8].

The scheduling solution was optimized in terms of each metric specified in the QoS model using a QoS-driven CESS method [4]. It has been suggested to use an OP-MLB system, which combines a number of algorithms that cooperate to provide effective resource management for cloud environments [6] [7]. With more effective storage and V/F scaling improvement, the EARU model significantly reduces LLC disappointments and thus more effectively utilizes asset. In terms of CPU utilization, preparation time, and energy output, it also achieves preferred execution to the board's current asset management plan [11]. Greedy method named MESA is recommended due to the high computing complexity of addressing the MINLP problem [18] in order to arrive at the best solution [13]. Also, advanced metaheuristic models are in need to proceed with optimal scheduling process.

This paper introduces a new optimization assisted task scheduling, and the main contributions are as follows:

- Initially, modified AHP process is introduced for ranking the task.
- A hybrid optimization model, namely, CUCMCA method for optimal scheduling of task with appropriate allocation of resources to execute the task.
- The proposed method is implemented using CloudSim simulator.

The work progresses initiating with intense literature review in Section II paving path for improvisation issues addressed with modified AHP based task ranking and hybrid optimal resource allocation provided in Section III followed by proposed CUCMCA Algorithm in Section IV formulates results and discussions in Section V Section VI contributes to conclusion of work with a wide overview of work.

II. Related Work

Several Researchers have contributed innumerable solutions addressing issues of scheduling and resource allocation. Despite, leaving few coins unturned that are addressed in our work with enhancements improvising performance.

In 2020, Mahdi Abbasi et al. [1] presented two approaches, XCS and BCM-XCS, depending on XCS - LCS, to manage the network's edge power consumption and lessen workload delay. The outcomes of this tests show that BCM-XCS is superior to the standard XCS-based approach. The workloads were distributed using the suggested approaches in a way that both the communication and processing delay among cloud and fog nodes were kept to a minimum. Additionally, the suggested approaches can recharge the reusable batteries utilized at the network's edge 18% faster than the existing technique.

In 2020, Yunliang Chen et al. [2] suggested a detailed QoS model to evaluate the performance level of data center clouds. To improve the cumulative QoS and sojourn time of all activities, an enhanced CESS algorithm was created. According to experimental findings, this approach outperforms the baseline algorithm in terms of accumulative QoS as well as sojourn duration by up to 56.1% and 25.4%, correspondingly. The algorithm's duration only increases linearly as more Cloud data centres and workloads are added. This technique constantly develops scheduling solutions with acceptable QoS without compromising sojourn time when the arrival rate as well as service rate ratio are kept constant.

In 2022, M. Hasan Jamal et al. [6] suggested a HAWDA and HASRA depending on thermal profiling considering outlet temperature detection. In order to reduce the peak output temperatures, HAWDA distributed workload on servers in a thermally efficient manner, while HASRA optimized server positioning in thermal hotspot areas. To evaluate the effectiveness of HAWDA against the TASA and GRANITE methods, performance comparison is done. Results showed that HAWDA, which reduces peak outlet temperature, achieved average peak server utilization comparable to GRANITE as well as TASA without adding additional load to the cooling system, with or without server relocation.

In 2022, M. Hasan Jamal et al. [6] suggested a HAWDA and HASRA depending on thermal profiling considering outlet temperature detection. In order to reduce the peak output temperatures, HAWDA distributed workload on servers in a thermally efficient manner, while HASRA optimized server positioning in thermal hotspot areas. To evaluate the effectiveness of HAWDA against the TASA and GRANITE methods, performance comparison is done. Results showed that HAWDA, which reduces peak outlet temperature, achieved average peak server utilization comparable to GRANITE as well as TASA without adding additional load to the cooling system, with or without server relocation.

In 2020, Ali Asghari et al. [7] developed a new architecture made up of many cooperating agents that took into account all aspects of TS and resource provisioning and managed the QoS offered to users. The integrated model that was suggested included all processes for TS and resource provisioning, and its many components help with managing user applications and making better use of cloud resources. This framework performs effectively with concurrent dependent activities, which have a challenging scheduling procedure due to the dependency of their subtasks.

In 2021, A. M. Senthil Kumar et al. [8] suggested a new task allocation method employing BAT and CSO method. The BAT algorithm aids the CSO algorithm in overcoming a pre convergence problem. The suggested HGCSBAT algorithm's performance was assessed and contrasted with that of the well-known CSO & BAT methods. In regards to availability & throughput, HGCSBAT performs better than that of the BAT, Cat Swarm Optimization, & Genetic algorithms. Traditional work scheduling algorithms features and limitations are given in Table I.

TABLE I. FEATURES AND LIMITATIONS OF TRADITIONAL TASK SCHEDULING ALGORITHMS

Author [citation]	**Methodology**	**Features**	**Limitations**
Mahdi Abbasi	XCS and BCM-XCS	Processing delay gets minimized	Workload latency is increased
Yunliang Chen	CESS method	Minimize waiting time	QoS is need to be upgraded for responsibilityexecution
Deepika Saxena	OP-MLB Method	Less power Consumption	Need for cloud data centre's performanceimprovement
Uma Tadakamalla	FogQN-AC	optimized cost average response time	Unplanned resources demand
Lei Yu	Stochastic Load Balancing approach	Migration cost is minimized	It is crucial to evaluate how different workload distributions affect load balancing performance.
M. Hasan Jamal	HAWDA and HASRA	Low memory usage	Fairer resource allocation needs a moreaccurate approach.
Ali Asghari	RMFW method	Reduced resource utilization	Higher computation cost
A. M. Senthil Kumar	HGCSBAT	Increased throughputand availability	Needs to consider balancing problem

In 2023 K.Pradeep, Sharma, and Jishnu[29] proposed an intense review on Task scheduling parameters which shed light on various strategies that make way for efficient scheduling with fault tolerant approaches in Fog computing. Their work emphasizes on tuning QoS parameters for improved results.

III. MODIFIED AHP BASED TASK RANKING AND HYBRID OPTIMAL SCHEDULING WITH APPROPRIATE RESOURCE ALLOCATION

Fig. 1 depicts the scenarios of Task Scheduling where user requests assemble at various virtual machines pertaining to a physical machine. These VMs promise improved performance by scheduling the tasks using our proposed approach. Scheduling of Tasks is performed in two phases i.e., Task ranking using FCM clustering followed by CUCMCA optimization for enhanced results.

A. System Model

Considering a data center with M servers include servers $A \in \{A_1, A_2, ..., A_M\}$, here, various VM types are purchased by Q users for executing the applications on U VM's includes $vm \in \{vm_1, vm_2, ..., vm_U\}$. Assume application R_Q pertaining to Q^{th} number of users depicted as $\{Tsk_1, Tsk_2, ..., Tsk_Z\} \in R_Q$, here Tsk_z represents the application task. The tasks are scheduled according to their resource requirements, which chooses the best VM for i^{th} task (Tsk_i^{res}) execution, where res defines resources such as memory, CPU, etc. as well as $vm_A^{res}, vm_Q^{res}, vm_L^{res}, vm_{XL}^{res}$ were small, medium, large as well as extra-large VM sets. The number of VM types available at a specific data center can be expanded. If need for resources of i^{th} task (Tsk_i^{res}) is equal or lesser to capacity of resource of vm_A, then smaller VM types were given to it. The proposed workload scheduling and resource management is progressed with two different steps:

1) *Task ranking:* This phase handles the process for ranking the tasks as per their priorities by first identifying priorities and generating Task queues pertaining to priority groups.

2) *Optimal scheduling:* Optimal scheduling phase deals with the assignment of corresponding resources according to the constraints resulting in better performance.

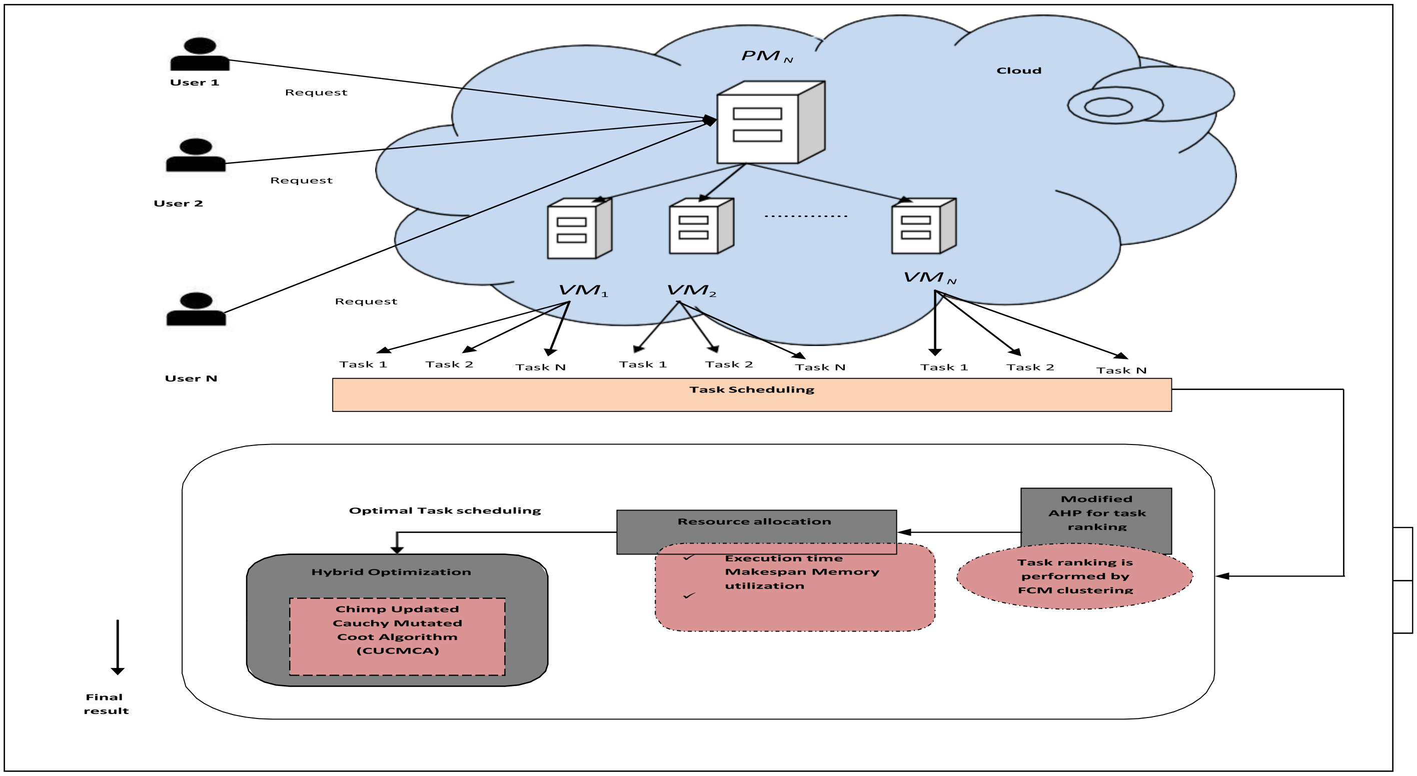


Fig. 1. Task scheduling in proposed approach.

B. *Task Ranking*

This is the first stage, where the ranking of task was done by using modified AHP (Analytical Hierarchy Process). Conventional AHP steps for ranking purpose are as follows:

- Implement the Saaty preference table [23]. Given below is Saaty preference table which offers point scale including descriptors (Table II).

TABLE II. SAATY TABLE

Points	Descriptor
1	Conditions were equally important
3	First condition is slightly more significant than second condition
5	First condition is rather more significant than second condition.
7	Obviously, the first condition is more significant than the second one.
9	The first condition is unquestionably more significant than the

- Afterwards each column summation and each column normalization and weighted sum is computed [24].
- Based on the Saaty preference table, AHP ranking is performed.

According to recommended concept, modified AHP process is followed for ranking purpose. Modified AHP steps are as follows:

- Instead of using the Saaty preference table, in modified AHP, FCM (Fuzzy c-Means Clustering) is followed on the basis of 1-9 clustering for implementing the table. A data collection is divided into N clusters using the FCM data clustering method, with each cluster having some of the data points in the dataset.
- Based on the FCM, AHP ranking is performed for task scheduling. Modified AHP based ranking is shown in Fig. 2.

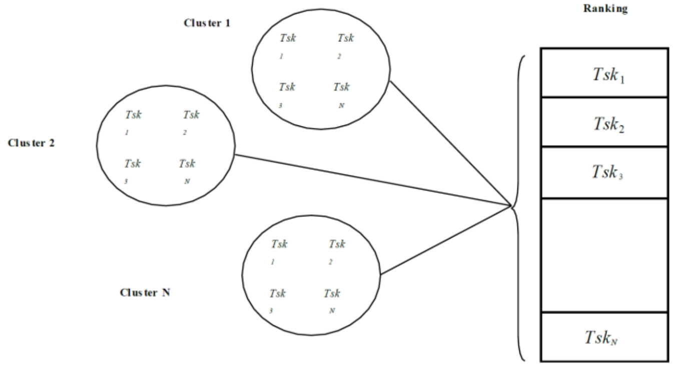


Fig. 2. Modified AHP based ranking paradigm.

C. *Optimal Task Scheduling and Resource Management*

In this step, the process of optimal scheduling is done by selecting the optimal PM and VM to execute the tasks. During this process, certain constraints are evaluated to ensure the scheduling in a precise manner. Also, the constrains involve in the assignment of corresponding resources ∇ to execute the tasks involved.

Constraints involved in this optimal scheduling process are as follows:

- Execution time
- Make span
- Memory utilization
- Communication cost

Optimal assignment of PM and VM for task scheduling: In this stage, optimal assignment of PM and VM for task scheduling is done by hybrid optimization combines Coot and Chimp optimization algorithms which is detailed in upcoming section. The factors used for optimal TS were:

- **Make span** ($F1$): Make span is described as the whole amount of time needed to complete the task. Make span is defined in eq. (1), where m denotes VM count, n denotes task count, Tsk denotes task, len denotes theTsk size in MI (Million Instructions), and $pesnum$ denotes PE (processing Element) in VM.

$$Makespan = Maxi_{1 \le i \le m}\{fTsk_i\} \quad (1)$$

- **Communication Cost** ($F2$): Here, communication cost among task i and task j are given to different VM's.

  Communication cost Com_Cost is calculated in eq. (2).

$$Com\_Cost = \sum_{Tsk_i=1}^{n} Com\_Cost(Tsk_i, i = 1,2, \dots n) \quad (2)$$

- **Execution time (** $F3$**):** The time duration needed by VM to finish each task is known as execution time.
- **Memory Utilization (** $F4$ **):** Memory Utilization is a measure of average memory usage that is calculated by averaging the percentage of memory space that is being used at any given time across the reporting interval. Memory Utilization is defined in eq. (3).

$$Mem - Util = \left(100^*(TotalMBytes - AvailMBytes/TotalMBytes)\right) \quad (3)$$

$$TotalMBytes \rightarrow vm(Mem)$$

Here,

$$AvailMem \rightarrow (VirtualMem - TaskMem)$$

Solution encoding: The selection process will be decided by the proposed optimization algorithm where the solution including both PM and VM set from which the model selected corresponding PM and VM to execute the respective task. Fig. 3 gives solution encoding ofproposed CUCMCA method.

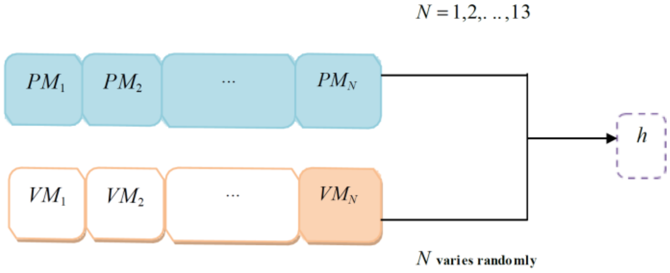


Fig. 3. Solution encoding of proposed CUCMCA method.

Weighted Objective Function: Minimization: The objective function obj defined in the model is given in eq. (4), where $w1$, $w2$, $w3$, $w4$ are the weights assigned to each parameter.

These weights are calculated using chaotic cubic map function.

$$obj = w1^*F1 + w2^*F2 + w3^*F3 + w4^*F4 \quad (4)$$

Cubic map: The cubic map is one of the maps that are most frequently used to generate chaotic sequences in many applications. In eq. (5), cubic map is defined.

$$L_{y+1} = 259 \times L_y\left(1 - L_y^2\right) \quad (5)$$

IV. Chimp Updated Cauchy Mutated Coot Algorithm (CUCMCA) Optimization

Hybrid optimizations are a new class of optimization methods that we develop to solve the optimization issue with more convergence efficiency. For hybrid optimizations, two or more algorithms must have been used for the same optimization. In this paper, we hybridized two algorithms named coot and chimp optimization. The solution update is done by this hybridized algorithm. Here the random number K^2 is estimated by using Tent map function. Also, Cauchy mutation is introduced in our proposed concept. Our proposed hybrid algorithm concept is given below:

Small water birds called Coots [21] were the rail family members. In coot optimization, with the formula (6), a small area is used to produce the population at random, where $Coot^{pos(i)}$ represents the coot position, g represents the variables count, and ub,lb represents the upper as well as lower bound of search space.

$$C^{pos(i)} = rnd(1, O)^*(ub - lb) + lb \quad (6)$$

Random motion to this direction and that direction: Different areas of the search space are explored by coot migration. This movement will let the algorithm escape the local optimal if it becomes stuck in the local optimal. Coot's new position is determined in eq. (7).

$$C^{pos(i)} = C^{pos(i)} + G \times K^2 \times \left(Z - C^{pos(i)}\right)$$

$$Z = rnd(1, O)^*(ub - lb) + lb \quad (7)$$

Where, K^2 represents the random number which is calculated using tent map function according to proposed model, $G = 1 - Y \times \left(\frac{I}{iter}\right)$ where Y represents the current iteration, $iter$ represents the maximum iteration.

Tent map: Tent chaotic map is also refereed as the logistic map represents particular chaotic effects. The following equation (8) gives the definition of this map:

$$q_{r+1} = \begin{cases} 2qr, qr < 0.5 \\ 2(1 - q_r), q_r \geq 0.5 \end{cases} \tag{8}$$

Chain movement: It is possible to construct chain movement by using the average position of 2 coots. The formula (9) is used to calculate the coot's new position, where $C^{Pos}(i-1)$ represents the second coot.

$$C^{Pos(i)} = 0.5 \times \left(C^{Pos}(i-1) + C^{Pos}(i)\right) \tag{9}$$

Adjusting position in accordance with the group leaders: To carry out this movement, a system is deployed based on the formula (10) to choose the leader, where i represents current coot index, D represents leader index, and lc represents the leader count.

$$D = 1 + (iMODlc) \tag{10}$$

Depending on leader O , $C^{Pos}(i)$ update its position. Formula (11) uses the chosen leader to determine the coot's subsequent position, where $C^{Pos}(i)$ represents new coot position, $L^{pos}(i)$ represents selected leader position, and *I1*, *I* represents the random number.

$$C^{Pos}(i) = L^{Pos}(O) + 2 \times I1 \times cos(2I\pi) \times \left(L^{Pos}(O) - C^{Pos}(i)\right) \tag{11}$$

According to proposed model, position update is done by hybridizing Coot and Chimp position [22] which is specified in eq. (12) to eq. (16).

Proposed update equation: $h_{chimp} = h_{prey}(t) - u.v$

Substitute $L^{Pos} = LP, h_{prey} = h, C^{Pos} = h$

$$h - u.v = LP(O) + 2\times I1\times\cos(2I\pi)\times LP(O) - C^{Pos}(i) \tag{12}$$

$$h - u.v = LP + 2I1\times\cos(2I\pi)\times(LP - h) \tag{13}$$

$$h - u.v = LP + 2I1\times\cos(2I\pi)\times LP - 2I1\times\cos(2I\pi)\times h \tag{14}$$

$$h - u.v = LP(1 + 2I1\times\cos(2I\pi) - 2I1\times\cos(2I\pi)\times h \tag{15}$$

$$h = LP(1 + 2I1\times\cos(2I\pi) + u.v - 2I1\times\cos(2I\pi)\times h \tag{16}$$

Leader movement: Leaders need to change their location with respect to the objective in order to move the group toward an objective (the ideal region). It is advised to update leader position using formula (17), where $OBest$ represents the best position, $J3$, $J4$ represents the random number.

$$L^{Pos(i)} = \begin{cases} S \times I3 \times cos(2I\pi) \times \begin{pmatrix} FBest - L^{Pos(i)} \\ +FBest,\ I4 < 0.5 \end{pmatrix} \\ S \times I3 \times cos(2I\pi) \times \begin{pmatrix} FBest - L^{Pos(i)} \\ -FBest,\ I4 \geq 0.5 \end{pmatrix} \end{cases} \tag{17}$$

Here, $S = 2 - R\left(\frac{1}{S_t}\right), S^t$ represents maximum iteration, and R represents the current iteration.

Cauchy mutation: Cauchy mutation is also employed in this algorithm to produce the solution. Due to its wider search range, Cauchy mutation has a significant ability to seek globally. This ensures the high convergence rate. Below Algorithm 1 gives the pseudocode of suggested CUCMCA model:

Algorithm 1: Pseudo code of Chimp Updated and Cauchy Mutated Coot (CUCMCA)

```
Input: Randomly initialize coot population (Tasks and VMs)
Output: Optimally mapped VMs and Tasks
Parameter initialization $\Pr = 0.5$, $Lc$, $Cnt_{coot}$ ( coot count)
$Cnt_{coot} = Cnt_{pop} - Cnt1$
Randomly select the leader of coot
Coot as well as leader fitness calculation
Identify best leader or coot as $FBest$
While end condition is not met
    Calculate $t, S$ parameter
    if $rnd < \Pr$
        $I, I1, I3$ were random numbers along the problem dimension
    Else
        $I, I1, I3$ were random number
    end if
    $I, I1, I3$ were random number
    for $i = 1$ to $Cnt_{coot}$
        Evaluate parameter of $D$
        if $rnd > 0.5$
            Position update using new evaluation given in eq. (16)
        Else
            if $rnd < 0.5i$ ~=1
                update coot position by eq. (7)
                In eq. (7), random number $K^2$ is calculated using the Tent map as per proposed model
            end if
        end if
        Evaluate fitness using eq. (4)
        if $C^{fitness} < L^{fitness}(O)$
            $temp = L(O)$
            $L(O) = coot$
```

```
$coot = temp$
end if
for leader count
Leader position update by eq. (17.1)
Else
Leader position update by eq. (17.2)
end for
if $L^{fitness} < FBest$
$temp = FBest$
$FBest = L$
$L = temp$
end if
end for
$S^t = S^t + 1$
Cauchy mutation is performed for global search
end while
```

V. Results and Discussion

A. Simulation Procedure

The proposed Chimp Updated and Cauchy Mutated Coot Algorithm (CUCMCA) method for task Scheduling and Resource Management was done in Cloudsim. The dataset considered for our work are extracted from internet sources i.e. google cluster traces 2019. The assessment was done on Bald Eagle Search (BES), Arithmetic Optimization Algorithm Method (AOAM) [26], Moth Flame Optimization (MFO), Hybrid Swarm Optimization (HSO) [27], Elephant Herding Optimization (EHO), Chimp and COOT, regarding Communication Cost, Execution Time, Fitness, Makespan and Memory Utilization. Also, it was examined by altering the number of virtual machines to 10, 20, 30, 40 and 50.

B. Dataset Description

This is a trace of the workloads running on eight Google Borg compute clusters for the month of May 2019 [25]. The trace describes every job submission, scheduling decision, and resource usage data for the jobs that ran in those clusters.

It builds on the May 2011 trace of one cluster, which has enabled a wide range of research on advancing the state-of-the-art for cluster schedulers and cloud computing, and has been used to generate hundreds of analyses and studies.

Since 2011, machines and software have evolved, workloads have changed, and the importance of workload variance has become even clearer. The new trace allows researchers to explore these changes.

The new dataset includes additional data, including:

- CPU usage information histograms for each 5 minute period, not just a point sample;
- Information about allow sets (shared resource reservations used by jobs); and
- Job-parent information for master/worker relationships such as MapReduce jobs.

Just like the last trace, these new ones focus on resource requests and usage, and contain no information about end users, their data, or access patterns to storage systems and other services."

C. Evaluation of Communication Cost

The proposed CUCMCA method is compared to extant systems in terms of communication cost. Fig. 4 depicts the study of communication cost. A variety of Virtual Machines, including 10, 20, 30, 40 and 50 are evaluated. A successful system should have minimal communication costs. On examining the communication cost, the values gained by recommended method are considerably lower than other models. That is, the communication cost of (~) 125 is obtained by adopted method at the 40th VM. In contrast, the compared models like BES, AOAM, MFO, HSO, EHO, Chimp and COOT has obtained relatively higher communication cost of 174, 176, 132, 177, 175, 163 and 164, respectively. While compared to other schemes such as BES, AOAM, MFO, HSO, EHO, Chimp and COOT, the communication cost achieved using suggested approach is the smallest in the 50th VM. Thus, the outcomes of the experiment reveal that the proposed CUCMCA method's communication cost score is preferable to the established approaches.

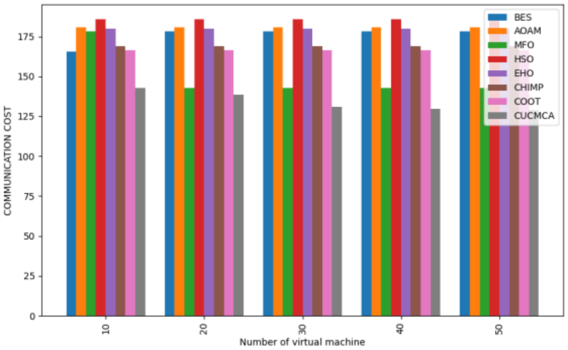


Fig. 4. Communication cost analysis: proposed CUCMCA method vs conventional models.

D. Evaluation on Execution Time

In this section, the analysis on Execution Time is examined for varied VMs. The analysis on developed CUCMCA method over BES, AOAM, MFO, HSO, EHO, Chimp and COOT for varied VMs is exposed in Fig. 5. In order to improve the system's performance, the execution time should be reduced. The developed approach holds minimal execution time of 8.24 at the VM 50; whereas, the traditional models holds the highest execution time for BES (39.53), AOAM (47.92), MFO (49.67), HSO (62.18), EHO (18.65), Chimp (48.85) and COOT (53.67), respectively. Moreover, it is observed that, the suggested method has accomplished better outcomes at 30th VM than at 10th and 20th VMs. As a result, the superiority of the suggested CUCMCA work is proved.

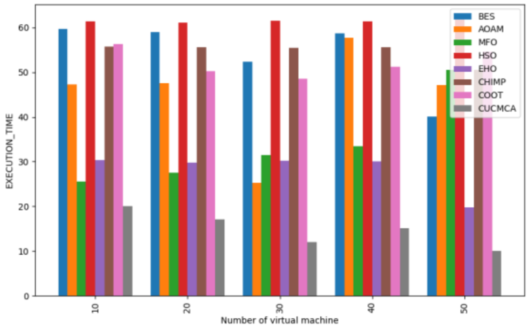


Fig. 5. Execution time analysis: proposed CUCMCA method vs conventional models.

E. Evaluation of Fitness

The Fitness analysis of the proposed CUCMCA classifier is computed over the existing classifiers and the graphical illustration is represented in Fig. 6. Further, the Fitness of the suggested model attains better outcomes than other conventional approaches. The models like BES, AOAM, MFO, HSO, EHO, Chimp and COOT acquired the highest fitness of (~) 28.69, 23.54, 20.71, 28.67, 17.46, 26.82 and 25.18, whilst the proposed strategy yielded the lowest fitness of 17.82, at the 30th VM. Likewise, the fitness of the adopted approach obtained the better value of (~) 14.28; however, the existing schemes like BES, AOAM, MFO, HSO, EHO, Chimp and COOT holds the lowest values for the VM 40. Hence, the improvement of the suggested CUCMCA model is established over others in terms of fitness.

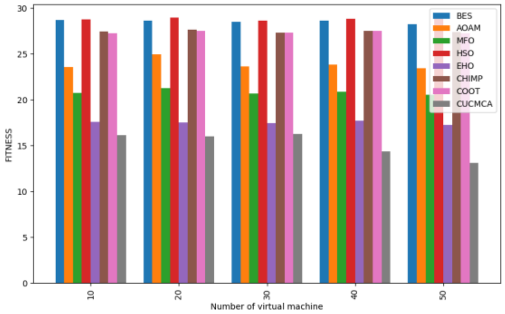


Fig. 6. Fitness analysis: proposed CUCMCA method vs conventional models.

F. Evaluation of Makespan Time

The Makespan using suggested CUCMCA method is analyzed over BES, AOAM, MFO, HSO, EHO, Chimp and COOT for varied VMs as shown in Fig. 7. Low makespan is required for enhanced system performance. The suggested method successfully achieves our goal, whose Makespan values exceed the traditional approaches. Moreover, the Makespan of the developed model attains lower value of 43.56, in the 20th VM than other existing classifiers like BES (82.67), AOAM (68.42), MFO (76.84), HSO (63.45), EHO (81.96), Chimp (80.87) and COOT (78.12), respectively. This analysis shows that the developed CUCMCA approach makes the system more robust at scheduling workloads and managing resource than the conventional approaches.

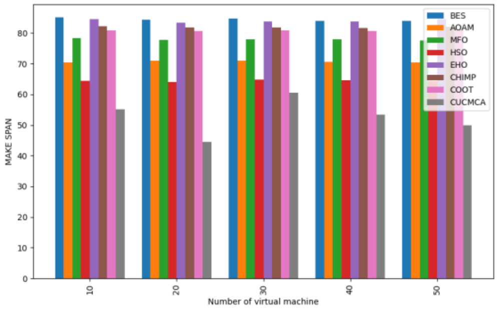


Fig. 7. Makespan analysis: proposed CUCMCA method vs conventional models.

G. Analysis on Memory Utilization

The performance of adopted CUCMCA model regarding Memory Utilization is displayed in the Fig. 8. The memory utilization should be minimal for better system performance. In this manner, it is observed that the adopted model achieves least memory utilization when compared to models like BES, AOAM, MFO, HSO, EHO, Chimp, and COOT, respectively. Particularly, incredible outcomes for both proposed and current approaches have been obtained for all measures at the 10th, 30th, and 50th VM. Nevertheless, the developed approach has delivered more determinative outcomes than distinct strategies for every VM. For instance, at 50th VM, the memory utilization of suggested approach is 0.04, which is better than the values obtained for existing schemes like BES is 0.05, AOAM is 0.06, EHO is 1.2, COOT is 1.1 and HSO is 0.07, respectively. As a consequence, this assessment proves that the proposed CUCMCA model is better to make an efficient workload scheduling when compared to other conventional models.

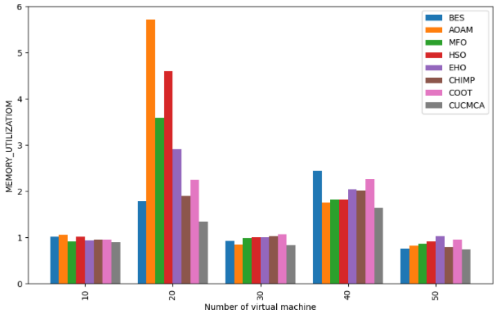


Fig. 8. Memory utilization analysis: proposed CUCMCA method vs conventional models.

H. Convergence Analysis

The convergence study of the proposed CUCMCA work is contrasted to the traditional methods like BES, AOAM, MFO, HSO, EHO, Chimp and COOT are shown in Fig. 9. In addition, it was examined by altering the iterations to 5, 10, 15, 20 and 25, respectively. When comparing the suggested method to other extant schemes, the findings show that the developed model has the lowest error rate. Here, the COOT algorithm has exhibited the worst performance in the initial (0th) iteration. From iteration 5 through iteration 25, the proposed approach and other current classifiers have lower error rates. Nevertheless, at the last 25th iteration, the adopted approach recorded the lowest error rate of (~) 1.0. Thus, recommended strategy resulted in a slightly lower error rate than BES, AOAM, MFO, HSO, EHO, Chimp and COOT. Therefore, the proposed CUCMCA strategy is appropriate for the workload scheduling and resource management. The improvisation in the performance of proposed logic shows the impact of proposed hybrid algorithm in enhancing the convergence rate and speed.

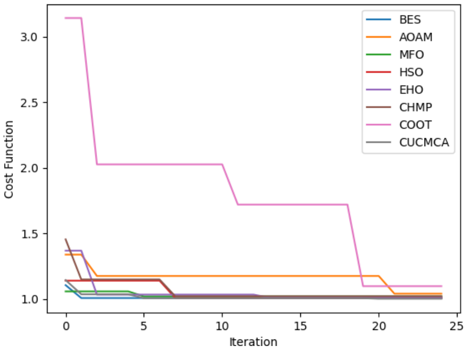


Fig. 9. Convergence analysis: proposed CUCMCA method vs conventional models.

I. Statistical Analysis

Table III represents the statistical analysis with respect to Fitness, Makespan, Communication Cost, Memory Utilization and Execution Time for the proposed CUCMCA method over the established models. Also, the analysis was carried out with five different case scenarios including Mean, Maximum, Standard Deviation, Minimum and Median. The optimization schemes are stochastic, and to substantiate the fair assessment, every model is examined several times. On examining the resultants, the suggested scheme has achieved minimal values for the majority of the scenarios. Based on the mean case scenario analysis, the proposed model obtains lower execution time of 12.06345 than other traditional models like AOAM is 44.99084, MFO is 33.69218, HSO is 61.50506, EHO is 28.04894, Chimp is 54.63884, COOT is 52.19459 respectively. According to the Median analysis, the Memory utilization of the suggested work is 0.892668, which is superior to the existing models like BES, AOAM, MFO, HSO, EHO, Chimp and COOT.

TABLE III. STATISTICAL ANALYSIS WITH RESPECT TO OBJECTIVE FUNCTION

Fitness								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	28.54696	23.87702	20.79797	28.80408	17.49941	27.45825	27.3536	15.17077
Maximum	28.70305	24.95545	21.27113	28.96292	17.69324	27.62056	27.50861	16.2401
Standard Deviation	0.153195	0.554059	0.260911	0.119192	0.14309	0.102286	0.125922	1.229137
Minimum	28.27161	23.43405	20.51185	28.60521	17.2719	27.34739	27.22187	13.12047
Median	28.60423	23.61853	20.71881	28.84593	17.47804	27.46037	27.28607	15.99167
Makespan								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	84.38679	70.68699	77.88326	64.44666	83.98068	81.87688	80.84271	54.48431
Maximum	85.10643	71.04397	78.3	64.82723	84.51164	82.1506	81.07052	55.07267
Standard Deviation	0.43992	0.27665	0.252149	0.259594	0.427563	0.176513	0.151499	0.35984
Minimum	83.9178	70.32722	77.54787	64.01315	83.39151	81.65603	80.657	53.93727
Median	84.28038	70.64215	77.90364	64.45928	83.84175	81.84449	80.85056	54.4618
Communication Cost								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	175.4903	180.7314	149.715	185.774	179.7471	168.8011	166.3097	141.8741
Maximum	178.0185	180.7314	178.0185	185.774	179.7471	168.8011	166.3097	142.6391
Standard Deviation	5.056402	0	14.15177	0	0	0	0	0.38248
Minimum	165.3775	180.7314	142.6391	185.774	179.7471	168.8011	166.3097	141.6829
Median	178.0185	180.7314	142.6391	185.774	179.7471	168.8011	166.3097	141.6829
Memory Utilization								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	1.380963	2.036544	1.634815	1.869821	1.586945	1.334175	1.496101	1.08832
Maximum	2.442056	5.718419	3.585442	4.599554	2.915935	2.009994	2.267544	1.637185

Standard Deviation	0.635769	1.871789	1.03697	1.403908	0.780297	0.513353	0.621799	0.342108
Minimum	0.753165	0.812619	0.859829	0.914704	0.940985	0.793372	0.95027	0.74304
Median	1.015725	1.058777	0.993748	1.016745	1.026041	1.025408	1.063376	0.892668
Execution Time								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	53.95939	44.99084	33.69218	61.50506	28.04894	54.63884	52.19459	12.06345
Maximum	59.70747	57.66733	50.49382	62.11024	30.40323	55.78705	56.38825	20.06346
Standard Deviation	7.443277	10.623	8.849737	0.337976	4.119379	2.040145	2.887903	6.542089
Minimum	40.02318	25.31515	25.57135	61.08178	19.81941	50.56235	48.49736	1.063507
Median	58.68128	47.28658	31.5239	61.39719	30.01077	55.6951	51.19773	12.06383

VI. Conclusion and Future Scope

Cloud system must be able to manage its resources autonomously and adaptively in response to the changes in workload needs with its vast computing power and flexibility for ever evolving challenges. Despite credibility features, it faces many challenges such as Scheduling, Security, and Energy Management etc. Among aforementioned issues the one of concern that owes to be improved is Task scheduling, intending the maximization of user-favoured application QoS parameters our proposed hybrid algorithm performs task scheduling and allocates resources efficiently in cloud computing environments. Our work considers Google cloud (May 2019) workload traces as input using modified AHP to rank the task via FCM. Furthermore, the optimal task scheduling and resource allocation are done by the developed Chimp Updated and Cauchy Mutated Coot Algorithm (CUCMCA). Deliberately the outcome promises improved results in comparison to the existing conventional BES, AOAM, MFO, HSO, EHO, Chimp, COOT models with respect to makespan time, execution time, communication cost and memory utilization gives improved results. As a scope for further enhancement classifying user tasks prior to scheduling promises improved results in terms of QoS metrics and extends scope for better resource allocation.

References

[1] Mahdi Abbasi, Mina Yaghoobikia, Milad Rafiee, Alireza Jolfaei, Mohammad R. Khosravi,, “Efficient resource management and workload allocation in fog-cloudcomputing paradigm in IoT using learning classifier systems”, Computer Communications, vol.153, 2020

[2] Y. Chen et al., "Stochastic Workload Scheduling for Uncoordinated Datacenter Clouds with Multiple QoS Constraints," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1284-1295, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2016.2586048.

[3] D. Saxena, A. K. Singh and R. Buyya, "OP-MLB: An Online VM Prediction based Multi-objective Load Balancing Framework for Resource Management at Cloud Datacenter," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2021.3059096.

[4] U. Tadakamalla and D. A. Menasce, "Autonomic Resource Management for Fog Computing," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2021.3064629.

[5] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang and Y. Pan, "Stochastic Load Balancing for Virtual Resource Management in Datacenters," in IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 459-472, 1 April-June 2020, doi: 10.1109/TCC.2016.2525984.

[6] M. Hasan Jamal, M. Tayyab Chaudhry, Usama Tahir, Furqan Rustam, SoojungHur and Imran Ashraf,”Hotspot-Aware Workload Scheduling and Server Placement forHeterogeneous Cloud Data Centers”, 2022

[7] Ali Asghari, Mohammad Karim Sohrabi, FarzinYaghmaee,”A cloud resource management framework for multiple online scientific workflows using cooperative reinforcement learning agents”, Computer Networks, vol. 179, 2020

[8] Senthil Kumar, A.M., Padmanaban, K., Velmurugan, A.K. et al. A novel resource management framework in a cloud computing environment using hybrid cat swarm BAT (HCSBAT) algorithm. Distrib Parallel Databases (2021) https://doi.org/10.1007/s10619-021-07339-w

[9] Sukhpal Singh Gill, et al. “ThermoSim: Deep learning-based framework for modeling and simulation of thermal-aware resource management for cloud computing environments”, Journal of Systems and Software, vol.166, 2020

[10] Neda Khorasani, SaeidAbrishami, Mehdi Feizi, Mahdi AbolfazliEsfahani, FaezeRamezani,"Resource management in the federated cloud environment using Cournotand Bertrand competitions”, Future Generation Computer Systems, vol.113, 2020

[11] Murgesh V. Jambigi, Dr. M.V. Vijay Kumar, Dr. D.V. Ashoka, Dr. Prabha R,”Energy Aware Resource Utilization Technique for Workflow Scheduling in Cloud Computing Environment”, Turkish Journal of Computer and Mathematics Education, vol.12, 2021

[12] Yanqi Zhang, Weizhe Hua, Zhuangzhuang Zhou, G. Edward Suh, Christina Delimitrou,"Sinan: ML-Based and QoS-Aware Resource Management for Cloud Microservices"2021

[13] Mostafa HadadianNejadYousefi, AmirmasoudGhiassi, Boshra Sadat Hashemi, MaziarGoudarzi, “Workload Scheduling on heterogeneous Mobile Edge Cloudin 5G networks to Minimize SLA Violation”, 2020

[14] Shreshth Tuli, Sukhpal Singh Gill, Minxian Xu, Peter Garraghan, Rami Bahsoon, SchahramDustdar, RizosSakellariou,Omer Rana, Rajkumar Buyya, Giuliano Casale, Nicholas R. Jennings,”HUNTER: AI based Holistic Resource Management for Sustainable Cloud Computing’, Distributed, Parallel, and Cluster Computing, 2021

[15] R. Pinciroli, A. Ali, F. Yan and E. Smirni, "CEDULE+: Resource Management for Burstable Cloud Instances Using Predictive Analytics," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 945-957, March 2021, doi: 10.1109/TNSM.2020.3039942.

[16] Sohan Kumar Pande, Sanjaya Kumar Panda, Satyabrata Das, Kshira Sagar Sahoo, Ashish Kr. Luhach, N. Z. Jhanjhi, RoobaeaAlroobaea and Sivakumar Sivanesan,” A Resource Management Algorithm for Virtual MachineMigration in Vehicular Cloud Computing”, Tech Science Press, vol.67, 2021

[17] M.Buvana, Dr.K.Loheswaran, et al., ”Improved Resource Management and Utilization Based on a Fog-Cloud Computing System with IoT Incorporated with Classifier Systems”,Microprocessors and micro systems,2021.

[18] FarzinZaker, Marin Litoiu, and Mark Shtern,, “Formally Verified Scalable Look Ahead Planning For Cloud ResourceManagement”, 2022

[19] Wang, B., Wang, C., Song, Y. et al. A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds. Cluster Comput 23, 2809–2834 (2020).https://doi.org/10.1007/s10586-020-03048-8

[20] Saif, M.A.N., Niranjan, S.K. & Al-ariki, H.D.E. Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis. Wireless Netw 27, 2829–2866 (2021). https://doi.org/10.1007/s11276-021-02614-1.

[21] IrajNaruei and FarshidKeynia, "A new optimization method based on COOT bird natural life model ", Expert Systems With Applications, vol. 183, 2021

[22] M.Khishe and M.R.Mosavi, "Chimp optimization algorithm", Expert Systems with Applications, vol. 149, 2020.

[23] Alena Paulikova, Zdenka Gyurák Babel'ová and Monika Ubárová, "Analysis of the Impact of Human–Cobot Collaborative Manufacturing Implementation on the Occupational Health and Safety and the Quality Requirements", International Journal of Environmental Research and Public Health, 2021.

[24] Mahendra Bhatu Gawali and Subhash K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach", Journal of Cloud Computing: Advances, Systems and Applications, 2018.

[25] Datasets https://research.google/tools/datasets/google-cluster-workload-traces-2019/

[26] Mohamed Abd Elaziz , Laith Abualigah , Rehab Ali Ibrahim and Ibrahim Attiya,"IoT Workflow Scheduling Using Intelligent Arithmetic Optimization Algorithm in Fog Computing", Computational Intelligence and Neuroscience, 2021.

[27] Heba M. Eldesokey, Saied M. Abd El-atty, Walid El-Shafai, Mohammed Amoon, Fathi E. Abd El-Samie, "Hybrid swarm optimization algorithm based on task scheduling in a cloud environment", communication systems, vol.34, 2021.

[28] Syed Karimunnisa,Vijaya Sri ,"Cloud Computing: Review on recent research progress and issues", International Journal of Advanced Trends in Computer Science and Engineering, 2019, 8(2), pp. 216–223.

[29] Pradeep K, Sharma, and Jishnu , "Review of Task Scheduling based on different parameters in Cloud Environment", Micro Processors and Microsystems,2023.

DMobile-ELA: Digital Image Forgery Detection via Cascaded Atrous MobileNet and Error Level Analysis

Karma M.Fathalla[1], Malak Sowelem[2], Radwa Fathalla[3]
Computer Engineering Dept., College of Engineering. AAST. Alexandria, Egypt [1, 2]
Computer Science Dept., College of Computing and Information Technology, AAST.Alexandria, Egypt[3]

Abstract—**With the current developments in technology, not only has digital media become widely available, the editing and manipulation of digital media has become equally available to everyone without any prior experience. The need for detecting manipulated images has grown immensely as it can now cause false information in news media, forensics, and daily life of common users. In this work, a cascaded approach DMobile-ELA is presented to ensure an image's credibility and that the data it contains has not been compromised. DMobile-ELA integrates Error Level Analysis and MobileNet-based classification for tampering detection. It was able to achieve promising results compared to the state of the art on CASIAv2.0 dataset. DMobile-ELA has successfully reached a training accuracy of 99.79% and a validation accuracy of 98.48% in detecting image manipulation.**

Keywords—Tampering detection; MobileNet; error level analysis; CASIAv2.0

I. Introduction

Digital imaging use has recently prevailed in various domains, starting from social networks [1], through medical diagnostics [2] till reaching its use as digital forensics court evidence [3]. Coupled with its use in different critical fields, technology advancements has led to the ease of digital image manipulation and forgery[4].

Image tampering and forgery include a wide range of types such as copy-move, splicing, retouching and image morphing [5]. Copy-move forgery includes copying a piece of the same picture and moving it to cover another part of the image, while splicing involves copying a part of an image to place it in another image. Retouching often involves changes in shape, color and texture of image parts to improve its visual and technical quality, whereas image morphing perform images interpolation to create an image blend. Recently, Generative Adversial Networks (GANs) made it possible to create full face fake images and media using DeepFake technology[6].

The wide availability of editing and enhancement tools may encourage the malicious use of such tools in criminal acts. Such possibility raises public concern and demand for verifying the originality of the images. Hence, effective approaches are required for detecting image forgery [7]. Forgery detection revolves around the recognition of image manipulation and authenticity validation. Active techniques such as digital signatures and watermarking can be used, in addition to passive detection techniques [7].

In this study, a passive forgery detection approach DMobile-ELA is proposed to automatically detect copy-move and splicing image edits with high accuracy. A dilated modified MobileNet architecture is presented to determine whether an image is authentic or tampered. Error Level Analysis (ELA) is used to preprocess the investigated image at different compression levels before being input to the Dilated-MobileNet. The proposed approach has the advantages of encompassing a light weight architecture suitable for mobile device use. Also, the atrous modification allows the network to capture larger spatial context; which increases its ability to reconstruct more complex edge structures. In addition, the adopted ELA preprocessing enables the detection of the tampered areas easily, due their characteristic aspects in the ELA representation.

This paper is organized as follows: a background on deep learning is given in Section II covering two of the most popular architectures VGG16 and ResNet-50 to allow further comparison. In Section III , a briefing on the related studies will be provided before presenting DMobile-ELA forgery detection system in Section IV. The experimental setup and results will be discussed in Section V. Finally, the conclusions will be drawn in Section VI.

II. Background

A. Deep Neural Network

Deep learning or Deep neural network (DNN) belongs to the class of machine learning, which models high level abstractions in the data with multiple nonlinear transformations [8]. DNN is a subclass of neural networks requiring large volumes of data to increase the efficiency of the training processes. The term "deep" also known as hierarchical learning represents the large number of multiple hidden layers, which includes nonlinear processing units for the purpose of conversion and automatic feature extraction [8].

1) Convolution neural networks: Convolution Neural Networks (CNNs) can extract automatic discriminative features which have some invariance properties (e.g. translation invariance) [9]. It consists of three main layers which are convolution layers, pooling layers and fully connected layers [8].

The early convolution layers of the architecture are used for extracting local low-level features from the raw input while the deeper convolution layers of CNN are used for combining features together to generate global high-level features. The pooling layers are used to down sample the dimensionality of the extracted feature. The fully connected layers form an ANN network where each neuron in the previous layer is connected to all the neurons in the current layer. The total number of fully connected neurons in the final layer determines the number of classes [8].

The advantages of CNNs include that they are well suited for end-to-end learning that generates automatic features from the raw data without any a priori feature selection. Moreover, CNNs scale well to large datasets. The disadvantages of CNNs include the large amount of training data, the long training time compared to simpler models, and the large number of hyper parameters to be learned. Two of the most famous CNN are VGG-16 [10] and ResNet-50 [11], which will be described briefly below.

a) Visual graphic group net (VGG Net) model: This net was developed by the technicians at the Visual Graphics Group from the Oxford and is in pyramid shape. The model consists of the bottom layers which are wide and the top layers are deep. There are two versions of VGG which are VGG-16 and VGG-19. VGG-16 is a combination 13 convolutional layers and Three fully connected layers as shown in . The VGG-19 is a much deeper network with 16 convolutional layers and three fully connected layers.

b) depicts VGG16 layered architecture.

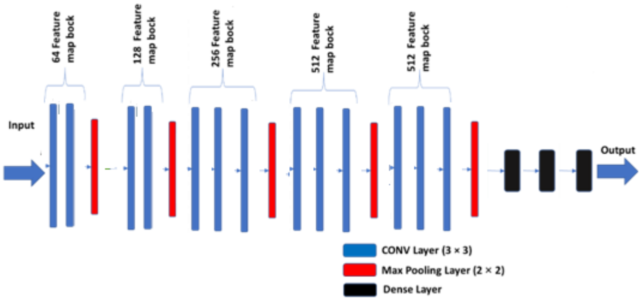


Fig. 1. VGG16 layered architecture.

c) ResNet model: It is a type of deep network based on residual learning. There are different versions of ResNet which are ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152. All of them have the same building units or residual blocks and formed by stacking the building residual blocks over each other. Any ResNet starts with block has a structure as shown in The residual blocks; are divided into two types which are Identity shortcut and Projection shortcut. The first block shown in Fig. 2 is the identity shortcut bottleneck block which is composed of a sequence of convolution layers of kernel size (1 × 1) and stride = 1 connected to a convolution layer with kernel (3 × 3) and stride = 1 followed by a convolution layer followed by kernel (1 × 1) and stride = 2. This block is used when the input and output of feature map are the same. The other block shown in Fig. 2 is the projection shortcut bottleneck block which has the same sequence of layers with a newly added convolution layer in the projection shortcut which has a kernel size of (1 × 1) with stride = 2. It is applied when shortcuts go across the feature map of two sizes. In the two blocks all the convolution layers are followed by batch normalization and RELU activation function. The difference between different ResNet versions are the number of stacked residual blocks. For example, ResNet-50 which has 16 residual blocks and ends with fully connected layer as shown in Fig. 2.

III. Related Work

Due to the recent advancements in computer vision and the growing need for forgery detection, resources for targeted algorithms are vastly diverse in their approaches and practices.

Originally, the leading method in identifying tampered and non-tampered images was Support Vector Machine (SVM), as seen in [12], [13] and [14]. Shen et al. [12] were able to achieve quite high accuracies using the datasets CASIAv1.0 and CASIAv2.0 reaching 98% and 97% respectively. TF-GLCM method was proposed, which combines textural features extraction with grey level co-occurrence matrices. This method was directed at spliced images in particular. They used calculated textural features as components in feature vectors in order to recognize genuine and spliced images employing SVM as the classifier.

Similarly, Han et al. [13] used SVM to classify spliced images but after extracting features using the Markov method. They presented three types of Markov feature vectors and achieved accuracies up to 97.86% for CASIA v1 and 97.33% for CASIA v2 even with a small range of features.

Recent approaches are now leaning towards more complex neural network architectures, especially the ones to be able to detect more than one type of tampering rather than only splicing which was previously the case. One prominent study by Rao et al. [17] introduces a new CNN designated for the detection of copy-move and splicing forgeries. It utilizes high-pass filters to calculate residual maps in a special rich model (SRM) to capture any subtle pattern that is produced when image manipulation happens. The used CNN extract features from the test images, and a feature fusion method is then applied to acquire the final key features that are fed to SVM for classification. This method was able to achieve 98.04% for CASIA v1 and 97.83% for CASIA v2.

An interesting approach was presented by Sudiatmika et al. [5] , who utilized the idea of error-level analysis (ELA) in conjunction with CNNs to create a more universal tool for detecting various types of forgery. Sudiatmika proposed normalizing the images before pursuing ELA calculation and feeding the resulting images to a VGG16 network. Sudiatmika et al. reached 92.2% accuracy on CASIA v2.0.

Kuznetsov [18] took a slightly different tactic in detecting forgeries using VGG network. The adopted method did not use the entire image for classification but rather small patches that are identified by either being forged on original expanding the training pool. He used a sliding window method to analyse each fragment of the image regarding its authenticity. This approach achieved very good results reaching 97.8% accuracy, 97.1% precision and 96.8% recall.

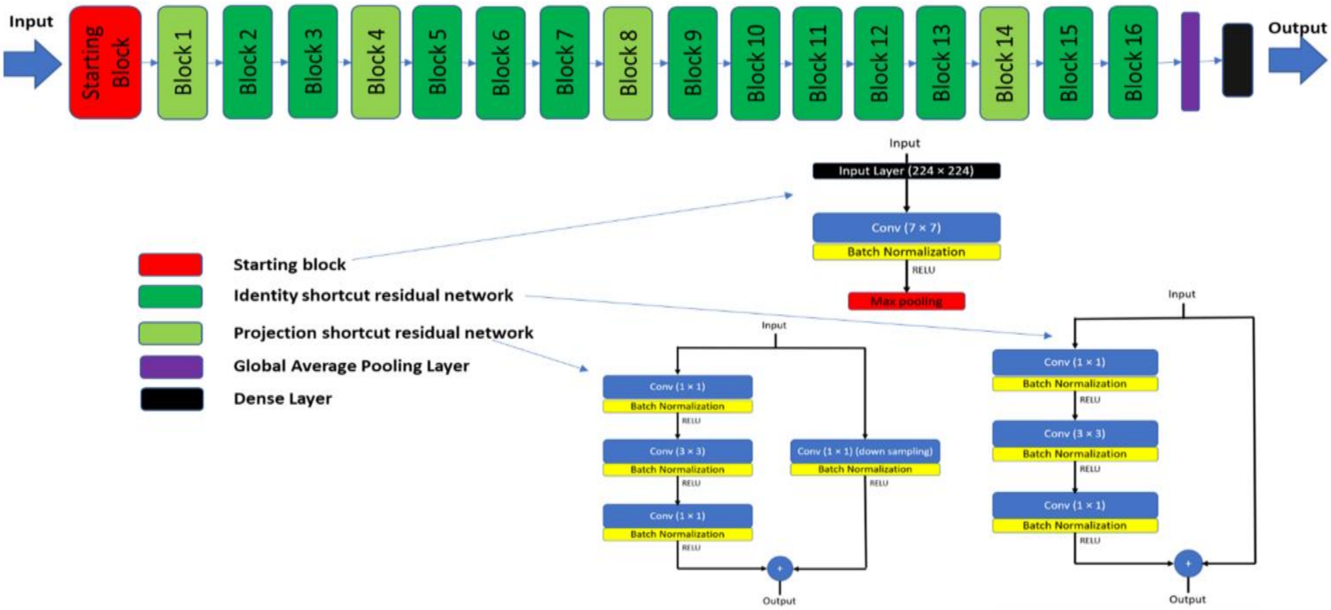


Fig. 2. ResNet layered architecture.

A modified ResNet architecture was used by Nath and Naskar [19] to automatically extract features, followed by a dense Artificial Neural network (ANN) for classification. The yielded results that exceeded 96% on sampled CASIA v2.0 to balance the classes.

Ding et al. [20] proposed a dual channel U-Net (DCU-Net), which accepts two inputs- the original tampered image and the residual tampered image. The residual image is generated by high pass filters to obtain the edges. The experimental results were shown on Casia2.0 and Columbia datasets, where the accuracy reached 97.93% and 97.27 % respectively.

The related work presents studies that either use traditional learning or deep learning approaches. With the increasing volumes of media and the advancements of editing technologies, traditional models will not provide adequate solution to the problem [21]. On the other hand, the used deep learning architectures are computationally intensive reducing their applicability on mobile real time applications [22]. In addition, further performance enhancement is needed to handle the problem.

IV. DMobile-ELA Proposed Model

A cascaded model is proposed to analyze whether images are tampered or authentic. The flow of the process model is shown in Fig. 3. The images are preprocessed applying ELA, then passed to a Dilated Mobile Net for classification.

A. *Error Level Analysis*

Error Level Analysis (ELA) is a concept that measures and visualizes the difference between an image and a re-compressed version of the same image which emphasizes certain parts that have been altered during previous edits. ELA measures the amount of error based on 8x8, relying on two main conditions applicable to JPEG images:

- A JPEG is said to be original if all 8x8 blocks have a similar error pattern. Therefore, the 8x8 pixel block can be said to have attained local minima.
- A JPEG is said to be manipulated if any 8x8 block has a higher error pattern and an 8x8 pixel block is not at its local minima.

In general, the computation of Error Level Images (ELIs) follows the formulation in Eq (1)....

$$\begin{gathered}\mathbf{I_o - I_{rc1} = ELI_1}\\ \mathbf{I_o - I_{rc2} = ELI_2}\\ \mathbf{I_o - I_{rc3} = ELI_3}\end{gathered} \tag{1}$$

where I_o denote the original image and I_{rc1} represent a recompressed image at a given rate of compression. ELI is generated through pixel-wise difference of the two images.

The resultant ELI conveys the different quality levels within an image through varying intensities. For example if the image is forged, the added regions will be compressed at a different rate than the remaining original image. Such variation will be reflected through a distinct error pattern as the forged regions will be quantized through a non-linear ratio. Thus, can be used to localize the tampered areas. In this study, three different levels of image compression were examined, namely 10%, 50%, and 90% compression.

Fig. 3 depicts the localized error pattern of the spliced person, when applying three level of compression. The images clarify the potential of ELA in locating tampered regions. As can be seen from Fig. 3, higher compression rates better localize the tampered region. A difference Error Level Image (ELI) is produced between compression rates 50% and 90% to eliminate details and detect changes. The resultant image is shown in Fig. 4. The difference image is input to the Dilated-Mobile Net.

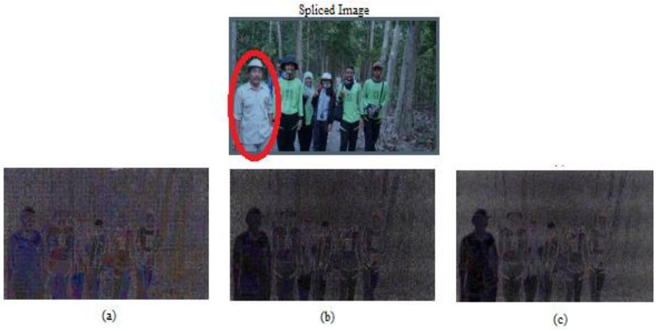


Fig. 3. ELIs at different compression rates (a) 10%, (b) 50% and (c) 90% emphasizing the spliced person (marked in red).

Fig. 4. Difference image generation from the available ELIs.

B. *Multiscale Dilated-MobileNet*

A dilated or atrous MobileNet deep learning architecture is developed for classifying images into authentic and tampered. A detailed description of the architecture will be given below.

1) Multiscale dilation : Multiple dilated filters are applied to the input differece ELI. The aim of applying filters with various dilation rates is to increase the receptive field of the filters. Expansion of the receptive field help in considering all the relevant regions in an image and and capturing all important information [23].

The dilation process inserts zeros depending on the dilation rate, hence increasing the receptive field of the filter while maintaining the number of parameters to be learnt. For example, with a dilation rate of 2 the receptive field of 3x3 filter is expanded to 5x5 convolution. Similarly, a dilation rate of three enlarges the filter to 7x7. The output of the multiscale dilation convolution is concatenated and input the first layer of of the MobileNet.

2) Light weight convolution : MobileNet is a light weight architecture known for its applicability on mobile devices [24]. It is characterized by fewer parameters, small convolution filters 3x3 and hence lower computation demand compared to other CNN architectures. MobileNet architecture employ Depthwise Separable Convolution (DSConv) Layer (shown in Fig. 5) instead of the standard convolution layer.

During depthwise separable convolution, each channel is convolved with each filter separately. The process is divided into depthwise convolution (3x3 depthwise convolution, batch normalization BN and RELU), followed by pointwise convolution (1x1 convolution, batch normalization BN and RELU). Splitting the convolution task into two steps speed up the computation task by a factor that reaches $\frac{1}{f^2}+\frac{1}{n}$, where f is the filter (kernel) size assuming squared dimensions and n is the number of filters (corresponding output channels). In addition, DSConv helps maintain a shallower network than traditional CNNs with competitive accuracies.

Despite the advantages of DSConv, the small sized convolution filters may reduce the goodness of the captured filters. Hence, the atrous filters with varying dilation rates offers a promising solution to this issue.

3) Fully connected classification: A dense layer of fully connected neurons is utilized to produce the final classification of whether the investigated image is authentic or tampered.

Overview of the network architecture is presented in Fig. 6, depicting the multiscale dilation and DSConv layers.

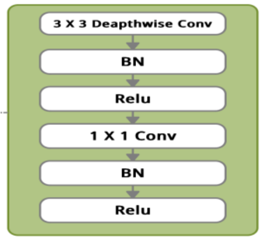


Fig. 5. Depthwise separable convolution layer structure.

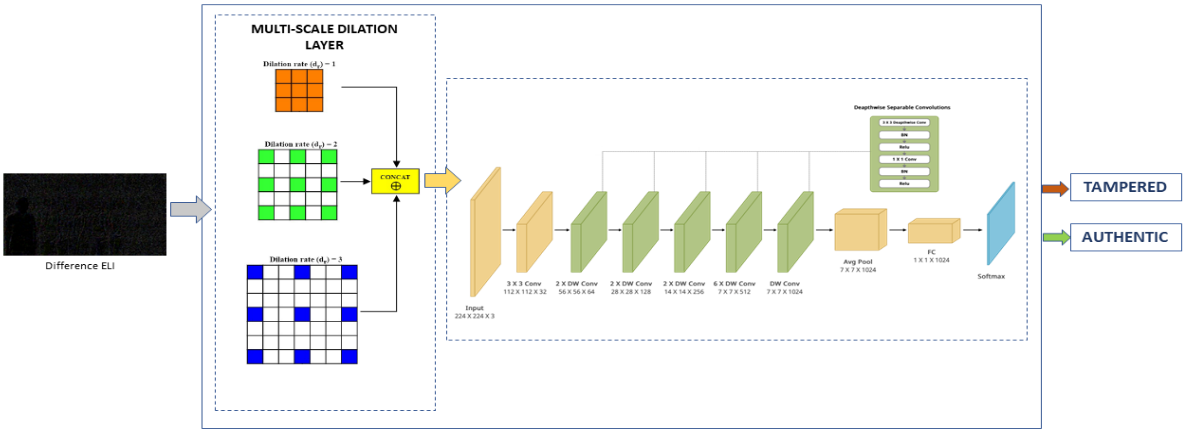


Fig. 6. Multiscale dilated MobileNet forgery classification on different ELI.

V. Results and Discussion

A briefing of the CASIAV2.0 dataset used in our experiments is given, followed by the performance measures applied to validate the performance of the proposed approach. The devised experimental setup is described for reproducibility of results. Then, the achieved results are presented and compared to recent forgery detection systems.

A. *Dataset Description*

CASIAv2.0.[25] dataset is used in the following experiments to validate the performance of DMobile-ELA. CASIAv2.0 is a benchmark dataset created by Dong et al. [25] at the Institute of Automation, the Chinese Academy of Sciences, with the purpose of aiding the research and development of image tampering detection methods. It contains 5123 tampered images and 7491 authentic images. Tampered images contain both copy-move and splicing altered images, at 3295 and 1828, respectively. This dataset is a successor of CASIAv1.0 which only included spliced images. Fig. 7 displays samples of authentic and tampered images from the dataset.

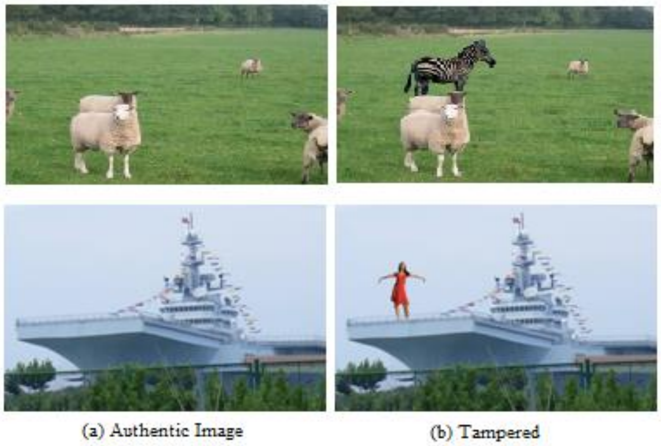

(a) Authentic Image
(b) Tampered

Fig. 7. Sample authentic and tampered images from CASIAV2.0 dataset.

B. *Performance Measures*

Four performance measures are used to evaluate DMobile-ELA and allow its comparison with recent studies. The used measures are Accuracy (Acc), Precision (P), Recall (R) and $F1_{score.}$ The computation of these measures relies on the confusion matrix given in Fig. 8.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Fig. 8. Confusion matrix outlining true positive (TP), false negative (FN), false positive (FP) and true negative (TN).

The measures are calculated according to the following Eq. (2) to (5).

$$\boldsymbol{Acc} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{2}$$

$$\boldsymbol{P} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{3}$$

$$\boldsymbol{R} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{4}$$

$$\boldsymbol{F1} = \boldsymbol{2}\,\frac{\boldsymbol{P\times R}}{\boldsymbol{P+R}} \tag{5}$$

C. *Experimental Setup*

The performance of DMobile-ELA is analyzed and compared to variable counterparts systematically. First, the performance of DMobileNet structure is contrasted to VGG16, ResNet-50 and MobileNet standard architectures. Also, the impact of transfer (pretrained on ImageNet [26]) learning or retraining from scratch is investigated. In addition, the effect of ELA on performance is elucidated through a comparison between models' performance with and without ELA. Finally, DMobile-ELA performance is compared against recent related studies.

The resolution of input images was adapted to the largest quadratic value that the MobileNet network supported which was 224x224. The default settings we used for assessing each model are splitting into 80% training and 20% validation sets, running the training for 10 epochs, and using 0.0001 learning rate. The model utilized Adam-optimizer while maintaining a batch size of 16.

D. *DMobile-ELA Performance Results*

Forgery detection accuracy is measured for VGG16, ResNet-50, MobileNet and DMobileNet on the original image without ELA preprocessing. The results are shown in Fig. 9. Also, the performance of tuning pretrained models versus retraining of the models is tested. The results show that DMobile Net attains the highest accuracy, while VGG16 scores the lowest accuracy. Another observation is that retraining is better suited to the problem under study, as there is an evident performance gap that reaches around 15% in case of ResNet-50.

The accuracy of the models with ELA preprocessing is depicted in Fig. 10. From the shown accuracies, it can be seen that ELA aided the models to score higher accuracies than without ELA with differences ranging from 5% to7% in case of retrained models. For pre-trained ResNet-50, the improvement arrived at 13%. Retrained models still presents higher accuracies compared to pre-trained models. Overall, retrained DMobile-ELA records the highest accuracy of 98.48%. The difference in accuracy between DMobileNet and MobileNet is around 3%, which is a considerable difference given that the number of parameter to be learned is the same.

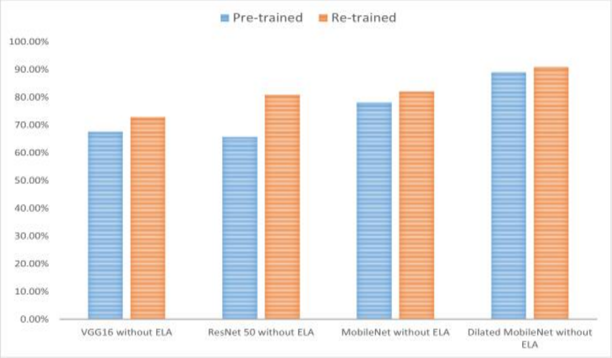


Fig. 9. Accuracy of CNN models without ELA preprocessing.

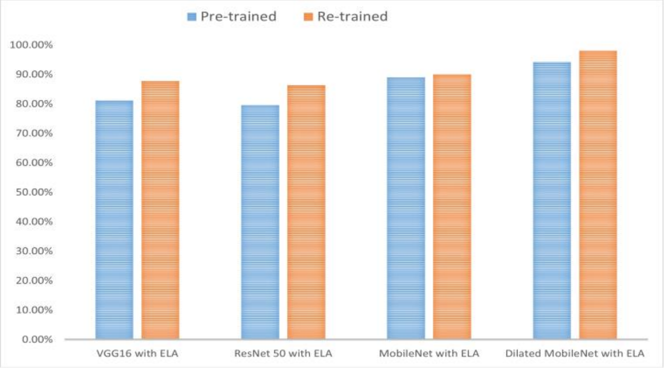


Fig. 10. Accuracy of CNN models with ELA preprocessing.

The training and validation accuracies learning curves for 10 epochs are shown in Fig. 11. The validation curve follows smoothly the training curve in the last three epochs diminishing the possibility of overfitting.

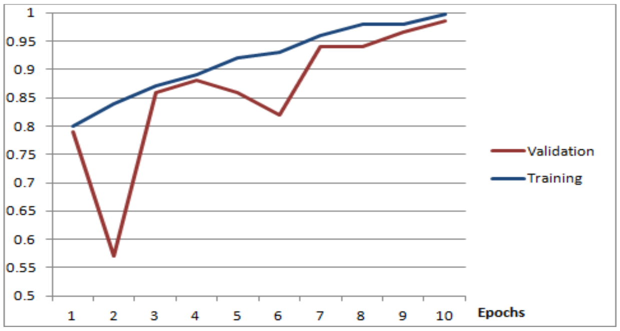


Fig. 11. **V**alidation and training accuracy learning curves of DMobile – ELA.

TABLE I. PERFORMANCE COMPARISON BETWEEN DMOBILE-ELA AND RECENT RELATED STUDIES

Approach	Performance Measures			
	Acc	*P*	*R*	*F1$_{score}$*
Alahmadi et al. [14]	0.9645	0.9669	0.9415	0.9540
Ding et al. [20]	0.9793	0.8772	0.8893	0.8667
Kanwal et al. [16]	0.9759	🟩	🟩	🟩
Niyishaka et al. [15]	0.9459	0.9000	0.9900	0.9300
A Kuznetsov [18]	0.9780	0.9710	0.9680	🟩
Proposed DMobile-ELA	0.9848	0.9781	0.9862	0.9821

TABLE II. NUMBER OF PARAMETERS COMPARISON BETWEEN DMOBILE-ELA AND RECENT RELATED STUDIES

Approach	Parameters
Ding et al. [20]	17.2M
A Kuznetsov [18]	138M
Proposed DMobile-ELA	3.4M

Table I details the measures for assessing DMobile-ELA against some of the recent studies. The metrics show that the proposed DMobile-ELA surpasses its counterparts. It presents superior P, R and F1$_{score}$ than Ding et al. [20] with a gap of around 0.1 in all these measures. Also, it scores higher accuracy, P and F1$_{score}$ compared to Niyishaka et al. [15] with differences of around 0.04, 0.07 and 0.05 respectively. Kanwal et al. [16] and Alahmadi et al. [14] offer solutions with comparable accuracy. Similarly, Kuznetsov [18] presents competitive performance in terms of all metrics. However, Kuznetsov used VGG16 for detection, which is a computationally demanding architecture. Table II outlines the number of parameters to be learned for each backbone architecture. The given numbers highlight the favorable low computation demand of DMobileNet as a light weight architecture.

VI. CONCLUSION

In this study, a forgery detection approach named DMobile-ELA is proposed. It integrates dilated MobileNet and Error Level Analysis (ELA), which leads to a lightweight high performing solution. The conducted experiments confirmed the success of DMobile-ELA in forgery detection, emphasizing the advantageous effect of ELA on performance. In addition, the experiments indicated the higher suitability of model retraining to the problem of forgery detection. Retrained DMobile-ELA performance reached Acc, P, R and F1$_{score}$ of 0.9848, 09781, 0.9862 and 0.9821 respectively on CASIAv2.0 dataset. Further improvements can be applied such as integrating different preprocessing procedures and merging textural features. Also, forgery types other than copy-move and splicing can be investigated to increase the applicability scope of the proposed approach.

REFERENCES

[1] F. Marcon, C. Pasquini, and G. Boato, “Detection of Manipulated Face Videos over Social Networks: A Large-Scale Study,” J. Imaging, vol. 7, no. 10, 2021.

[2] M. M. Eltoukhy, M. Elhoseny, K. M. Hosny, and A. K. Singh, “Computer aided detection of mammographic mass using exact Gaussian–Hermite moments,” J. Ambient Intell. Humaniz. Comput., 2018.

[3] A. Ross, S. Banerjee, and A. Chowdhury, “Security in smart cities: A brief review of digital forensic schemes for biometric data,” Pattern Recognit. Lett., vol. 138, pp. 346–354, 2020.

[4] W. Luo, Z. Qu, F. Pan, and J. Huang, “A survey of passive technology for digital image forensics,” Front. Comput. Sci. China, vol. 1, no. 2, pp. 166–179, 2007.

[5] K. D. Kadam, S. Ahirrao, and K. Kotecha, “Efficient Approach towards Detection and Identification of Copy Move and Image Splicing Forgeries Using Mask R-CNN with MobileNet V1,” Comput. Intell. Neurosci., vol. 2022, p. 6845326, 2022.

[6] M. Westerlund, “The Emergence of Deepfake Technology: A Review,” Technol. Innov. Manag. Rev., vol. 9, pp. 39–52, 2019.

[7] P. Sharma, M. Kumar, and H. Sharma, “Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: an evaluation,” Multimed. Tools Appl., 2022.

[8] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, “A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning,” Arch. Comput. Methods Eng., vol. 27, no. 4, pp. 1071–1092, 2020.

[9] J. Heaton, Artificial Intelligence for Humans: Deep learning and neural networks. Heaton Research, Incorporated, 2015.

[10] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016, pp. 770–778.

[12] X. Shen, Z. Shi, and H. Chen, “Splicing image forgery detection using textural features based on the grey level co-occurrence matrices,” IET Image Process., vol. 11, pp. 44–53, 2017.

[13] J. G. Han, T. H. Park, Y. H. Moon, and I. K. Eom, “Efficient Markov feature extraction method for image splicing detection using maximization and threshold expansion,” J. Electron. Eng., vol. 25, pp. 21 –30, 2016.

[14] A. Alahmadi, M. Hussain, H. Aboalsamh, G. Muhammad, G. Bebis, and H. Mathkour, “Passive detection of image forgery using DCT and local binary pattern,” Signal, Image Video Process., vol. 11, no. 1, pp. 81–88, 2017.

[15] P. Niyishaka and C. Bhagvati, “Image splicing detection technique based on Illumination-Reflectance model and LBP,” Multimed. Tools Appl., vol. 80, no. 2, pp. 2161–2175, 2021.

[16] N. Kanwal, A. Girdhar, L. Kaur, and J. S. Bhullar, “Digital image splicing detection technique using optimal threshold based local ternary pattern,” Multimed. Tools Appl., vol. 79, no. 19, pp. 12829–12846, 2020.

[17] Y. Rao and J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images,” in 2016 IEEE International Workshop on Information Forensics and Security (WIFS), 2016.

[18] A. Kuznetsov, “Digital image forgery detection using deep learning approach,” J. Phys. Conf. Ser., vol. 1368, pp. 28 – 32, 2019.

[19] S. Nath and R. Naskar, “Automated image splicing detection using deep CNN-learned features and ANN-based classifier,” Signal, Image Video Process., vol. 15, no. 7, pp. 1601–1608, 2021.

[20] H. Ding, L. Chen, Q. Tao, Z. Fu, L. Dong, and X. Cui, “DCU-Net: a dual-channel U-shaped network for image splicing forgery detection,” Neural Comput. Appl., vol. 35, no. 7, pp. 5015–5031, 2023.

[21] A. L’Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, “Machine Learning With Big Data: Challenges and Approaches,” IEEE Access, vol. 5, pp. 7776–7797, 2017.

[22] V. Isuyama and B. Albertini, “Comparison of Convolutional Neural Network Models for Mobile Devices,” in Anais do XX Workshop em Desempenho de Sistemas Computacionais e de Comunicação, 2021, pp. 73–83.

[23] W. Luo, Y. Li, R. Urtasun, and R. Zemel, “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks,” in Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 4905–4913.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv Prepr., vol. arXiv:1704, 2017.

[25] J. Dong, W. Wang, and T. Tan, “CASIA Image Tampering Detection Evaluation Database,” in 2013 IEEE China Summit and International Conference on Signal and Information Processing, 2013, pp. 422–426.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

A Comprehensive Study on Medical Image Segmentation using Deep Neural Networks

Loan Dao, Ngoc Quoc Ly
Dept. of Computer Vision and Cognitive Cybernetics
University of Science, VNUHCM, Ho Chi Minh, Vietnam
Viet Nam National University, Ho Chi Minh City, Vietnam

***Abstract*—Over the past decade, Medical Image Segmentation (MIS) using Deep Neural Networks (DNNs) has achieved significant performance improvements and holds great promise for future developments. This paper presents a comprehensive study on MIS based on DNNs. Intelligent Vision Systems are often evaluated based on their output levels, such as Data, Information, Knowledge, Intelligence, and Wisdom (DIKIW), and the state-of-the-art solutions in MIS at these levels are the focus of research. Additionally, Explainable Artificial Intelligence (XAI) has become an important research direction, as it aims to uncover the "black box" nature of previous DNN architectures to meet the requirements of transparency and ethics. The study emphasizes the importance of MIS in disease diagnosis and early detection, particularly for increasing the survival rate of cancer patients through timely diagnosis. XAI and early prediction are considered two important steps in the journey from "intelligence" to "wisdom." Additionally, the paper addresses existing challenges and proposes potential solutions to enhance the efficiency of implementing DNN-based MIS.**

Keywords*—Medical image segmentation (MIS); SOTA solutions in MIS; XAI; early disease diagnosis

I. Introduction

Computers store images as grids of pixels, each containing a color value. These digital images are considered unstructured data, and image segmentation is the process of partitioning pixels into separate regions that correspond to a single object or class. This is accomplished by labeling each pixel with its corresponding class.

Image segmentation provides a deeper understanding of the structure of an image and is a crucial processing step in many image and video applications. It serves as the foundation for other challenges, such as object detection, image classification, and image analysis.

In medical imaging, image segmentation involves separating organs, disease regions, tumors, or anomalies to assist in diagnosis, detect pathology, and monitor the progression of diseases. MIS is a challenging task because of the slow grayscale variation in medical images, making it difficult to distinguish objects.

Currently, AI applications have reached the "Intelligence" level in DIKIW [1], which stands for Data, Information, Knowledge, Intelligence, and Wisdom. In MIS, “Intelligence” is demonstrated through highly accurate segmentation results, even on low-contrast, blurry, noisy images [2]. Additionally, "Intelligence" in MIS is not only demonstrated through the ability to segment organs, but also through the ability to segment lesions [2].

"Wisdom", the highest level in the DIKIW hierarchy, represents humanity's ultimate goal. In MIS systems, the output not only provides segmented medical images but also offers an explanation for the segmentation results through eXplainable Artificial Intelligence (XAI). This helps to increase trust in the results among both doctors and patients, satisfying the demands for transparency and medical ethics, which is a critical aspect in realizing the practical application of AI-based disease diagnosis. Transparent segmentation results also contribute to early disease diagnosis, leading to improved treatment, monitoring, and healthcare processes. In healthcare systems, predicting future health conditions to plan appropriate care and potentially reducing the risk of death holds significant humanitarian significance. XAI and early prediction are two crucial steps towards bridging the "Intelligence-to-Wisdom" gap.

The following major contributions are presented:

- The state-of-the-art solutions in MIS focus on key factors such as network architecture, data, loss function, and evaluation metrics. The paper specifically explains the development process of the network architecture from the perspective of three levels of an intelligent vision system: the backbone, the typical network architecture, and applications of MIS.
- The paper focuses on the state-of-the-art solutions in MIS and highlights the current interest in XAI-based MIS to meet ethical and legal requirements.
- The paper presents a new perspective on MIS by incorporating the capability of making early predictions based on the results, which contributes to the improvement of community healthcare systems.

The rest of the paper is organized as follows: In Section II the state-of-the-art solutions for deep neural network-based medical image segmentation are presented. Section III focuses on the explanation of black-box models using eXplainable Artificial Intelligence (XAI) to increase trust among end-users. The paper analyzes early prediction techniques for disease progression in Section IV. The challenges and proposed solutions for improving the efficiency of future clinical applications are discussed in Section V. Finally, the conclusions of this comprehensive study and suggestions for future research are presented in Section VI.

II. State-of- the-Art Solutions in Medical Image Segmentation

This section presents state-of-the-art solutions in MIS, including network architectures, data, loss functions, and evaluation metrics. The specifics are depicted in Fig. 1.

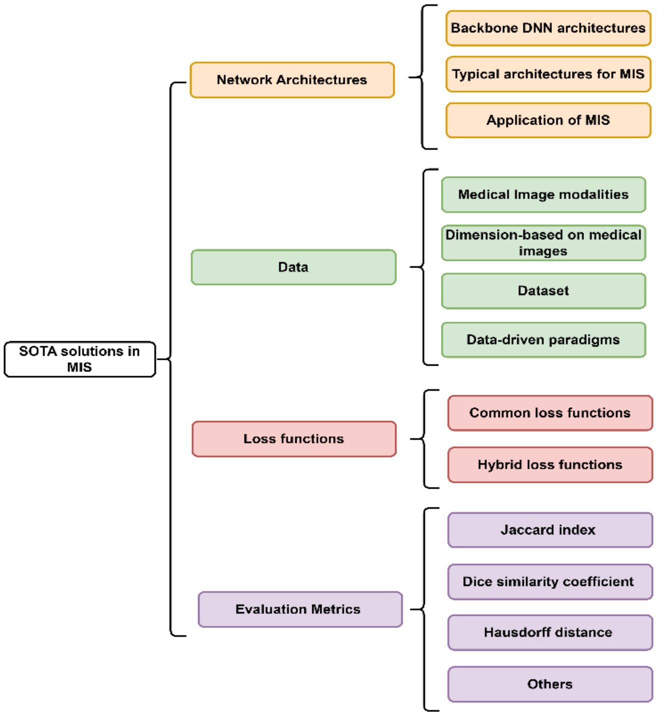


Fig. 1. The pipeline of SOTA solutions in MIS.

A. Network Architectures

This study conducts a survey of state-of-the-art solutions in MIS based on the standard framework of intelligent vision systems (IVS). An IVS encompasses three main levels.

Level 1 encompasses the backbone deep neural network (DNN) architectures used in image segmentation.

Level 2 builds on Level 1 to create specialized image segmentation models.

Level 3 leverages the knowledge from Levels 1 and 2 to develop typical applications for medical image segmentation.

This study thoroughly describes and explains the three levels in the IVS framework (TABLE I.)

In the following sections, typical modules and architectures for each level will be thoroughly examined.

Level 1: Background DNN architectures:

This section introduces typical background neural network architectures and state-of-the-art modules aimed at enhancing the efficiency of segmentation.

1) The typical background network architectures: The paper delves into several typical background architectures, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Graph Neural Networks (GNNs), Generative Adversarial Networks (GANs), and Transformers.

- Convolutional neural network (CNN) (1982, [3]). Before gaining popularity, Convolutional Neural Networks (CNNs) have gone through several historical stages. The first foundation for convolution was laid by K. Fukushima et al. through a series of works, including "Cognitron: A self-organizing multilayered neural network" (1975), “Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron” (1979), “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position” (1982). Since neighboring pixels in an image typically have strong inter-dependencies, K. Fukushima introduced the concept of "connectable areas" to extract features in "neighborhoods" instead of fully connecting the layers. This marked the first paradigm for unsupervised pattern recognition. The breakthrough of Convolutional Neural Networks (CNNs) lies in the use of two-dimensional filters, which are capable of extracting meaningful features from locally connected subsamples. These filters have smaller size, resulting in more optimal computing and storage capacity compared to earlier fully connected networks. The output from the filters is fed into an activation function, which adds non-linearity to the feature space and can be learned during training. Additionally, the non-linear activation function generates outputs that are typically monitored through subsampling. This allows for the aggregation of outputs, making the input insensitive to geometric deviations such as distortion, resizing, and repositioning of the sample input. CNNs have been successful in overcoming challenges in MIS, such as noise, blur, and low contrast. CNN architecture has several different backbones that are used in MIS, such as VGG (2014, [4]), ResNet (2015, [5]), DenseNet, DeepLabv3, MobileNets (2017, [6][7][8]), EfficientNet (2019, [9]).

- Recurrent Neural Networks (RNN) (1989, [10]) Barak A. Pearlmutter has explored various approaches to constructing the foundational concept of continuous time-recurrent networks. The Recurrent Neural Network (RNN) aims to mitigate the weight gradient to minimize the time-orbit error of the states in the continuous regression network. RNNs are particularly well-suited for continuous time domains such as signal, control, and speech processing. In medical image segmentation, RNN has been applied to model the time-dependence of image sequences (video). By leveraging the relationships between space-time information, RNNs can be combined with other architectures to improve the accuracy of image segmentation. RNNs can capture both local and global spatial features of the image by considering the context information [11].

TABLE I. THE COMMON FRAMEWORK OF INTELLIGENT VISION SYSTEMS (IVS) FOR MEDICAL IMAGE SEGMENTATION (MIS)

	Level of IVS	Explanation	Specific Solutions
1	**Backbone DNN Architectures**	Traditional segmentation methods, such as thresholding, edge detection, and region-based techniques, have limitations when dealing with noisy, fuzzy, and low-contrast medical images. Convolutional Neural Networks (CNNs) alleviate these drawbacks. Additionally, Graph Neural Networks (GNNs) and Transformers address the limitations of convolution kernels with regards to locality and invariance	**Background Networks Architectures:** ***CNN*** (1982, [3]), ***RNN*** (1989, [10]), ***GNN*** (2009, [12]), ***GAN*** (2014, [14]), ***Transformer*** (2017, [16]**).** **Backbone DNN**: VGG (2014, [4]), ResNet (2015, [5]), DenseNet, DeepLabv3, MobileNets (2017, [6][7][8]), EfficientNet (2019, [9]). **Some additional modules:** Inception (2015, [19]), Dilation convolution (2016, [21]), ASPP (2017, [7]), Attention (2015, [29]), Squeeze-and-excitation (2018, [33]), Residual (2015, [5]), Dense (2017, [6]), Residual Dense (2018, [37]).
2	**Specific DNN Architectures for MIS**	UNet is a widely used method in medical image segmentation. Its variants and hybrid network architectures have been developed to achieve higher accuracy in segmentation	***Unet*** (2015, [38]), ***V-net***, 3D U-Net (2016, [39], [40]), Mask R-CNN (2017, [41]), ***U-Net++*** (2018, [42]), UNet 3+, DRU-Net, DoubleU-net (2020, [43] – [45]), ***TransUNet,*** Swin Transformer (2021, [46], [47]), UNETR++, LE-UDA, ***TransUNet+*** (2022, [48], [49], [50]), SEP (2022, [51]), ESFPNet (2022, [52])
3	**Applications of MIS**	Medical image segmentation has evolved from single-organ segmentation to multi-organ segmentation and from organ segmentation to lesion segmentation. This has allowed for a more comprehensive and effective diagnosis of diseases, thereby improving human health care.	***Kidney Tumor Segmentation*** (F.Isensee et al., 2019, [53]), Brain Tumor Segmentation (S. Li et al. 2021, [54]), COVID-19 infection localization and severity grading from chest X-ray images (A. M. Tahir et al, 2021, [2][1]), ***Abdominal Multi-Organ Segmentation*** (F.Isensee, et al., 2022, [55]), and so on.

- Graph Neural Network (GNN) (2009, [12]) Scarselli et al. introduced the Graph Neural Network (GNN) architecture, which expands upon traditional neural network methods for processing data represented in graph regions. Geometric Deep Learning, also known as GNN, is a nascent field of study that extends deep neural modeling to non-Euclidean domains. The structures of medical images often have irregular and unordered patterns, making it challenging to represent them as matrices for CNNs. As a result, graph-based representations are becoming increasingly popular in MIS [13].

- Generative Adversarial Networks (GAN) (2014, [14]) The Generative Adversarial Network (GAN) architecture was introduced by Goodfellow et al. based on game theory. In this architecture, two players, a generator and a discriminator, play against each other to minimize their respective costs. The discriminator's cost encourages it to correctly classify data as real or fake, while the generator's cost encourages it to generate the most realistic fake samples that the discriminator finds difficult to distinguish. In medical image segmentation, GANs can be used to create synthetic medical images and their corresponding segmented masks, leading to improved segmentation accuracy thanks to GAN's powerful generation ability and ability to capture the data distribution [15][14].

- Transformer (2017, [16]): The Transformer architecture was originally designed for natural language processing (NLP) tasks, where it achieved remarkable improvements. Its success in NLP has drawn the attention of the computer vision community. The Transformer enables parallel processing of input sequences while supporting long-term dependencies between sequence elements, thus overcoming the explicit long-term dependency limitations of the Unet model [17]. Transformers, unlike CNNs, are designed with less inductive bias and can fit into any data structure as easily as established functions. Their fundamental structure also demonstrates great scalability, making them suitable for networks with high capacity and large data sets. This allows for multi-modal processing, including images, videos, text, and audio, using the same processing blocks. For medical imaging, organs that are frequently spread across a large receptive field can be efficiently encoded by modeling relationships between distant pixels. Therefore, the ability of Transformers to model the global context is crucial for accurate medical image segmentation, such as lung segmentation. Additionally, medical images are often blurred, noisy, and have low contrast, such as in ultrasound scans. Understanding the overall context between pixels against the background can help models avoid mis-segmentation [18].

2) SOTA modules for enhancing segmentation efficiency

- For aggregating features at multiple scales:
  - The Inception module (2015, [19]) which concatenates multiple parallel convolutional filter banks with varying kernel sizes to extract features at multiple scales. Example of the application of the Inception module in medical image segmentation can be found in [20].
  - The dilation convolution (or "atrous convolution") kernel, introduced in 2016 [21], increases the size of the kernel, and the corresponding receptive field without significantly increasing the number of pixels processed. This results in improved speed and accuracy. Example of the use of the dilation module in medical image segmentation networks can be found in [22].
  - The Atrous Spatial Pyramid Pooling (ASPP, [7]) module, introduced in 2017, utilizes dilated (or atrous) convolution to gather information at multiple scales. This helps to preserve local features while capturing multi-scale contextual information, leading to improved segmentation

efficiency. ASPP module is applied in medical image segmentation such as retinal segmentation [23], segmentation of abdominal organs from CT images [24], SAR-U-Net liver segmentation from CT images [25], U-Net-ASPP segmented COVID-19 [26], localized skin lesions [27]; spinal segments [28]

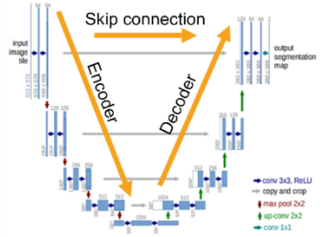
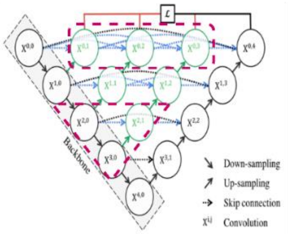
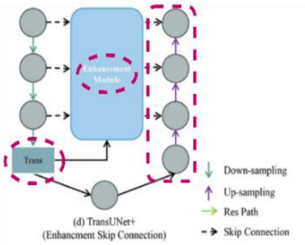
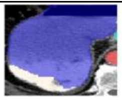
- Focusing on important features:
  - The attention mechanism (2015, [29]) focuses on spatially significant features. This mechanism is commonly applied in medical image segmentation problems such as [30] - [32]
  - Squeeze-and-excitation block (2018, [33]) focus on features based on channel-weighted adjustment. Medical image segmentation problems that apply this block like [34] - [36].
- Connecting to the previous layers and solving the vanishing gradient problem:
  - Residual block (2015, [5]), which adds the previous layer outputs to feature maps learned from the current layer;
  - Dense block (2017, [6]), which connects the outputs of all previous layers to the feature maps learned by the current layer;
  - Residual Dense block (2018, [37]), which allows full using the local and global features.

Level 2: Specific DNN Architecture for MIS.

Background network architectures at level 1 such as CNN, GNN, transformer, and so on can be used to implement various tasks such as detection, recognition, classification, and segmentation. This paper focuses on DNN architectures for **image segmentation.** TABLE II. compares three specific networks for the medical imaging segment, UNet (2015, [38]), UNet++ (2018, [42]), and TransUNet+ (2022, [50]) in terms of network architecture, pros, cons, and performance.

From the comparison results, it can be observed that network architectures are continuously improving in terms of performance. The current trend is the use of hybrid networks, such as TransUNet+ [50], to meet the increasing demand in healthcare systems.

TABLE II. COMPARES THREE SPECIFIC NETWORKS FOR THE MEDICAL IMAGING SEGMENT

	Unet (2015, [38])	**Unet++ (2018, [42])**	**TransUNet+ (2022, [50])**
Architectures	 Skip connection Encoder Decoder input image tile output segmentation map conv 3x3, ReLU copy and crop max pool 2x2 up-conv 2x2 conv 1x1 Unet includes **encoder** to extract features, **decoder** synthesize extracted features to segmentation results and **skip connection** copies low-resolution (encoder) to high-resolution (decoder) feature maps.	 $\mathcal{L}$ $X^{0,0}$ $X^{0,1}$ $X^{0,2}$ $X^{0,3}$ $X^{0,4}$ $X^{1,0}$ $X^{1,1}$ $X^{1,2}$ $X^{1,3}$ $X^{2,0}$ $X^{2,1}$ $X^{2,2}$ $X^{3,0}$ $X^{3,1}$ $X^{4,0}$ Backbone Down-sampling Up-sampling Skip connection $X^{i,j}$ Convolution Uses dense blocks to **Re-designed skip pathways** and use **Deep supervision**	 Enhancement Module Trans (d) TransUNet+ (Enhancment Skip Connection) Down-sampling Up-sampling Res Path Skip Connection **Encoder**: CNN and Trans**; decoder:** original decoder and the enhanced features; **skip connection:** enhancement module.
Pros	- Train with fewer annotated images (at most 35 annotated images) - Fast training time in 2015 (On a NVidia Titan GPU (6 GB), segmenting a 512x512 image takes less than a second.). - Easily applied to more tasks.	- Added redesigned skip pathways and deep supervision for ***more precise segmentation, model pruning, and increased speed.***	- Combining Transformer (***global self-attention mechanisms***) and CNN (enhance finer details by recovering ***localized spatial information***), Redesigning the skip connection to ***enhance features and improve the focus on the key patches***. In the decoder, cascaded up-sampler contains an up-sampling layer and a linear layer. - Performance in small organ segmentation.
Cons	- Limits on network depth and skip connection - Limitations on long-range information extraction	- Reduces the robustness of feature representation and increases the number of parameters.	- High computational cost and memory usage
Performances	 IoU: 76.62 (**<u>LiTS Challenge</u>**)	 **IoU: 82.90** **(<u>LiTS Challenge</u>)**	
		 DSC 76.09 (**<u>Synapse multi-organ CT dataset</u>**)	 **DSC: 81.57** **(<u>Synapse multi-organ CT dataset</u>)**

The paper not only compares three of the most specific network architectures for MIS, but it also tracks the latest MIS rankings on paperswithcode. According to the latest updates, the two top models, SEP (2022, [51]) and ESFPNet-L (2022, [52]), are at the forefront on two datasets, Kvasir-SEG and CVC-ClinicDB, respectively.

Spatially Exclusive Pasting (SEP) (2022, [51]) is an innovative data augmentation technique designed to tackle the issue of data scarcity in polyp segmentation, an important task in the diagnosis of intestinal diseases such as tumors and precancerous lesions.

Fig. 2 illustrates the procedure of SEP technique. The core concept of this technique is to copy the polyp region and paste it to other locations in order to generate a large number of new images. The augmentation process is divided into three modules: (1) a Potential Map Generation Module that generates a potential value for each coordinate, (2) a Pasting Module, and (3) an Update Module that updates the potential values for each coordinate.

According to the latest statistics on the rankings of paperswithcode.com (Fig. 3), SEP has achieved the highest mean Dice score (0.941) surpassing UNet (0.818), UNet++ (0.821), and FCBFormer (0.939).

The limitation of SEP is that it is only applicable to a limited number of data sources and is specifically designed for the task of polyp segmentation.

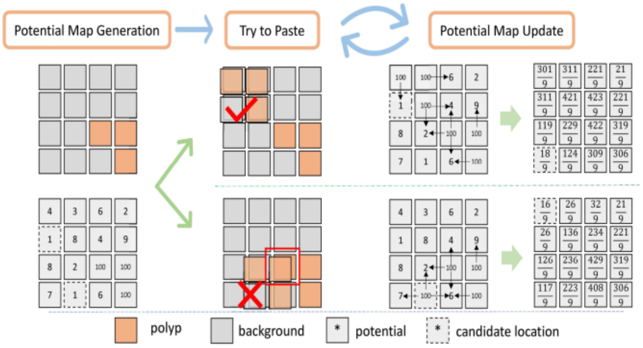


Fig. 2. The process of SEP [51].

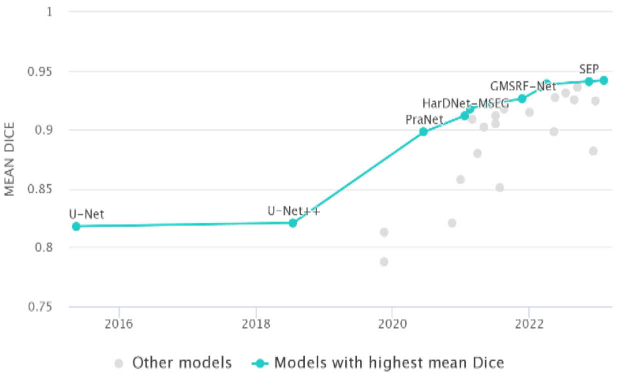


Fig. 3. Leaderboard the models with the highest mean Dice on the Kvasir-SEG.

ESFPNet (2022, [52]) is a deep learning architecture designed for real-time accurate segmentation and robust detection of bronchial lesions in autofluorescent bronchoscopy (AFB) video streams. Fig. 4 depicts the architecture of ESFPNet, which consists of a pre-trained Mix Transformer (MiT) encoder that leverages the encoder structure and an efficient Intelligent Phased Feature Pyramid (ESFP) decoder structure. ESFPNet-L produces superior results, with a mean Dice score of 0.949, compared to other recent architectures such as DuAT (0.948) and ColonFormer (0.947). Fig. 5 displays the rankings of the models with the highest mean Dice scores on the CVC-ClinicDB dataset. Additionally, with a processing speed of 27 frames per second, ESFPNet provides clinicians with a useful tool for confidently segmenting and detecting lesions in real-time during direct airway bronchoscopy. However, one drawback of ESFPNet is the high cost and difficulty in acquiring more live human video data.

Level 3: Applications of MIS

Level 3 is dedicated to the investigation of specific applications designed to perform medical image segmentation. This inherits and develops the background and backbone of the Deep Neural Network (DNN) architectures from Level 1 with a specific DNN architecture designed for image segmentation processing at Level 2.

According to the estimate of cancer cases in the United States in 2022, the most common types of cancer were prostate (268,490 cases in men) and breast (287,850 cases in women). The second most common cancers were lung and bronchus (117,910 cases in men and 118,830 cases in women) [56].

TABLE III. lists the number of papers with codes corresponding to each task, based on the latest statistics (as of November 2022) available on paperswithcode.com in the "Browse SoTA > Medical > Medical Image Segmentation" section.

Furthermore, the paper considers papers that have won first place in challenges at the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)[1] in the past three years (2019 - 2022). TABLE IV. summarizes these challenges [57]

According to the research results, it is evident that state-of-the-art (SOTA) applications are focusing on improving segmentation performance, from single organ segmentation to multi-organ segmentation (abdomen, brain, genital organs, etc.), and from organ segmentation to tumor and infection area segmentation, to provide comprehensive care for human health.

B. Data

1) Medical image modalities: Data is a crucial component in the learning process. Medical imaging is a method and approach to create visual images of the interior of the body using non-invasive technologies [58]. It is used to aid in the diagnosis or treatment of various diseases. Some common

[1] https://miccai.org/

medical imaging techniques include X-rays, computed tomography (CT) scans, magnetic resonance imaging (MRI), ultrasound (US), positron emission tomography (PET), and single-photon emission computed tomography (SPECT), among others [59]. This paper explores the four most commonly used medical imaging modalities—X-ray, CT, MRI, and Ultrasound. TABLE V. compares these four imaging modalities in terms of their advantages, disadvantages, SOTA applications, and health effects [60].

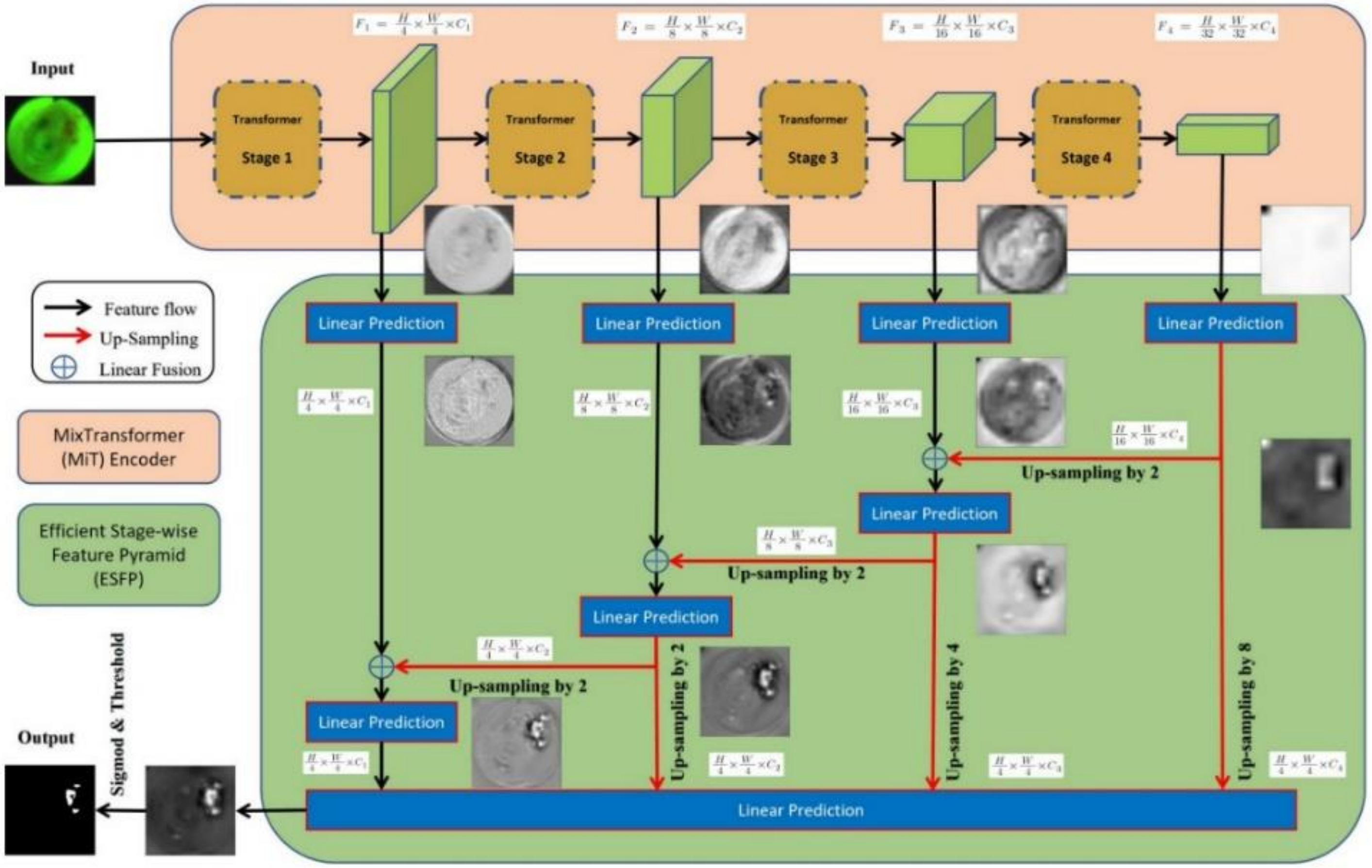

Input
$F_1 = \frac{H}{4} \times \frac{W}{4} \times C_1$
$F_2 = \frac{H}{8} \times \frac{W}{8} \times C_2$
$F_3 = \frac{H}{16} \times \frac{W}{16} \times C_3$
$F_4 = \frac{H}{32} \times \frac{W}{32} \times C_4$
Transformer Stage 1
Transformer Stage 2
Transformer Stage 3
Transformer Stage 4
Feature flow
Up-Sampling
Linear Fusion
MixTransformer (MiT) Encoder
Efficient Stage-wise Feature Pyramid (ESFP)
Linear Prediction
Up-sampling by 2
Up-sampling by 4
Up-sampling by 8
Sigmod & Threshold
Output

Fig. 4. ESFPNet architecture [52].

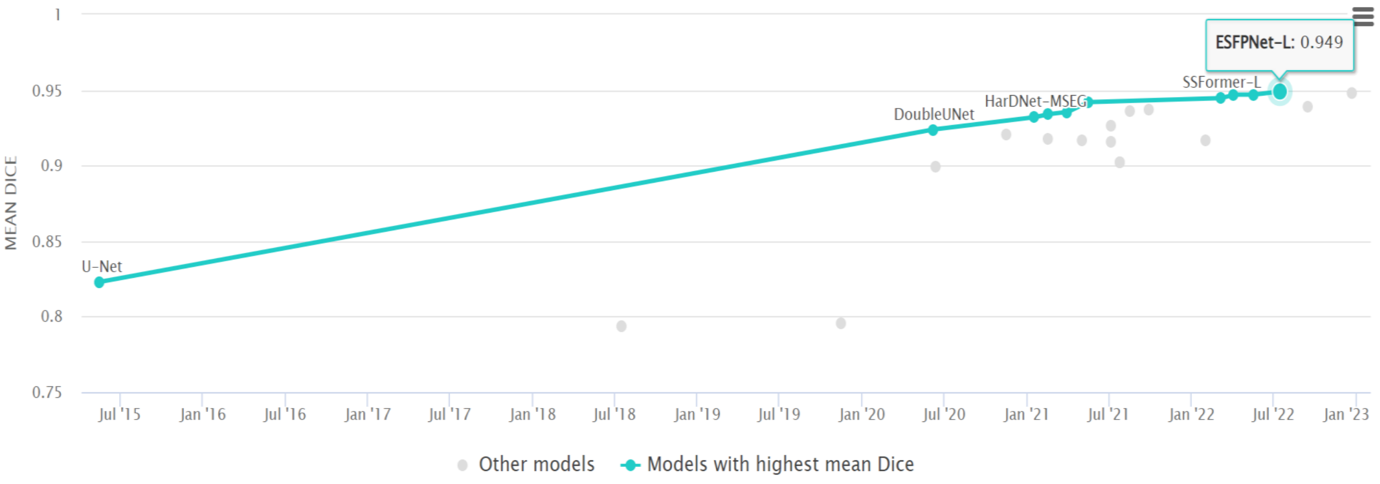


Fig. 5. Leaderboard the models with the highest mean Dice on the CVC-ClinicDB.

TABLE III. Statistics of Number of Papers with Code on Paperswithcode.Com[2]

	Segmentation tasks	Benchmarks	Papers with code		Segmentation tasks	Benchmarks	Papers with code
1.	Medical Image	104	407	**17.**	Video Polyp	4	11
2.	Lesion	8	142	**18.**	COVID-19 Image		10
3.	Brain Tumor	9	94	**19.**	Lung Nodule	5	8
4.	Brain	1	51	**20.**	Nuclear	1	8
5.	Cell	8	41	**21.**	Skin Cancer	2	8
6.	Skin Lesion	2	39	**22.**	Electron Microscopy Image	3	7
7.	Retinal Vessel	4	36	**23.**	Infant Brain Mri	1	5
8.	MRI		32	**24.**	Brain Lesion From Mri		5
9.	3D Medical Image	3	28	**25.**	Ischemic Stroke Lesion		4
10.	Cardiac		26	**26.**	Automatic Liver And Tumor		3
11.	Liver	1	23	**27.**	Placenta		3
12.	Semi-supervised MIS	2	17	**28.**	Acute Stroke Lesion		1
13.	Brain Image Segmentation	6	14	**29.**	Cerebrovascular Network		1
14.	Volumetric MIS	1	12	**30.**	Automated Pancreas		1
15.	Pancreas	2	12	**31.**	Semantic Segmentation of Orthoimagery		1
16.	Iris	3	12	**32.**	Pulmonary Vessel		1

TABLE IV. Summary of Challenges at the International Conference MICCAI in the Last Three Years (2019 - 2022)[3] [57]

	Challenges	First Author	Title
1.	2022 MICCAI: Multi-Modality Abdominal Multi-Organ Segmentation Challenge (AMOS22) (Results)	Fabian Isensee, Constantin Ulrich and Tassilo Wald	Extending nnU-Net is all you need (paper) (code)
2.	2021 ISBI: MitoEM Challenge: Large-scale 3D Mitochondria Instance Segmentation (MitoEM) (Results)	Mingxing Li	Advanced Deep Networks for 3D Mitochondria Instance Segmentation (paper) (code)
3.	2021 MICCAI: Fast and Low GPU memory Abdominal oRgan sEgmentation (FLARE) (Results)	Fan Zhang	Efficient Context-Aware Network for Abdominal Multi-organ Segmentation (paper) (code)
4.	2021 MICCAI: Kidney Tumor Segmentation Challenge (KiTS) (Results)	Zhaozhong Chen	A Coarse-to-fine Framework for The 2021 Kidney and Kidney Tumor Segmentation Challenge (paper)
5.	2020 MICCAI: Automatic Evaluation of Myocardial Infarction from Delayed-Enhancement Cardiac MRI (EMIDEC)	Yichi Zhang	Cascaded Convolutional Neural Network for Automatic Myocardial Infarction Segmentation from Delayed-Enhancement Cardiac MRI (arxiv)
6.	2019 MICCAI: Kidney Tumor Segmentation Challenge (KiTS19)	Fabian Isensee	Automated Design of Deep Learning Methods for Biomedical Image Segmentation (arxiv).

[2] https://paperswithcode.com/area/medical/medical-image-segmentation

[3] https://github.com/JunMa11/SOTA-MedSeg

TABLE V. COMPARISON BETWEEN THE MEDICAL IMAGING MODALITIES

	X-ray	CT	MRI	US
Advantages	Low cost Fast imaging time.	Quick imaging Excellent spatial resolution It is possible to combine it with angiographic techniques.	There is no ionizing radiation. Exceptional spatial resolution Outstanding soft tissue contrast Dynamic angiographic imaging.	Low cost and real-time nature Fast imaging time No ionizing radiation, good spatial resolution More prevalent, portability.
Disadvantages	Ionizing radiation Low sensitivity.	Ionizing radiation Low sensitivity Limited soft tissue contrast.	High cost Long imaging time Contraindications in some patients	Operator dependent The difficulty of distinguishing imaging structures between tissue and gas Noise, shadow, speckle, low contrast, and blurred edges Limited penetration/sensitivity
Applications of MIS	Bone [61], Lung [2], [62], Caries lesion [63]	Lung [42], [64] - [66], Proximal femur segmentation [67], Kidney tumor segmentation [68], tooth and alveolar bone segmentation [69]	Brain [66], [70] - [72], Retinal Vessel [73] - [75], Cardiac [66], [76] – [78], prostate [66], [39], [78], Osteosarcoma [79]	Breast [80], kidney [81], prostate [82], multi-organs [83]
Health effects	Biological effect, need protection against unnecessary does.	High radiation, dangerous to health.	Less harmful effects, better for the fetus.	Safe, painless, non-invasive and non-ionized.

Some comments on medical imaging modalities:

- Regarding the segmentation problem: Due to challenges such as noise, shadow, speckling, low contrast, and blurred edges, US images pose more difficulties in segmentation compared to MRI and CT images
- Regarding health effects: X-ray and CT imaging have various negative impacts on human health.
- Regarding segmentation applications: Multiple medical imaging modalities can be utilized for various health care applications to provide comprehensive health care for individuals

2) MIS in different dimensionality of medical images: This section provides an overview of 2D, 3D, and 4D data types in medical imaging, as well as image segmentation issues in these three data types.

- Structured 2D images have a defined height and width and exist in a flat space. The most common type of medical image in this category is X-rays. This paper focuses on examining segmentation models, such as 2D CNNs, original UNets, and their variants, that are applicable to 2D image.
- Structured 3D images have a defined height, width, and depth, and can be considered as a collection of stacked 2D frames. This type of medical imaging, which utilizes spatial relationships, is commonly used in modalities such as CT and MRI scans. 3D imaging is widely used in clinical practice due to its ability to provide rich information about the imaged regions, which aid in the visualization and quantification of various tissues and organs. However, manual segmentation of 3D images is challenging and time-consuming, highlighting the importance of automated computer-aided segmentation models. or 3D medical image segmentation, CNN models often employ a 3D kernel to extract spatial features or utilize GNN or Transformer. Some of SOTA models in this field include 3D U-Net [40], V-Net [39], nnU-Net [84], HighRes3dNet [85], 3D-Res-Unet [86], DenseVNet [87], UNETR [88], SegResNet [89], Point-Unet [90], and others.
- Structured 4D images, which are comprised of height, width, depth, and temporal dimensions, are commonly referred to as dynamic volumes and represent moving data in real-time. In the medical field, 4D images are often used to measure various parameters such as heart rate [91], lung breathing [92], blood flow rate, fetal movement, and more. In the field of medicine, 4D imaging techniques include dynamic volume CT, 4D CT, MRI, and ultrasound. The segmentation of 4D medical images has great potential in uncovering disease progression and monitoring disease trajectory [93]. However, traditional image segmentation can be complex and costly without the aid of AI models. Commonly used models in 4D image segmentation include "balloon" models and deformable (time-varying) models, such as, LSTM, FCSLSTM [93], and XCAT [94].

Based on the characteristics of each medical imaging modality, as well as the statistics from the MICCAI 2023 and 2022 challenges, and the latest papers on paperswithcode, it appears that 3D medical imaging techniques, such as MRI and CT, are currently the most popular. The advantages of 3D medical images include the utilization of spatial relationships, the ability to obtain more information from image regions than can be obtained from 2D images, and a lower cost for processing and acquisition compared to 4D images.

3) Medical image dataset: Data is a crucial aspect of learning, and medical image segmentation datasets often contain sensitive information, making them highly privacy-sensitive. However, in order to evaluate the performance of image segmentation methods, the datasets must be made publicly available. To accomplish this, organizers gather and anonymize medical imaging datasets, and host various challenges to advance the field of medical imaging. There are several medical databases available for free use in research, including:

- World Health Organization (WHO): WHO is a United Nations agency that connects nations, partners, and people to promote health and ensure the safety of the world. Established in 1948, WHO serves the most vulnerable populations so that everyone, everywhere, can attain the best possible health. The World Health Data Hub of the WHO is a comprehensive digital platform for global health data that provides efficient solutions for collecting, storing, analyzing, and sharing timely, reliable, and actionable data. The WHO is responsible for managing and preserving a vast array of data collections related to global health and well-being, as mandated by its Member States.

- Medical ImageNet is a large-scale resource for machine learning based on medical images. The Stanford Center for Artificial Intelligence in Medicine and Imaging (AIMI Center) was established in 2018 with the primary objective of using AI to address clinically significant medical issues. The Stanford Medical ImageNet is a petabyte-scale searchable repository of annotated and de-identified clinical images (radiology and pathology) that are linked to genomic data and electronic medical records, providing a platform for the speedy development of computer vision systems.

- Kaggle: Kaggle offers a customizable Jupyter Notebook environment with no setup required. It provides free access to GPUs and a vast repository of community-submitted data and code. Within Kaggle, you will find all the necessary code and data to complete your data science projects. With over 50,000 public datasets and 400,000 public notebooks available, you can quickly complete any analysis. Currently, Kaggle has 930 medical datasets, including 210 medical image datasets.

- Paperswithcode: The mission of Papers with Code is to create a free and open resource for machine learning research, including papers, code, datasets, methods, and benchmarks. It lists 2029 results for medical image datasets. Simpson, Amber L., et al. released ten datasets for medical image segmentation for various tasks under the Medical Segmentation Decathlon.

- mridata.org is an open platform for researchers to share raw magnetic resonance imaging (MRI) datasets. The website was created through a collaboration between Professor Michael Lustig's group at UC Berkeley and Professor Shreyas Vasanawala's group at Stanford's Lucile Packard Children's Hospital. The datasets available on the website can be used for a variety of purposes.

- Medical image datasets for segmentation are collected by researchers from various sources, including scientific journals and conferences, and collaborations between health organizations and partnerships. These datasets can be found by searching Google Scholar and Google using keywords such as 'medical segmentation', 'medical image datasets for segmentation', etc.

4) Taxonomy of data-driven learning paradigms: Data-driven learning paradigms can be classified into four categories: supervised learning, unsupervised learning, semi-supervised learning, and weakly supervised learning.

- Supervised learning: The first type of machine learning that humans introduced was supervised learning. In this method, the learning model is trained by providing labeled data, where both the input data and the corresponding output labels are given. The model then uses this information to make predictions on new, unseen data based on what it has learned from the training data. Learning from a forest to predict a tree is an example of supervised learning. The first model to utilize this method was the perceptron. Most ML algorithms employ supervised learning. In the medical field, supervised learning is commonly used to estimate risk. With risk modeling, the computer not only replicates a doctor's expertise, but it can also uncover novel relationships that might not be noticeable to humans [95].

- Unsupervised learning: However, labeling data is often difficult, expensive, and time-consuming, requiring the assistance of specialists. Furthermore, as network architecture advances, models can learn from unlabeled data. This is known as unsupervised learning. The Deep Belief Network (DBN) was proposed by Hinton in 2006, which uses this unsupervised mechanism. Unsupervised learning is based on data structures for performing clustering and data classification. However, since the data is not labeled, the learning challenge will increase. The usefulness of the patterns identified through unsupervised learning needs to be assessed either by human inspection or through additional supervised learning tasks [96]. Studies using unsupervised learning for Medical Image Segmentation (MIS) include the following: breast cancer segmentation on MRI [97]; COVID segmentation on CT lung tissue [98]; brain segmentation on 3D MRI [99]; multi-modality segmentation, including 2D hand x-ray, 3D abdominal magnetic resonance (MR) image, and 3D cardiovascular MR images [100], as well as cardiac substructure segmentation and abdominal multi-organ segmentation between MRI and CT images [101], etc.

- Semi-supervised learning combines supervised and unsupervised learning paradigms to solve the problem of training an accurate classifier with less human effort and time. This framework leverages both a limited amount of labeled data and a large amount of unlabeled (undiagnosed) data to achieve this goal [102]. Studies using semi-supervised learning in the field of MIS can be found in the range of studies [103] to [105].

- Weakly supervised learning paradigms are used when raw data is not fully processed or processed inaccurately. The goal of weakly supervised learning is similar to that of supervised learning, but instead of using a carefully labeled and processed training set, weak supervision is provided through one or more weakly annotated examples. These examples may come from community sources, be the output of rule heuristics, the results of remote monitoring, or the output of other classifiers [106]. Numerous studies have employed weakly supervised learning for the segmentation of medical images, as demonstrated in researches [107] to [109].

One thing to remember about medical imaging data is that the amount of annotated (labeled) data is very small. TABLE VI. TABLE VI. compares the four paradigms discussed above in terms of labeling costs.

TABLE VI. COMPARISON OF LABELING COSTS OF DATA-DRIVEN LEARNING PARADIGMS

Methods	Data	Labeling Cost
Supervised Learning	Labeled data	High
Unsupervised Learning	Unlabeled data	No
Semi-supervised Learning	Labeled data + unlabeled data	Medium
Weakly supervised Learning	Labeled data (small) + noise data (incomplete and inaccurate labels) + unlabeled data	Low

The accuracy of data-driven learning models is directly proportional to the amount of annotated data and inversely proportional to the cost of labeling. To improve the accuracy of learning models while reducing the cost of labeling, two solutions that have been proposed are transfer learning and active learning.

Traditional machine learning methods assume that the training and test data come from the same domain, with the same input feature space and data distribution characteristics. However, this assumption is not always true in real-world machine learning scenario. In some cases, collecting and annotating training data can be expensive or difficult. Transfer learning addresses this issue by training models on labeled data from domains where data collection and annotation are easier. The knowledge gained from the training data is then transferred to the test data domain [110]. Many models today are pre-trained on the ImageNet dataset [111]. Transfer learning has been applied in the field of medical image analysis in works such as [112] and [113].

While the ImageNet dataset is large and well-labeled, medical image data has distinctive characteristics that set it apart from natural images in datasets like ImageNet. Active learning is a method of selecting a subset of data from a larger dataset, such as a data lake [114] for annotation. The goal of this method is to increase the amount of annotated data, reduce data annotation costs, and improve model performance. This paradigm has been used in papers ranging from [115] to [118]

C. Loss Functions

1) Common loss functions:

1) Common loss functions: The image segmentation labels each pixel with the corresponding class. Therefore, it is necessary to use a mechanism to calculate the loss weight for each pixel. The most commonly used loss functions in segmentation are cross entropy and its variants.

- Cross-Entropy loss is a fundamental function in medical image segmentation. It is derived from the Kullback-Leibler (KL) divergence, which measures the difference between two probability distributions, such as those provided by the training set. The minimum KL divergence is equivalent to the minimum Cross-Entropy. The Cross-Entropy is defined as follows:

$$L_{CE} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N} gt_i^c logms_i^s \qquad (1)$$

where gt_i^c is a binary indicator representing whether class label c is the correct classification for pixel i, and ms_i^s is the corresponding predicted probability.

- A variant of the Cross-Entropy loss is the Weighted Cross-Entropy loss (WCE). This loss function takes into account class imbalance by assigning weights to different classes. Another emerging variant of the Cross-Entropy loss is the Focal Loss, which adjusts the weights of well-classified training samples to reduce their impact.

$$L_{WCE} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N} w_c gt_i^c logms_i^s \qquad (2)$$

where w_c is the class weight assigned to penalize majority classes, w_c is typically set inversely proportional to the frequency of each class in the training set. In the experiments to follow, we will set the class weight w_c to be the reciprocal of class frequency in the training set.

Aside from cross-entropy, other standard loss functions used in image segmentation are the **Dice loss**, and the **Intersection over union (IoU) loss** - which is derived from the Jaccard index and measures the ratio of sample intersection to its union. Dice loss and IoU loss are often used to improve the corresponding evaluation metrics.

$$L_{Dice} = 1 - \frac{2\sum_{c=1}^{C}\sum_{i=1}^{N} gt_i^c ms_i^s}{\sum_{c=1}^{C}\sum_{i=1}^{N} gt_i^c + \sum_{c=1}^{C}\sum_{i=1}^{N} ms_i^c} \qquad (3)$$

$$L_{IoU} = 1 - \frac{\sum_{c=1}^{C}\sum_{i=1}^{N} gt_i^c ms_i^s}{\sum_{c=1}^{C}\sum_{i=1}^{N}(gt_i^c + ms_i^s - gt_i^c ms_i^s)} \qquad (4)$$

2) Hybrid loss functions: Jun Ma et al. presented an overview of loss functions in MIS sorted according to the following classification system: distribution-based, region-based, boundary-based, and association-based. At the same time, the authors also found the relationship (connection) between different loss functions (see Fig. 6), as well as the specific use case of loss functions in different applications of MIS [119]. These loss functions are installed on GitHub [4]

Loss function relationships: As shown in Fig. 6 there are strong connections between loss functions.

[4] https://github.com/JunMa11/SegLoss.

- Most distribution-based and region-based loss functions are variants of Cross entropy and Dice loss.
- Although boundary-based losses are designed to minimize the distance between two boundaries, they share some similarities with Dice loss because both are calculated using region-based methods.
- A compound loss is a combination of multiple loss functions.

Recommendations for selecting loss functions: It is impossible to determine which loss function is the best. The data balance can be used to select the appropriate loss function.

- Mild imbalance issues are well handled by Dice loss or General Dice loss (GD)
- For highly imbalanced segmentation tasks, the compound loss functions are more commonly used.

The Combo Loss function proposed by Saeid Asgari Taghanaki et al. [120] aims to improve multi-organ segmentation performance in cases where the input and output are imbalanced. This loss function has been shown to achieve higher Dice scores and reduce false negatives and false positives, and can be applied to 3D U-Net, 3D V-Net, and 3D Seg-Net architectures.

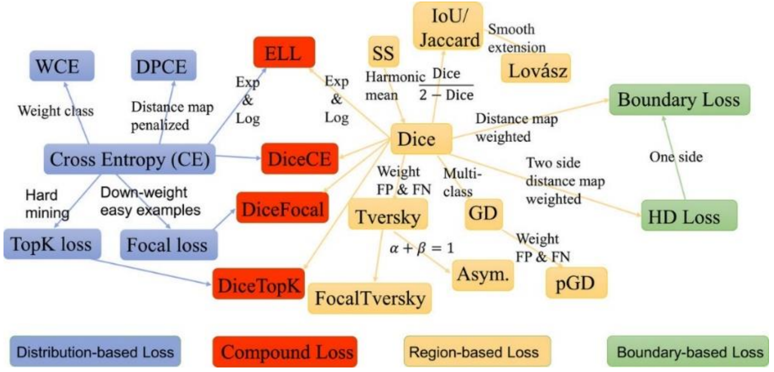


Fig. 6. An overview of 20 loss functions for MIS and their relationships [119].

D. Evaluation Metrics

Segmentation performance evaluation involves comparing the similarity between a manually generated and a DNN-generated segmentation. Many metrics are used for evaluation, but some of the most commonly used ones in medical imaging are presented below.

In MIS, the regions of interest (ROI) are often quite small compared to the overall image. This results in an imbalanced distribution of pixels between different classes. To address this issue, two common metrics used in MIS are the Dice Similarity Coefficient (DSC), also known as the F1 score, and the Intersection over Union (IoU), also known as the Jaccard index.

Explanation of some acronyms: GT: Ground Truth; MS: Machine Segmentation; TP: True positive; TN: True negative; FP: False positive; FN: False negative; these terms are illustrated in Fig. 7, which has been modified from source [121].

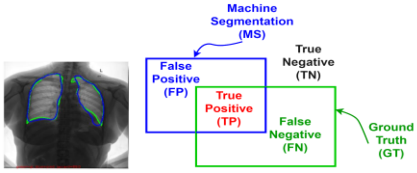


Fig. 7. Illustrate GT, MS, TP, TN, FP, and FN modified from the source [121].

1) Jaccard index (JAC - IoU) evaluates the overlap between GT and MS regions.

$$JAC = IoU = \frac{▨}{▨} = \frac{|GT \cap MS|}{|GT \cup MS|} = \frac{TP}{TP + FP + FN} \quad (5)$$

2) Dice similarity coefficient (DSC - F1 score) measures the similarity between GT and MS regions.

$$DSC = F1\ Score = \frac{2 * ▨}{□ + □} = \frac{2|GT \cap MS|}{|GT| + |MS|} \quad (6)$$

$$= \frac{2TP}{2TP + FP + FN} = \frac{2JAC}{1 + JAC}$$

The following are other metrics that are less commonly used in the evaluation of medical image segmentation due to their sensitivity to the size of the segment, meaning they penalize errors in small segments more heavily than in larger segments.

3) The Hausdorff distance (HD) between two GT and MS regions is defined as "the maximum of all minimum distances" (see Fig. 8) [122].

$$HD\ (GT, MS) = \ max\{\{d(gt, ms)\}\} \quad (7)$$

where gt and ms represent the pixels of regions GT and MS respectively, and d (gt, ms) is any metric between these pixels; for the sake of simplicity, take d (gt, ms) is taken as the Euclidean distance between gt and ms.

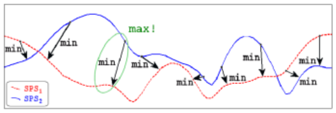


Fig. 8. Illustrate the Hausdorff distance [122].

4) Other evaluation metrics

- Sensitivity / Recall / True Positive Rate (TPR):

$$Sensitivity = Recall = TPR = \frac{TP}{TP+FN} \quad (8)$$

- Specificity or True Negative Rate (TNR):

$$Specificity = TNR = \frac{TN}{TN+FP} \quad (9)$$

- Accuracy (ACC):

$$Acc = \frac{TP+TN}{TP+FP+TN+FP} \quad (10)$$

- False Positive Rate (FPR)

$$FPR = \frac{FP}{FP+TN} = 1 - TNR \quad (11)$$

- False Negative Rate (FNR)

$$FNR = \frac{FN}{FN+TP} = 1 - TPR \quad (12)$$

- Precision or Positive Predictive Value (PPV):

$$Precision = PPV = \frac{TP}{TP+FP} \quad (13)$$

III. Explainable Artificial Intelligence in Medical Image Segmentation (XAI in MIS)

Recently, the use of AI in the medical field has led to impressive results, with machine learning models achieving over 99% accuracy. However, the practical application of these models remains limited due to their "black box" nature. XAI learns how models make decisions, investigates the inner workings of its layers, and provides visualizations of neural networks. Interpretability, which supports the reasoning behind a model's output, is crucial, particularly in precision medicine where experts require more in-depth information from a model beyond just a binary prediction to make an accurate diagnosis. In general, people are cautious about using techniques that are not transparent and can't be easily understood, due to the increasing emphasis on ethical AI, particularly in fields that have a direct impact on human lives [123]. To build trust among physicians, regulators, and patients, medical diagnostic systems must be transparent, easily understandable, and capable of providing explanations [123]. Moreover, by using XAI, model designers can uncover weaknesses in existing architecture and debug and fine-tune models to improve their performance.

If an image-based diagnostic system is enhanced with XAI, it will reach a level of Wisdom, which is higher than mere Intelligence. This is because, in addition to having high IQ, it also has high EQ or a sense of responsibility.

XAI can be divided into two categories: interpretability and explainability.

- "Interpretability" focuses on the underlying processes and events. It answers the "how" question by showing how the decision was made (based on the scoring criteria), but it does not explain "why" the criteria used are reasonable. The term "interpretability" addresses the "quantitative" aspects of the decision-making process.

- "Explainability" focuses on the inherent characteristics of events. It addresses the "why" question by explaining the reasoning behind the decision, rather than just the "how". The term "explainability" refers to the "qualitative" aspects of the decision-making process. Some commonly used methods for explainability include CAM [124], Grad-CAM [125], Grad-CAM++ [126], LIME [127], and SHAP [128]:
  - Bolei Zhou and colleagues introduced the Class Activation Maps (CAM) approach [124], which utilizes a global average pooling (GAP) layer at the end of the neural network. CAM provides an interpretation of the model and reveals the areas of the image that the network focuses on to make decisions. By producing heatmaps, CAM shows which regions of the image are most important for decision-making.
  - Building on the CAM approach, various variants have been developed for the XAI process, including the widely used Grad-CAM [125] and Grad-CAM++ [126].
  - Local Interpretable Model-agnostic Explanations (LIME) [127] were introduced by Marco Tulio Ribeiro and others as a method for creating an interpretable model that accurately reflects the classifier in a locally interpretable representation.
  - SHAP (SHapley Additive exPlanations) [128], developed by S. Lundberg and S.-I. Lee, is a method that assigns a significant value to each feature for a specific prediction. This approach is viewed as a comprehensive framework for interpreting predictions.

There are some XAI models that can only be interpretable, explainable or both.

1) Consider the linear regression model for disease diagnosis.

- It is considered "interpretable" because once the coefficients of the linear model are calculated, the new input can be used to determine how the predicted output was obtained. This process is clearly quantified.

- However, it is "unexplainable" because it does not provide an explanation for why there is a linear relationship between the independent and dependent variables, which is not clearly quantified.

- Both "interpretability" and "explainability" can be achieved if the statistical process includes a step for testing the linear hypothesis. This provides a way to determine the presence of a linear relationship and quantify the relationship between the independent and dependent variables, making the results both interpretable and explainable.

2) Example of a model that can only be explained: The lung infection segmentation model from X-ray images was trained using a Deep Convolutional Neural Network (DCNN) combined with Class Activation Maps (CAM) on data from two different scanners at two hospitals. It is possible to understand which areas of the lungs contribute to a positive outcome ("why"), but the reason why the sample test with the trained model at hospital 1 is positive but with the trained model at hospital 2 is negative is unclear ("how" to calculate it is not understood).

IV. Early Prediction with MIS

This section presents the role and typical applications of MIS in the field of early prediction.

Tumors often start as small, hard-to-detect nodules, which can lead to a high mortality rate among patients. However, if detected early and treated promptly, it can increase the patient's chances of survival and decrease the cost of treatment. This is the motivation behind researchers performing early prediction problems. The segmentation results can be used to anticipate disease progression and offer appropriate treatment. Early prediction has a critical role in reducing patient risk, which holds great humanitarian significance. Furthermore, it helps to elevate the level of the system from intelligence to wisdom.

A. Some Typical Applications

The Fuzzy C-means (FCM) intelligent segmentation algorithm [129] was developed for early detection of enlarged hematoma in patients with intracerebral hemorrhage (ICH) on CT images. The processing of cranial CT images using the FCM algorithm has high clinical value in predicting early hematoma in ICH patients.

Lung cancer is usually detected early when lesions appear in the bronchial epithelium of the airway wall. Autofluorescence bronchoscopy (AFB) [52] has been shown in recent studies to be particularly effective in detecting lesions, making it a potentially crucial approach for airway evaluation. Bronchoscopy is a commonly used method for detecting early-stage lung cancer. ESFPNet is a method for accurately segmenting and identifying bronchial lesions in AFB video streams.

There are also some other early prediction models based on segmentation results such as brain-related disease [130] - [132], evaluation of bone tumors [133], lung disease [134] - [136], breast cancer [137] - [138], tumor metastasis of ovarian cancer patients [139], stroke, ischemic coma [140] - [141], acute pancreatitis [142], and cancer radiation [143]- [144].

In addition, early segmentation is also applicable to cases with noisy and incomplete annotations [145].

V. Challenges and Solutions

A. Challenges with Dataset

1) Shortage of large-scale, annotated, and standardized datasets: High-quality and large amounts of data are crucial for deep neural network (DNN) models. However, obtaining data, especially sensitive information like medical images, can be difficult through crawling methods. Additionally, manual annotations on medical images can be time-consuming, costly, and require specialized knowledge.

Solutions to address the shortage of datasets include: (a) utilizing unsupervised learning techniques, such as semi-supervised learning, weak supervision, active learning, and transfer learning (as discussed in Section IV4); (b) utilizing open-source datasets, including those mentioned in previous sections, as well as others listed in "10 Open Repositories for the Medical Image Processing Community" by vin bigdata [146]; (c) creating simulated data using the Synthetic Minority Oversampling (SMOTE) technique, which generates new data points based on the closest data points of the minority class; (d) implementing data augmentation [51]; (e) and Deploying Pre-Trained DNN models [147].

2) Class Imbalance in Datasets: Medical datasets are often highly imbalanced with regards to class distribution, with some diseases having significantly more negative samples than positive samples, sometimes up to a 98% to 2% ratio. This level of imbalance can result in biased outcomes, and the prediction model may not be accurate for the minority class as it is frequently biased towards the majority class.

Solutions to class imbalance in medical datasets include adjusting the evaluation metric, under-sampling, over-sampling, collecting more data, incorporating model penalties, and using cross-validation.

In addition to class imbalance, medical image datasets can also face issues of sparse annotations and intensity inhomogeneities, which can affect model performance. There are specific solutions for each of these cases presented in the papers [148] and [149].

B. Challenges with DNN

1) Training time: Training a deep neural network (DNN) requires significant time and effort to extract the necessary features and enhance performance.

Solutions to improve the efficiency of DNN training include: (a) implementing batch normalization; (b) using pooling layers to reduce image size and the number of parameters, and employing the early stopping technique to minimize unnecessary training time and avoid overfitting.

2) Overfitting: Deep learning algorithms are susceptible to overfitting, where a model becomes too tailored to the training data and memorizes both meaningful patterns and random noise and fluctuations. This can result in poor generalization and low performance on unseen test data.

To prevent overfitting, various techniques can be utilized, as outlined in [150]. These include training with more data, implementing data augmentation, adding noise to input data, performing feature selection, utilizing cross-validation, simplifying data, applying regularization, using ensembling methods, employing early stopping, and incorporating dropout layers.

3) Gradient vanishing: Gradients tend to decrease in magnitude as back propagation be activated in a deep neural network (DNN). This means that the updates performed by Gradient Descent have limited impact on the weights of these layers, making convergence difficult and potentially leading to poor performance of the DNN. This phenomenon is referred to as the Vanishing Gradients problem.

To address the Vanishing Gradients problem, there are two main approaches: (a) Preprocessing, appropriate activation function selection, and proper weight initialization. (b) Using residual blocks and skip connections along the encoder-decoder path, as described in [151].

4) Computational complexity: Deep learning algorithms often involve complex calculations, and the training process is typically performed on high-performance hardware such as GPUs or supercomputers.

To address this, it is important to construct compact and portable models with a reduced number of parameters while still maintaining high performance for the intended task [152]. This can be achieved through various techniques such as regularization, model pruning, and efficient architectures

5) The "black box" nature of AI models: The "black box" nature of artificial intelligence raises important ethical, transparent, and explainability concerns in clinical medical applications. Despite numerous efforts in the field of XAI, this problem remains a significant challenge for researchers to address.

To address the challenges of explainability in AI, there are several solutions available, beyond those mentioned in Section III, such as CAM, LIME, and SHAP. Other measures include SAU-Net [153], saliency map, and others.

Building on the progress made by the XAI community, Seibold et al. [154] have proposed an indirect segmentation method that leverages Layer Relevancy Propagation (LRP) to extract binary segmentation maps at the pixel level. This approach demonstrates comparable results to UNet while only using image-level labeled data.

6) Early segmentation problems: This approach represents a new and promising opportunity for application in clinical settings. It has the potential to drive advancements in both medicine and AI, and ultimately contribute to better protection of human health. This is a significant breakthrough that holds great promise for the future.

Solutions:

In the early stages of tumor development, it can be difficult to detect small tumors due to their size. To address this challenge, it is important to build highly sensitive models that are capable of accurately segmenting small regions of interest, such as the TransUNet+ model described in [50].

To predict the progression of a disease in patients, one approach is to conduct a follow-up study that combines previous and current results to forecast future segmentation outcomes. In practical experience, malignant tumors often grow abnormally and out of control, making early and accurate detection and segmentation critical for reducing the risk of patient mortality. To achieve this goal, it is crucial to develop early and precise auto-segmentation technology. The use of past observations to predict future events is a key method for making informed decisions, as described in [155].

VI. Conclusions and Future Work

Over the past decade, AI-assisted automated medical image segmentation has gained significant attention from the computer vision community. This study provides a comprehensive overview of the state-of-the-art solutions in terms of network architecture, data, loss functions, and performance metrics. With a focus on network architecture, this research categorizes the solutions into three levels of intelligent vision systems. The study also considers the "intelligence" level of the DIKIW hierarchy.

The paper also highlights two issues that are currently gaining significant attention, which are Explainable AI (XAI) and Medical Image Segmentation (MIS)-based early prediction. Both of these topics are considered hot research areas due to their practical and humanistic values. By addressing XAI and early prediction, the level of the DIKIW hierarchy can be raised from "intelligence" to "wisdom". Despite some progress in these areas, there are still numerous challenges that remain for researchers to tackle.

The next step in the research direction of this paper is to develop a trusted XAI-based early diagnosis support system that can be used in hospitals by both patients and doctors.

The aim of the paper is to advance the use of AI in disease diagnosis to a level of wisdom and, as a result, improve the well-being of society.

Acknowledgment

The project is supported by the Department of Computer Vision and Cognitive Cybernetics, Faculty of Information Technology, VNUHCM-University of Science.

References

[1] A. Liew, “DIKIW: Data, information, knowledge, Intelligence, wisdom and their interrelationships,” 2013.

[2] *A. M. Tahir et al., “COVID-19 infection localization and severity grading from chest X-ray images,” Comput. Biol. Med., vol. 139, no. 105002, p. 105002, 2021.*

[3] K. Fukushima and S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” *Pattern Recognit.*, vol. 15, no. 6, pp. 455–469, 1982.

[4] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. ArXiv [Cs.CV]. https://doi.org/10.48550/ARXIV.1409.1556

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” arXiv [cs.CV], 2015.

[6] Huang, G., Liu, Z., Maaten, L. van der, & Weinberger, K. Q. (2017). Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. ArXiv [Cs.CV]. https://doi.org/10.48550/ARXIV.1706.05587

[8] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv [Cs.CV]. https://doi.org/10.48550/ARXIV.1704.04861

[9] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional Neural Networks. ArXiv [Cs.LG]. https://doi.org/10.48550/ARXIV.1905.11946

[10] Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. Neural Computation, 1(2), 263–269. https://doi.org/10.1162/neco.1989.1.2.263

[11] T. Lei, R. Wang, Y. Wan, X. Du, H. Meng, και A. K. Nandi, ‘Medical Image Segmentation Using Deep Learning: A Survey’, 2020.

[12] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

[13] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, “Graph-based deep learning for medical diagnosis and analysis: Past, present and future,” *Sensors (Basel)*, vol. 21, no. 14, p. 4758, 2021.

[14] I. J. Goodfellow et al., “Generative Adversarial Networks,” arXiv [stat.ML], 2014.

[15] S. Xun *et al.*, “Generative adversarial networks in medical image segmentation: A review,” *Comput. Biol. Med.*, vol. 140, no. 105063, p. 105063, 2021.

[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In arXiv [cs.CL]. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[17] J. Chen *et al.*, “TransUNet: Transformers make strong encoders for medical image segmentation,” *arXiv [cs.CV]*, 2021.

[18] F. Shamshad *et al.*, “Transformers in medical imaging: A survey,” *arXiv [eess.IV]*, 2022.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” arXiv [cs.CV], 2015.

[20] X. Ren, X. Guo, and S. Xu, “A medical image segmentation method combining inception module with U-net model,” Research Square, 2022.

[21] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (2016). p. 1–13.

[22] Wang κ.ά., ‘Stacked dilated convolutions and asymmetric architecture for U-Net-based medical image segmentation’, Computers in Biology and Medicine, τ. 148, σ. 105891, 2022.

[23] Liu, W., Lei, H., Xie, H., Zhao, B., Yue, G., & Lei, B. (2020). Multi-level light U-net and atrous spatial pyramid pooling for optic disc segmentation on fundus image. In Ophthalmic Medical Image Analysis (pp. 104–113). Springer International Publishing.

[24] Qayyum, A., Ahmad, I., Mumtaz, W., Alassafi, M. O., Alghamdi, R., & Mazher, M. (2020). Automatic segmentation using a hybrid dense network integrated with an 3D-atrous spatial pyramid pooling module for computed tomography (CT) imaging. IEEE Access: Practical Innovations, Open Solutions, 8, 169794–169803. https://doi.org/10.1109/access.2020.3024277

[25] Wang, J., Lv, P., Wang, H., & Shi, C. (2021). SAR-U-Net: Squeeze-and-excitation block and atrous spatial pyramid pooling based residual U-Net for automatic liver segmentation in Computed Tomography. Computer Methods and Programs in Biomedicine, 208(106268), 106268. https://doi.org/10.1016/j.cmpb.2021.106268

[26] X. Qiu, “U-Net-ASPP: U-Net based on atrous spatial pyramid pooling model for medical image segmentation in COVID-19,” Tamkang j. sci. eng., vol. 25, no. 6, pp. 1167–1176, 2022.

[27] M. M. Stofa, M. A. Zulkifley, M. A. A. M. Zainuri, and A. A. Ibrahim, “U-net with atrous spatial pyramid pooling for skin lesion segmentation,” in Lecture Notes in Electrical Engineering, Singapore: Springer Singapore, 2022, pp. 1025–1033.

[28] Z. Yang, L. Chen, T. Fu, Z. Yin, and F. Yang, “Spine Image Segmentation Based on U-Net and Atrous spatial pyramid pooling,” J. Phys. Conf. Ser., vol. 2209, no. 1, p. 012020, 2022.

[29] K. Xu et al., “Show, attend and tell: Neural image caption generation with visual attention,” arXiv [cs.LG], pp. 2048–2057, 07--09 Jul 2015.

[30] R. Gu et al., “CA-Net: Comprehensive attention convolutional Neural Networks for explainable medical image segmentation,” IEEE Trans. Med. Imaging, vol. 40, no. 2, pp. 699–711, 2021.

[31] J. Hu, H. Wang, J. Wang, Y. Wang, F. He, and J. Zhang, “SA-Net: A scale-attention network for medical image segmentation,” PLoS One, vol. 16, no. 4, p. e0247388, 2021.

[32] D. Nie, Y. Gao, L. Wang, and D. Shen, “ASDNet: Attention based semi-supervised deep networks for medical image segmentation,” in Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Cham: Springer International Publishing, 2018, pp. 370–378.

[33] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.

[34] A. G. Roy, N. Navab, and C. Wachinger, “Recalibrating fully convolutional networks with spatial and channel ‘squeeze and excitation’ blocks,” IEEE Trans. Med. Imaging, vol. 38, no. 2, pp. 540–549, 2019.

[35] J. Chen et al., “An efficient memristor-based circuit implementation of squeeze-and-excitation fully convolutional neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 4, pp. 1779–1790, 2022.

[36] A. Iantsen, D. Visvikis, and M. Hatt, “Squeeze-and-excitation normalization for automated delineation of head and neck primary tumors in combined PET and CT images,” in Head and Neck Tumor Segmentation, Cham: Springer International Publishing, 2021, pp. 37–43.

[37] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual Dense Network for Image Super-Resolution,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2472–2481.

[38] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In arXiv [cs.CV]. http://arxiv.org/abs/1505.04597

[39] Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth International Conference on 3D Vision (3DV), 565–571.

[40] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-net: Learning dense volumetric segmentation from sparse annotation. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016 (pp. 424–432). Springer International Publishing.

[41] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV).

[42] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. ArXiv [Cs.CV]. https://doi.org/10.48550/ARXIV.1807.10165

[43] Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.-W., & Wu, J. (2020). UNet 3+: A full-scale connected UNet for medical image segmentation. In arXiv [eess.IV]. http://arxiv.org/abs/2004.08790

[44] Jafari, M., Auer, D., Francis, S., Garibaldi, J., & Chen, X. (2020). DRU-Net: An Efficient Deep Convolutional Neural Network for Medical Image Segmentation. 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), 1144–1148.

[45] Jha, D., Riegler, M. A., Johansen, D., Halvorsen, P., & Johansen, H. D. (2020). DoubleU-Net: A deep convolutional neural network for medical image segmentation. ArXiv [Eess.IV]. https://doi.org/10.48550/ARXIV.2006.04868

[46] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021). TransUNet: Transformers make strong encoders for medical image segmentation. ArXiv [Cs.CV]. https://doi.org/10.48550/ARXIV.2102.04306

[47] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision Transformer using shifted windows. ArXiv [Cs.CV]. https://doi.org/10.48550/ARXIV.2103.14030

[48] Shaker, A., Maaz, M., Rasheed, H., Khan, S., Yang, M.-H., & Khan, F. S. (2022). UNETR++: Delving into efficient and accurate 3D medical image segmentation. In arXiv [cs.CV]. http://arxiv.org/abs/2212.04497

[49] Zhao, Z., Zhou, F., Xu, K., Zeng, Z., Guan, C., & Zhou, S. K. (2022). LE-UDA: Label-efficient unsupervised domain adaptation for medical image segmentation. In arXiv [eess.IV]. http://arxiv.org/abs/2212.02078

[50] Y. Liu, H. Wang, Z. Chen, K. Huangliang, and H. Zhang, "TransUNet ＋: Redesigning the skip connection to enhance features in medical image segmentation," Knowl. Based Syst., vol. 256, no. 109859, p. 109859, 2022.

[51] Zhou, L. (2022). Spatially exclusive pasting: A general data augmentation for the polyp segmentation. In arXiv [eess.IV]. http://arxiv.org/abs/2211.08284

[52] Chang, Q., Ahmad, D., Toth, J., Bascom, R., & Higgins, W. E. (2022). ESFPNet: efficient deep learning architecture for real-time lesion segmentation in autofluorescence bronchoscopic video. In arXiv [eess.IV]. http://arxiv.org/abs/2207.07759

[53] Isensee, F., Jäger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2019). Automated design of deep learning methods for biomedical image segmentation. In arXiv [cs.CV]. http://arxiv.org/abs/1904.08128

[54] Li, S., Sui, X., Luo, X., Xu, X., Liu, Y., & Goh, R. (2021). Medical Image Segmentation using Squeeze-and-Expansion Transformers. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence.

[55] Isensee, F., Ulrich, C., Wald, T., & Maier-Hein, K. H. (2022). Extending nnU-Net is all you need. In arXiv [eess.IV]. http://arxiv.org/abs/2208.10791

[56] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer statistics, 2022," CA Cancer J. Clin., vol. 72, no. 1, pp. 7–33, 2022.

[57] Jun, SOTA-MedSeg: SOTA medical image segmentation methods based on various challenges. .

[58] Wikipedia contributors, "Medical imaging," Wikipedia, The Free Encyclopedia, 22-Aug-2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Medical_imaging&oldid=1105973862

[59] KASBAN, Hany; EL-BENDARY, M. A. M.; SALAMA, D. H. A comparative study of medical imaging techniques. International Journal of Information Science and Intelligent System, 2015, 4.2: 37-58.

[60] International Journal of Radiology Sciences, "A comparative study of medical imaging modalities," Radiologyjournals.com. [Online]. Available: https://www.radiologyjournals.com/archives/2021.v3.i1.11. [Accessed: 03-Nov-2022].

[61] F. Olubusola Isinkaye, A. Gabriel Aluko, and O. Ayodele Jongbo, "Segmentation of medical X-ray bone image using different image processing techniques," Int. J. Image Graph. Signal Process., vol. 13, no. 5, pp. 27–40, 2021.

[62] H. Azimi et al., "Improving classification model performance on chest X-rays through lung segmentation," arXiv [eess.IV], 2022.

[63] H. Zhu, Z. Cao, L. Lian, G. Ye, H. Gao, and J. Wu, "CariesNet: a deep learning approach for segmentation of multi-stage caries lesion from oral panoramic X-ray image," Neural Comput. Appl., pp. 1–9, 2022.

[64] S. P. Primakov et al., "Automated detection and segmentation of non-small cell lung cancer computed tomography images," Nat. Commun., vol. 13, no. 1, p. 3423, 2022.

[65] P. Tang, P. Yang, D. Nie, X. Wu, J. Zhou, and Y. Wang, "Unified medical image segmentation by learning from uncertainty in an end-to-end manner," Knowl. Based Syst., vol. 241, no. 108215, p. 108215, 2022.

[66] F. Isensee et al., "nnU-Net: Self-adapting framework for U-Net-based medical image segmentation," arXiv [cs.CV], 2018.

[67] Y. Deng et al., "A deep learning-based approach to automatic proximal femur segmentation in quantitative CT images," Med. Biol. Eng. Comput., vol. 60, no. 5, pp. 1417–1429, 2022.

[68] L. B. da Cruz et al., "Kidney tumor segmentation from computed tomography images using DeepLabv3+ 2.5D model," Expert Syst. Appl., vol. 192, no. 116270, p. 116270, 2022.

[69] Z. Cui et al., "A fully automatic AI system for tooth and alveolar bone segmentation from cone-beam CT images," Nat. Commun., vol. 13, no. 1, p. 2096, 2022.

[70] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images," in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, Cham: Springer International Publishing, 2022, pp. 272–284.

[71] D. Karimi, H. Dou, and A. Gholipour, "Medical image segmentation using transformer networks," IEEE Access, vol. 10, pp. 29322–29332, 2022.

[72] L. Boone et al., "ROOD-MRI: Benchmarking the robustness of deep learning segmentation models to out-of-distribution and corrupted data in MRI," arXiv [eess.IV], 2022.

[73] Y. Wu et al., "Vessel-net: Retinal vessel segmentation under multi-path supervision," in Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 264–272.

[74] E. S. Uysal, M. Ş. Bilici, B. S. Zaza, M. Y. Özgenç, and O. Boyar, "Exploring the limits of data augmentation for retinal Vessel Segmentation," arXiv [eess.IV], 2021.

[75] C. You, Y. Zhou, R. Zhao, L. Staib, and J. S. Duncan, "SimCVD: Simple contrastive voxel-wise representation distillation for semi-supervised medical image segmentation," arXiv [cs.CV], 2021.

[76] Z. Xiao, Y. Su, Z. Deng, and W. Zhang, "Efficient combination of CNN and transformer for Dual-teacher uncertainty-guided semi-supervised medical image segmentation," Comput. Methods Programs Biomed., vol. 226, no. 107099, p. 107099, 2022.

[77] Q. Wang, Y. Du, H. Fan, and C. Ma, "Towards collaborative appearance and semantic adaptation for medical image segmentation," Neurocomputing, vol. 491, pp. 633–643, 2022.

[78] L. Ma and L. Liang, "Adaptive adversarial training to improve adversarial robustness of DNNs for medical image segmentation and detection," arXiv [eess.IV], 2022.

[79] B. Lv, F. Liu, F. Gou, and J. Wu, "Multi-scale tumor localization based on Priori guidance-based segmentation method for osteosarcoma MRI images," Mathematics, vol. 10, no. 12, p. 2099, 2022.

[80] G. Chen, Y. Dai, and J. Zhang, "C-Net: Cascaded convolutional neural network with global guidance and refinement residuals for breast ultrasound images segmentation," Comput. Methods Programs Biomed., vol. 225, no. 107086, p. 107086, 2022.

[81] G. Chen, J. Yin, Y. Dai, J. Zhang, X. Yin, and L. Cui, "A novel convolutional neural network for kidney ultrasound images segmentation," Comput. Methods Programs Biomed., vol. 218, no. 106712, p. 106712, 2022.

[82] J. Jiang, Y. Guo, Z. Bi, Z. Huang, G. Yu, and J. Wang, "Segmentation of prostate ultrasound images: the state of the art and the future directions of segmentation algorithms," Artif. Intell. Rev., vol. 56, no. 1, pp. 615–651, 2023.

[83] Q. Zhou, Q. Wang, Y. Bao, L. Kong, X. Jin, and W. Ou, "LAEDNet: A Lightweight Attention Encoder–Decoder Network for ultrasound medical image segmentation," Comput. Electr. Eng., vol. 99, no. 107777, p. 107777, 2022.

[84] F. Isensee, P. F. Jäger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Automated design of deep learning methods for biomedical image segmentation," arXiv [cs.CV], 2019.

[85] W. Li, G. Wang, L. Fidon, S. Ourselin, M. J. Cardoso, and T. Vercauteren, "On the compactness, efficiency, and representation of 3D convolutional networks: Brain parcellation as a pretext task," in Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 348–360

[86] S. Mathlouthi, A. G. Blaiech, M. Said, A. B. Abdallah, and M. Hedi Bedoui, "A novel deep learning model for knee cartilage 3D segmentation," in 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA), 2021, pp. 1–7.

[87] E. Gibson et al., "Automatic multi-organ segmentation on abdominal CT with dense V-networks," IEEE Trans. Med. Imaging, vol. 37, no. 8, pp. 1822–1834, 2018.

[88] Hatamizadeh, A., Tang, Y., Nath, V., Yang, D., Myronenko, A., Landman, B., Roth, H. R., & Xu, D. (2022). UNETR: Transformers for 3D Medical Image Segmentation. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 574–584.

[89] C. Hsu, C. Chang, T. W. Chen, H. Tsai, S. Ma, and W. Wang, "Brain tumor segmentation (BraTS) challenge short paper: Improving three-

dimensional brain tumor segmentation using SegResnet and hybrid boundary-dice loss," in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, Cham: Springer International Publishing, 2022, pp. 334–344.

[90] N.-V. Ho, T. Nguyen, G.-H. Diep, N. Le, and B.-S. Hua, "Point-unet: A context-aware point-based neural network for volumetric segmentation," in Medical Image Computing and Computer Assisted Intervention – MICCAI 2021, Cham: Springer International Publishing, 2021, pp. 644–655.

[91] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis," *Comput. Med. Imaging Graph.*, vol. 19, no. 1, pp. 69–83, 1995.

[92] M. C. Aznar, G. F. Persson, I. M. Kofoed, D. E. Nygaard, and S. S. Korreman, "Irregular breathing during 4DCT scanning of lung cancer patients: is the midventilation approach robust?," *Phys. Med.*, vol. 30, no. 1, pp. 69–75, 2014.

[93] Y. Gao, J. M. Phillips, Y. Zheng, R. Min, P. T. Fletcher, and G. Gerig, "Fully convolutional structured LSTM networks for joint 4D medical image segmentation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 1104–1108.

[94] W. P. Segars, B. M. W. Tsui, J. Cai, F.-F. Yin, G. S. K. Fung, and E. Samei, "Application of the 4-D XCAT phantoms in biomedical imaging and beyond," *IEEE Trans. Med. Imaging*, vol. 37, no. 3, pp. 680–692, 2018.

[95] Deo, R. C. (2015). Machine learning in medicine. Circulation, 132(20), 1920–1930. https://doi.org/10.1161/CIRCULATIONAHA.115.001593

[96] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, M. J. Lee, and H. Asadi, "eDoctor: machine learning and the future of medicine," J. Intern. Med., vol. 284, no. 6, pp. 603–619, 2018.

[97] C. Militello *et al.*, "On unsupervised methods for medical image segmentation: Investigating classic approaches in breast cancer DCE-MRI," *Appl. Sci. (Basel)*, vol. 12, no. 1, p. 162, 2021.

[98] F. Gholamiankhah, S. Mostafapour, N. A. Goushbolagh, S. Shojaerazavi, H. Arabi, and H. Zaidi, "A novel unsupervised COVID-19 lesion segmentation from CT images based-on the lung tissue detection," in *2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2021, pp. 1–3.

[99] X. Wang, C. Guo, and X. Zhou, "Robust segmentation of 3D brain MRI images in cross datasets by integrating supervised and unsupervised learning," in *2020 10th International Conference on Information Science and Technology (ICIST)*, 2020, pp. 194–201.

[100] I. Aganj, M. G. Harisinghani, R. Weissleder, and B. Fischl, "Unsupervised medical image segmentation based on the local center of mass," *Sci. Rep.*, vol. 8, no. 1, p. 13012, 2018.

[101] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng, "Unsupervised bidirectional cross-modality adaptation via deeply Synergistic Image and Feature Alignment for medical image segmentation," *IEEE Trans. Med. Imaging*, vol. 39, no. 7, pp. 2494–2505, 2020.

[102] A. Chebli, A. Djebbar, and H. F. Marouani, "Semi-supervised learning for medical application: A survey," in 2018 International Conference on Applied Smart Systems (ICASS), 2018, pp. 1–9.

[103] Y. Li, J. Chen, X. Xie, K. Ma, and Y. Zheng, "Self-loop uncertainty: A novel pseudo-label for semi-supervised medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, Cham: Springer International Publishing, 2020, pp. 614–623.

[104] C. You, Y. Zhou, R. Zhao, L. Staib, and J. S. Duncan, "SimCVD: Simple contrastive voxel-wise representation distillation for semi-supervised medical image segmentation," *IEEE Trans. Med. Imaging*, vol. 41, no. 9, pp. 2228–2237, 2022.

[105] Zhang, Y., & Zhang, J. (2021). Dual-task mutual learning for semi-supervised medical image segmentation. In *Pattern Recognition and Computer Vision* (pp. 548–559). Springer International Publishing.

[106] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," arXiv [cs.LG], 2020.

[107] Yang, H., Wu, G., Shen, D., & Liao, S. (2021). Automatic prostate cancer detection on multi-parametric MRI with hierarchical weakly supervised learning. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 316–319.

[108] Lyu, F., Ma, A. J., Yip, T. C.-F., Wong, G. L.-H., & Yuen, P. C. (2022). Weakly supervised liver tumor segmentation using Couinaud segment annotation. *IEEE Transactions on Medical Imaging*, *41*(5), 1138–1149. https://doi.org/10.1109/TMI.2021.3132905

[109] Belharbi, S., Rony, J., Dolz, J., Ayed, I. B., Mccaffrey, L., & Granger, E. (2022). Deep interpretable classification and weakly-supervised segmentation of histology images via max-min uncertainty. *IEEE Transactions on Medical Imaging*, *41*(3), 702–714. https://doi.org/10.1109/TMI.2021.3123461

[110] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," J. Big Data, vol. 3, no. 1, 2016.

[111] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[112] van Opbroek, A., Ikram, M. A., Vernooij, M. W., & de Bruijne, M. (2015). Transfer learning improves supervised image segmentation across imaging protocols. *IEEE Transactions on Medical Imaging*, *34*(5), 1018–1030. https://doi.org/10.1109/TMI.2014.2366792

[113] Van Opbroek, A., Achterberg, H. C., Vernooij, M. W., & De Bruijne, M. (2019). Transfer learning for image segmentation by combining image weighting and kernel learning. *IEEE Transactions on Medical Imaging*, *38*(1), 213–224. https://doi.org/10.1109/TMI.2018.2859478

[114] N. Miloslavskaya and A. Tolstoy, "Big data, fast data and data lake concepts," Procedia Comput. Sci., vol. 88, pp. 300–305, 2016.

[115] Nath, V., Yang, D., Landman, B. A., Xu, D., & Roth, H. R. (2021). Diminishing uncertainty within the training pool: Active learning for medical image segmentation. *IEEE Transactions on Medical Imaging*, *40*(10), 2534–2547. https://doi.org/10.1109/TMI.2020.3048055

[116] Li, W., Li, J., Wang, Z., Polson, J., Sisk, A. E., Sajed, D. P., Speier, W., & Arnold, C. W. (2022). PathAL: An active learning framework for histopathology image analysis. *IEEE Transactions on Medical Imaging*, *41*(5), 1176–1187. https://doi.org/10.1109/TMI.2021.3135002

[117] Wang, J., Chen, Z., Wang, L., & Zhou, Q. (2019). An active learning with two-step query for medical image segmentation. *2019 International Conference on Medical Imaging Physics and Engineering (ICMIPE)*, 1–5.

[118] Li, W., Zhang, M., & Chen, D. (2020). Fundus retinal blood vessel segmentation based on active learning. *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, 264–268.

[119] Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., & Martel, A. L. (2021). Loss odyssey in medical image segmentation. Medical Image Analysis, 71(102035), 102035. https://doi.org/10.1016/j.media.2021.102035

[120] S. A. Taghanaki et al., "Combo loss: Handling input and output imbalance in multi-organ segmentation," Comput. Med. Imaging Graph., vol. 75, pp. 24–33, 2019.

[121] X. P. Burgos-Artizzu, "Computer-aided covid-19 patient screening using chest images (X-Ray and CT scans)," bioRxiv, 2020.

[122] D. Kaspar et al., "Figure 3.4: The Hausdorff distance is defined as the maximum of all," ResearchGate. [Online]. Available: https://www.researchgate.net/figure/The-Hausdorff-distance-is-defined-as-the-maximum-of-all-minimum-distances-between-two_fig20_242202827. [Accessed: 09-Dec-2022].

[123] A. C. N. Matcha, "Class activation maps: Visualizing neural network decision-making," Heartbeat, 28-Oct-2019. [Online]. Available: https://heartbeat.comet.ml/class-activation-maps-visualizing-neural-network-decision-making-92efa5af9a33. [Accessed: 11-Nov-2022].

[124] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[125] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-CAM: Visual explanations from deep networks via Gradient-based localization. In arXiv [cs.CV].

[126] Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized gradient-based visual explanations

for deep convolutional networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 839–847.

[127] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[128] S. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” arXiv [cs.AI], 2017.

[129] W. Xu et al., “Early prediction of cerebral computed tomography under intelligent segmentation algorithm combined with serological indexes for hematoma enlargement after intracerebral hemorrhage,” Comput. Math. Methods Med., vol. 2022, p. 5863082, 2022.

[130]S. Shekhar, D. Eswaran, B. Hooi, J. Elmer, C. Faloutsos, and L. Akoglu, “Benefit-aware early prediction of health outcomes on multivariate EEG time series,” arXiv [cs.LG], 2021

[131]R. Ali et al., “A self-training deep neural network for early prediction of cognitive deficits in very preterm infants using brain functional connectome data,” Pediatr. Radiol., vol. 52, no. 11, pp. 2227–2240, 2022.

[132]Saleem, H., Shahid, A. R., & Raza, B. (2021). Visual interpretability in 3D brain tumor segmentation network. Computers in Biology and Medicine, 133(104410), 104410. https://doi.org/10.1016/j.compbiomed.2021.104410

[133]J. Wu et al., “Intelligent segmentation medical assistance system for MRI images of osteosarcoma in developing countries,” Comput. Math. Methods Med., vol. 2022, p. 7703583, 2022.

[134]W. Chen, M. Yao, Z. Zhu, Y. Sun, and X. Han, “The application research of AI image recognition and processing technology in the early diagnosis of the COVID-19,” BMC Med. Imaging, vol. 22, no. 1, p. 29, 2022.

[135]U. Niyaz, A. S. Sambyal, and Devanand, “Advances in deep learning techniques for medical image analysis,” in 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2018, pp. 271–277.

[136]Mattonen, S. A., Palma, D. A., Haasbeek, C. J. A., Senan, S., & Ward, A. D. (2014). Early prediction of tumor recurrence based on CT texture changes after stereotactic ablative radiotherapy (SABR) for lung cancer: Predicting recurrence after SABR using second-order texture statistics. Medical Physics, 41(3), 033502. https://doi.org/10.1118/1.4866219

[137] Eldesoky, A. R., Yates, E. S., Nyeng, T. B., Thomsen, M. S., Nielsen, H. M., Poortmans, P., Kirkove, C., Krause, M., Kamby, C., Mjaaland, I., Blix, E. S., Jensen, I., Berg, M., Lorenzen, E. L., Taheri-Kadkhoda, Z., & Offersen, B. V. (2016). Internal and external validation of an ESTRO delineation guideline – dependent automated segmentation tool for loco-regional radiation therapy of early breast cancer. Radiotherapy and Oncology: Journal of the European Society for Therapeutic Radiology and Oncology, 121(3), 424–430. https://doi.org/10.1016/j.radonc.2016.09.005

[138] de Oliveira, J. P. S., Conci, A., Perez, M. G., & Andaluz, V. H. (2015). Segmentation of infrared images: A new technology for early detection of breast diseases. 2015 IEEE International Conference on Industrial Technology (ICIT), 1765–1771.

[139] G. Danala et al., “Improving efficacy of metastatic tumor segmentation to facilitate early prediction of ovarian cancer patients’ response to chemotherapy,” in Biophotonics and Immune Responses XII, 2017, vol. 10065, pp. 47–52.

[140] Rekik, I., Allassonnière, S., Carpenter, T. K., & Wardlaw, J. M. (2012). Medical image analysis methods in MR/CT-imaged acute-subacute ischemic stroke lesion: Segmentation, prediction and insights into dynamic evolution simulation models. A critical appraisal. NeuroImage. Clinical, 1(1), 164–178. https://doi.org/10.1016/j.nicl.2012.10.003

[141] Zandbergen, E. G., de Haan, R. J., Stoutenbeek, C. P., Koelman, J. H., & Hijdra, A. (1998). Systematic review of early prediction of poor outcome in anoxic-ischaemic coma. *Lancet*, *352*(9143), 1808–1812. https://doi.org/10.1016/S0140-6736(98)04076-8

[142] Wu, B. U., Johannes, R. S., Sun, X., Tabak, Y., Conwell, D. L., & Banks, P. A. (2008). The early prediction of mortality in acute pancreatitis: a large population-based study. Gut, 57(12), 1698–1703. https://doi.org/10.1136/gut.2008.152702

[143] Hoeben, B. A. W., Troost, E. G. C., Span, P. N., van Herpen, C. M. L., Bussink, J., Oyen, W. J. G., & Kaanders, J. H. A. M. (2013). 18F-FLT PET during radiotherapy or chemoradiotherapy in head and neck squamous cell carcinoma is an early predictor of outcome. Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine, 54(4), 532–540. https://doi.org/10.2967/jnumed.112.105999

[144] Fang, Y.-J., Mukundan, A., Tsao, Y.-M., Huang, C.-W., & Wang, H.-C. (2022). Identification of early esophageal cancer by semantic segmentation. Journal of Personalized Medicine, 12(8), 1204. https://doi.org/10.3390/jpm12081204

[145] Liu, S., Liu, K., Zhu, W., Shen, Y., & Fernandez-Granda, C. (2022). Adaptive early-learning correction for segmentation from noisy annotations. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2606–2616.

[146] VinBigData, “10 open data repositories for the medical image processing community,” VinBigdata - Blog , 28-Jun-2021. [Online]. Available: https://blog.vinbigdata.org/10-kho-du-lieu-mo-danh-cho-cong-dong-xu-ly-anh-y-te/. [Accessed: 01-Oct-2022].

[147] N. Joshi, “4 Ways to Tackle the Lack of Machine Learning Datasets,” Bbntimes.com. [Online]. Available: https://www.bbntimes.com/technology/4-ways-to-tackle-the-lack-of-machine-learning-datasets. [Accessed: 01-Oct-2022].

[148] Z. Zhang, J. Li, Z. Zhong, Z. Jiao, and X. Gao, “A sparse annotation strategy based on attention-guided active learning for 3D medical image segmentation,” arXiv [cs.CV], 2019.

[149] U. Vovk, F. Pernus, and B. Likar, “A review of methods for correction of intensity inhomogeneity in MRI,” IEEE Trans. Med. Imaging, vol. 26, no. 3, pp. 405–421, 2007.

[150] “What is Overfitting in Deep Learning [+10 Ways to Avoid It],” V7labs.com. [Online]. Available: https://www.v7labs.com/blog/overfitting. [Accessed: 01-Oct-2022].

[151]V. Ashkani Chenarlogh et al., “Clinical target segmentation using a novel deep neural network: double attention Res-U-Net,” Sci. Rep., vol. 12, no. 1, p. 6717, 2022.

[152] Y. S. Jeon, H. Yang, and M. Feng, “FCSN: Global context aware segmentation by learning the Fourier coefficients of objects in medical images,” *arXiv [eess.IV]*, 2022.

[153] J. Sun, F. Darbehani, M. Zaidi, and B. Wang, “SAU-Net: Shape Attentive U-Net for Interpretable Medical Image Segmentation,” arXiv [eess.IV], 2020.

[154]C. Seibold, J. Künzel, A. Hilsmann, and P. Eisert, “From explanations to segmentation: Using Explainable AI for image segmentation,” arXiv [cs.CV], 2022.

[155]S. Ma, X. Li, J. Tang and F. Guo, "MIASNet: A medical image segmentation method predicting future based on past and current cases," 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021, pp. 1014-1019, doi: 10.1109/BIBM52615.2021.966963

An Ensemble Multi-layered Sentiment Analysis Model (EMLSA) for Classifying the Complex Datasets

Penubaka Balaji[1], D. Haritha[2]
Department of Computer Science & Engineering, K L University, Vaddeswaram, Andhra Pradesh, India

***Abstract*—Sentiment analysis is one domain that analyzes the feelings and emotions of the users based on their text messages. Sentiment analysis of short messages, reviews in online social media (OSM), and social networking sites (SNS) messages gives the analysis of given text data. Processing short text and SNS messages is a very tedious task because of the restricted detailed information generally contained. Solving this issue requires advanced techniques that are combined to give accurate results. This paper developed an Ensemble Multi-Layered Sentiment Analysis Model (EMLSA) that exploits the trust-based sentiment analysis on various real-time datasets. EMLA is the combined approach with VADER (Valence Aware Dictionary and sEntiment Reasoned) and Recurrent Neural Networks (RNNs). VADER is the lexicon and rule-based sentiment analysis model that predicts the sentiments extracted from input datasets and it is used for training. The feature extraction technique is term-frequency and inverse document frequency. Word-Level Embeddings (WLE) and Character-Level Embeddings (CLE) are the two models that increase the short text and single-word analysis. The proposed model was applied to four real-time datasets: Amazon, eBay, Trip-advisor, and IMDB Movie Reviews. The performance is analyzed using various parameters such as sensitivity, specificity, precision, accuracy, and F1-score.**

***Keywords*—Sentiment analysis; online social media; social networking sites; VADER; recurrent neural networks**

I. Introduction

Sentiment Analysis (SA) is the process of finding and dividing the opinions of the people expressed in text, voice, and videos. SA, also called opinion mining (OM), is natural language processing (NLP) that finds the emotions behind the body's text [1]. Opinions are expressed in various domains, such as movie reviews, e-commerce reviews, and Twitter reviews. Every day, many people and millions of reviews are generated by social media platforms regarding products, movies, and general topics [2]. An automated system is required to analyze the users' views, opinions, and sentiments. SA mainly focused on finding non-trivial, emotional information collected from various online sources belonging to social media [3]. Sentiment analysis can also be applied to multiple documents and phrases and analyzed single words. Finally, sentiment analysis divides the reviews into three types such as positive, negative, and neutral, based on the text data. Sentiment analysis helps e-commerce applications increase the sales of specific products [4] [5].

Natural language processing (NLP) is mainly focused on two aspects such as human language understanding and generation. It is a challenging task to analyze the natural language with the existing models. Several applications include speech recognition, text analysis, questioning and answering, synthesis of speech etc. [6]. NLP is divided into two significant areas such as sentiment analysis and recognition of emotions. Sometimes these two areas differ based on their aspects. "Emotion detection" is the domain that finds the feelings from the user's expressions like happiness, sadness, and depression. There is a significant connection between "sentiment analysis" and "emotion detection" [7]. From the emotions, the users express their feelings through text, video, and audio.

Sometimes sentiment analysis goes beyond people's opinions and views, such as sad, happy, angry, etc. [8]. Based on the feedback of the user or customer, sentiment analysis is required [9] [10] [11]. This paper describes various sentiment analyses belonging to several domains using deep learning algorithms with the integration of advanced fine-tuned models. The proposed approach focused on finding the sentiment analysis on multiple domains and analyzing the trust-based reviews in the input dataset. The proposed method also focused on aspect, Multilingual and emotion-based sentiment analysis.

II. Literature Survey

T. Gu et al. [12] proposed a novel sentiment analysis approach called MBGCV introduced to increase sentiment classification performance. MBGCV combined with various BiGRU, CNN, and VIB models. The proposed model obtained the high-level sentiment features from the given datasets. The real-time review dataset is used to analyze the performance of the proposed approach. M. K. Hayat et al. [13] introduced a DL model combined with the taxonomy-based approach to solving various issues in sentiment analytics. H. Liu [14] describes the comparative study among the lexicon, ML, and DL-based models that solve several accuracy issues. Various real-time datasets are used for experiments and analysis of sentiments. P. Gupta et al. [15] proposed the lexicon-based model that classifies the twitter data about COVID-19. The proposed model analysis the given Twitter data based on medicines, situations, and conditions faced by the users in lockdown time. This model aims to know the positive and negative opinions regarding the lockdown situation and the performance of the Indian government. Linear SVC is used to classify the data.

A. Elouardighi et al. [16] introduced the lexicon-based model combined with N-grams and TF-IDF model. The proposed approach is applied to comments in the Arabic language collected from Facebook. The data belongs to the Legislative Elections in Morocco in 2016. Several ML algorithms are used for performance evaluation, such as NB, RF, and SVM. Effective sentiment results were analyzed by using ML algorithms. P. Vyas et al. [17] introduced the framework that works on sentiment analysis regarding COVID-19. The proposed framework extracts the positive, negative, and neutral sentiments from the Twitter data and applies various ML algorithms present for classification. R. Khan et al. [18] introduced the deep LSTM model that predicts the sentiment polarity and emotions from the sentiment140 dataset. The accuracy of proposed model is 90.23%, this is very high compare with previous models. A. S. Imran et al. [19] introduced the LSTM model for detection of emotions and sentiments in terms of text messages collected from twitter. The main drawback of this model is lack of accuracy based on several emotions such as bad, good, anger. D. Antonakaki et al. [20] describe several DL models that work on sentiment analysis. The author mainly focused on three areas such as fake news, spam content, and threats messages given on Twitter. The proposed model analyzed the better sentiments based on the result analysis—the Twitter data used for performance evaluation. H. Strobelt et al. [21] proposed a model called as LSTMVIS that process the complex patterns present in various applications. S. Kumar et al. [22] proposed a hybrid recommended system that was applied to the movies dataset. The proposed model, combined with CF and CBF, provides a better recommendation system based on sentiment analysis. The proposed approach analyzed the present trends, people's sentiments, and users' responses. S. Bhatia [23] proposed a novel graph model that analyses duplicate phrases. The proposed approach focused on correcting the sentences by using graphs. To summarize the text and reduce the dimensions, PCA is used. The proposed method achieved better opinions mining based on sentiments.

S. Davis et al. [24] discussed various works on analyzing customer reviews based on E-commerce datasets. The comparative study shows the proposed approach applied to multiple user review datasets. M. A. Tayal et al. [25] submitted an integrated system based on several operations, such as pre-processing approach. Pre-processing is used to remove ambiguity from the given dataset. The proposed method mainly combines Semantic Sentence Similarity with n-gram co-occurrence relations belonging to specific sentences. Finally, the proposed model is applied to several benchmark datasets and analyzes the performances of existing and proposed models. E. Aslanian et al. [26] proposed the hybrid recommender system (HRS) that improves the high accuracy. The proposed approach, combined with the feature relationship matrix and collaborative filtering, was used to solve the cold-start problem. The proposed method achieves better accuracy compared with other existing algorithms. E. Cambria [27] proposed an automated approach for analyzing sentiments based on emotions. The proposed system combines emotions and reviews and gives better performance.

C. Du et al. [28] proposed a new classification approach that classifies the sentiment data using an advanced feature extraction technique. The softmax classifier is adopted to increase the proposed system's performance. The F1-score of the proposed approach shows the high values for two datasets. Maria Giatsoglou et al. [29] proposed a rapid and reliable model that finds the sentiments of different types of people's opinions from other languages. The ML approach combined with the proposed approach applied to text documents initialized by vectors and trained as a polarity classification model. The proposed model is analyzed using four datasets containing reviews in Greek and English.

III. PROPOSED METHODOLOGY

The proposed methodology is developed with various advanced models such as the pre-trained DL model stemming model for pre-processing, TF-IDF for feature extraction, Word-Level Embeddings (WLE) for text analysis, VADER for training and RNN for classification of text data. Fig. 1 shows the step by step process of implementation.

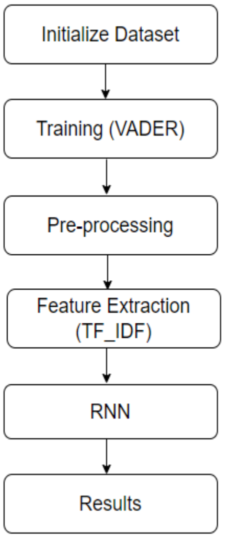


Fig. 1. System architecture.

IV. VADER

This paper uses VADER to train the given datasets to analyze the sentiments. It is the lexical database developed by using rule-based sentiment analysis. The lexicon collects the features (e.g., words) classified as positive or negative based on the sentiment polarity. VADER shows the positivity and negativity scores and also the strength of the positive and negative sentiments. The VADER is mainly based on a compound score measured by aggregation of valence scores of every word in the lexicon, find-tuned based on rules, and then normalized between -1 (high negative) and +1 (high positive). Thus this is considered the single uni-dimensional measure of sentiment for a given sentence.

$$x = \frac{x}{\sqrt{x^2 + \alpha}} \tag{1}$$

Where x = sum of valence scores of constituent words, and α = Normalization constant (default value is 15).

A. TF-IDF (Term Frequency- Inverse Document Frequency)

TF-IDF is a feature extraction approach that can extract highly reputed words in the given documents and reviews. TF mainly measures the frequently appeared mentions in the given input datasets. The term frequency refers to the total time that appeared in the given input datasets, while the document frequency refers to complete documents that contain the word. IDF counts the word from papers or reviews divided by the phrase "document frequency." Every word initialized the score by measuring the TF by its IDF. Here features mean repeated words from multiple reviews from multiple documents.

B. Word-Level Embeddings (WLE)

WLE's are encoded by using column vectors within the embedding matrix $W^{wrd} \in R^{d^{wrd} \times |V^{wrd}|}$. Every column belongs to WLE of kth word in the vocabulary. By using matrix-vector product, the word W transformed into WLE r^{wrd}.

$$r^{wrd} = W^{wrd} v^{w}) \qquad (2)$$

Where v^{w} is size of vector $|V^{wrd}|$ which is value 1 at index w and 0 in all other portions. W^{wrd} → matrix. That learns and WLE size is given as d^{wrd} is hyper-parameter which is selected by the user.

C. RNN

All these layers are fully connected and are not associated with each other. RNN [30] performs better on the text sentiments dataset, and all the tasks involve sequential inputs. RNN considers one piece of information at a time and maintains the hidden units of a "state vector" consisting of data regarding the previous history based on the sequence. The outputs of hidden units are considered at various discrete time steps if the results of several neurons in a deep multi-layer network, this becomes easy to implement back-propagation to train RNN. RNN is a dynamic approach, and it is mighty to prepare them and solves the issues in back-propagated gradients either grow or shrink at each step; several times, this process is typically terminated as shown in Fig. 2.

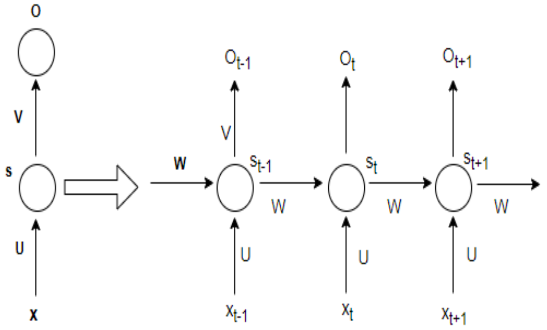


Fig. 2. Process of RNN.

The neurons in the hidden layers get the inputs from previous layers based on the time steps. Based on the above process, the RNN maps the input sequence elements represented as x_t, and the output sequence represents the elements with o_t, dependent on the previous x t′ (for t′ ≤ t). Similar metrics such as U, V, and W are utilized at every step. The back-propagation approach measures the unfolded network on the right and computes overall error based on general states s_t and all the metrics.

V. Dataset Description

1) Amazon dataset: The Amazon dataset consists of testing and training data. The training data contains 3Lakh, and testing data consists of 4Lakh data belonging to 568,000 customer's data. All these customers give reviews of the products. This is open source and free dataset collected from Kaggle: https://www.kaggle.com/datasets/bittlingmayer/amazonreviews?resource=download.

2) Ebay dataset: A data science Bootcamp project created the eBay dataset. This project aims to develop the best model for sentiment analysis. The author created this dataset using python web scraping scripts for the research work. This dataset consists of two files as ebay_reviews.csv file consists of four attributes: product category, title review, content review, and rating. The total instances are 44757. The rating attribute represents the integer value with one as the worst score and five gives the best score. The second file is a preprocessed file that consists of two attributes: rating, title review, and content review. The dataset available at: https://www.kaggle.com/data sets/wojtekbonicki/ebayreviews/discussion?select=ebay_reviews.csv.

3) Trip-advisor: This dataset consists of 20k reviews of various hotels given by customers. Trip-advisor extracts these reviews, and it is available on the Kaggle website. The dataset available at: https://www.kaggle.com/datasets/andrewmvd/tripadvisorhotel-reviews IMDB Movie Reviews Dataset: This dataset consists of 50k movie reviews and this contains 40k testing and 10 k training data. IMDB movie review dataset consists of 25k positive and 25k negative reviews and this data available at:

VI. Performance Metrics

The performance of the proposed model is analyzed by using a confusion matrix. The confusion matrix mainly measures the accurate values obtained from the proposed model. Based on the count of the importance, the performance is measured. The proposed approach focused on developing a model which can work on any dataset. The factors that show an impact on the performance of the proposed model are a true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

TP: The predicted value is true, and it is true.

TN: The predicted values is no and it is false.

FP: The predicted values is yes, originally it is not true.

FN: The predicted values is no, but the values are true.

Precision: The total positives measured from the overall positives give precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

Accuracy: The model accuracy is identified.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (4)$$

Recall: The overall TPs are identified.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+ FN}} \quad (5)$$

Specificity: The overall false values are correctly identified.

$$\text{Specificity} = \frac{\text{No of TN}}{\text{No of TN+ No of FP}} \quad (6)$$

F1-Score: This measure the coherence mean of Precision and Recall achieved the better computation which is incorrectly classified cases than the Accuracy.

$$\text{F1} - \text{Score} = 2 * \frac{(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})} \quad (7)$$

Result Analysis: From the results it is analyzed that the performance of various existing and proposed algorithms are given in Table I to IV. The comparative performance of RTA, IDER and EMLSA is implemented with four datasets. The proposed model EMLSA performed better on all the datasets by analyzing sentiments compared with existing models. The performance is measured by confusion matrix measures such as precision, accuracy, recall and specificity (see Fig. 3 to 6)

TABLE I. COMPARATIVE ANALYSIS OF EXISTING AND PROPOSED ALGORITHMS FOR AMAZON DATASET

Algorithms	**Precision**	**Accuracy**	**Recall**	**Specificity**	**F1-Score**
Reputational Trust Assessment (RTA) [31]	89.34	90.34	88.67	88.89	89.54
IDER [32]	93.21	92.23	90.45	93.89	94.9
EMLSA	98.78	98.45	97.45	98.43	98.45

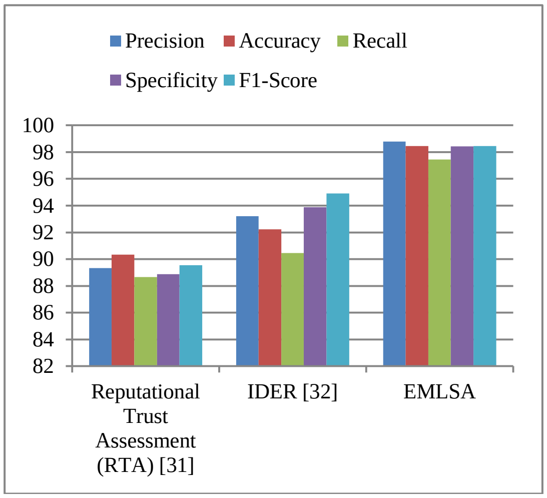


Fig. 3. Graph representation for existing and proposed algorithms for amazon dataset.

TABLE II. COMPARATIVE ANALYSIS OF EXISTING AND PROPOSED ALGORITHMS FOR EBAY DATASET

Algorithms	**Precision**	**Accuracy**	**Recall**	**Specificity**	**F1-Score**
Reputational Trust Assessment (RTA) [31]	87.12	88.56	87.23	87.34	88.65
IDER [32]	92.34	92.56	92.98	92.67	93.5
EMLSA	97.34	97.78	98.34	97.65	97.12

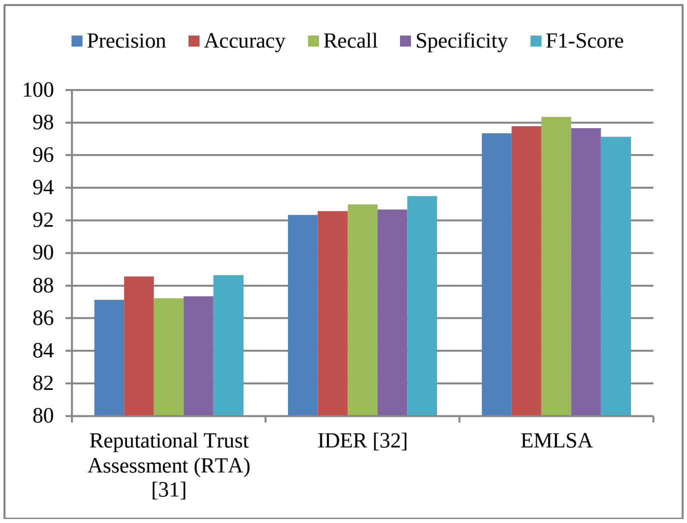


Fig. 4. Graph representation for existing and proposed algorithms for eBay dataset.

TABLE III. COMPARATIVE ANALYSIS OF EXISTING AND PROPOSED ALGORITHMS FOR TRIP-ADVISOR

Algorithms	**Precision**	**Accuracy**	**Recall**	**Specificity**	**F1-Score**
Reputational Trust Assessment (RTA) [31]	88.96	89.12	88.32	88.67	88.32
IDER [32]	93.45	94.67	93.87	93.45	93.76
EMLSA	99.23	99.56	99.22	98.11	98.78

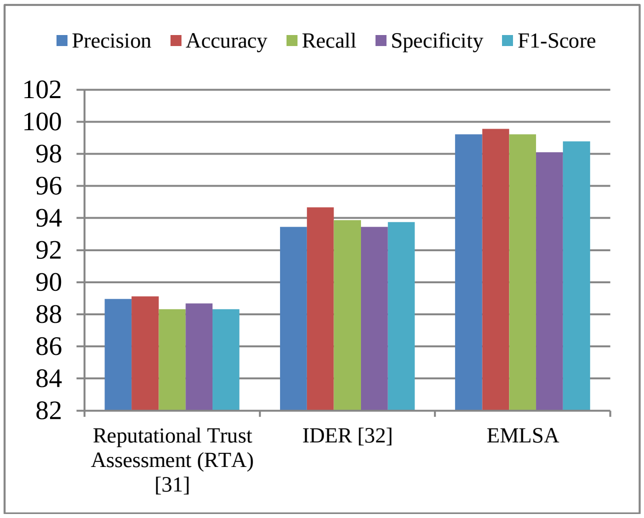


Fig. 5. Graph representation for existing and proposed algorithms for trip-advisor.

TABLE IV. PERFORMANCE OF EXISTING AND PROPOSED ALGORITHMS FOR IMDB MOVIE REVIEWS DATASET

Algorithms	Precision	Accuracy	Recall	Specificity	F1-Score
Reputational Trust Assessment (RTA) [31]	91.89	91.78	91.45	91.99	91.23
IDER [32]	94.34	94.1	94.9	94.6	94.67
EMLSA	99.8	99.8	99.7	99.3	99.43

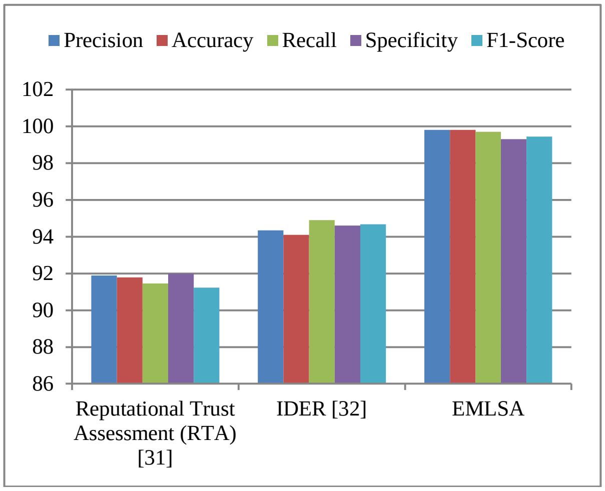


Fig. 6. Graph representation for existing and proposed algorithms for IMDB movie reviews dataset.

VII. CONCLUSION

Even though a conclusion may review the main results this paper describes the new DL model that can process complex datasets based on the reviews given by the users. The proposed approach was applied to four benchmark datasets that show the comparative performance in terms of sensitivity, specificity, accuracy, precision, and f1-score. The proposed DL model focused on extracting every aspect of the input reviews. The word embedding models TF-IDF, and Word2Vec combined with the DL model give high performance in terms of given input datasets. The proposed model achieved an accuracy of 98.45% for the amazon dataset, 97.78% for the TripAdvisor dataset, and 99.56% for the ebay dataset and for IMDB dataset the accuracy is 99.8%. Thus it is shown that the accuracy is more for the proposed model. In future, the multi-layered models are to be developed by improving the sentiments and emotion detection. Various combined and integrated models are required to increase the performance.

REFERENCES

[1] Cambria E, Dragoni M, Kessler B, Donadello I. Ontosenticnet 2: enhancing reasoning within sentiment analysis. IEEE Intell Syst. 2022;37(2):103–110.

[2] Cambria E, Xing F, Thelwall M, Welsch R. Sentiment analysis as a multidisciplinary research area. IEEE Trans Artif Intell. 2022;3(2):1–4.

[3] Chan JYL, Bea KT, Leow SMH, Phoong SW, Cheng WK. State of the art: a review of sentiment analysis based on sequential transfer learning. Artif Intell Rev. 2022 doi: 10.1007/s10462-022-10183-8.

[4] Cheng WK, Bea KT, Leow SMH, Chan JY-L, Hong Z-W, Chen Y-L. A review of sentiment, semantic and event-extraction-based approaches in stock forecasting. Mathematics. 2022;10(14):2437. doi: 10.3390/math10142437.

[5] Da'u A, Salim N, Rabiu I, Osman A. Recommendation System Exploiting Aspect-Based Opinion Mining with Deep Learning Method. Inf Sci. 2020;512:1279–1292. doi: 10.1016/j.ins.2019.10.038.

[6] Itani M, Roast C, Al-Khayatt S (2017) Developing resources for sentiment analysis of informal Arabic text in social media. Procedia Comput Sci 117:129–136.

[7] Munezero M, Montero CS, Sutinen E, Pajunen J (2014) Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. IEEE Trans Affect Comput 5(2):101–111.

[8] Penubaka Balaji, D Haritha, "Feature Based Summarization System for E-Commerce Based Products by Using Customer's Reviews," CCODE-2018, http://dx.doi.org/10.2139/ssrn.3168342.

[9] Penubaka Balaji, O. Nagaraju, D Haritha, "Levels of sentiment analysis and Its challenges", ICBDACI-2017,doi:10.1109/icbdaci.2017. 8070879

[10] Penubaka Balaji, O. Nagaraju, D Haritha, "CommuTrust: Reputation based trust evaluation in E-Commerce applications", ICBDACI-2017, doi: 10.1109/icbdaci.2017.8070856.

[11] Penubaka Balaji, O. Nagaraju, D Haritha, "An Overview on Opinion Mining Techniques and Sentiment Analysis", IJPAM, Vol.118 & Issu.19, Jan 2018.

[12] M.K. Rahmat, S. Jovanovic, K.L. Lo, Reliability and Availability T. Gu, G. Xu and J. Luo, "Sentiment Analysis via Deep Multichannel Neural Networks With Variational Information Bottleneck," in IEEE Access, vol. 8, pp. 121014-121021, 2020, doi: 10.1109/ACCESS.2020.3006569.

[13] M. K. Hayat et al., "Towards Deep Learning Prospects: Insights for Social Media Analytics," in IEEE Access, vol. 7, pp. 36958-36979, 2019, doi: 10.1109/ACCESS.2019.2905101.

[14] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu and A. Abusorrah, "Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods," in IEEE Transactions on Computational Social Systems, vol. 7, no. 6, pp. 1358-1375, Dec. 2020, doi: 10.1109/TCSS.2020.3033302.

[15] P. Gupta, S. Kumar, R. R. Suman and V. Kumar, "Sentiment Analysis of Lockdown in India During COVID-19: A Case Study on Twitter," in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 992-1002, Aug. 2021, doi: 10.1109/TCSS.2020.3042446.

[16] A. Elouardighi, M. Maghfour, H. Hammia and F. -z. Aazi, "A machine Learning approach for sentiment analysis in the standard or dialectal Arabic Facebook comments," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), 2017, pp. 1-8, doi: 10.1109/CloudTech.2017.8284706.

[17] P. Vyas, M. Reisslein, B. P. Rimal, G. Vyas, G. P. Basyal and P. Muzumdar, "Automated Classification of Societal Sentiments on Twitter With Machine Learning," in IEEE Transactions on Technology and Society, vol. 3, no. 2, pp. 100-110, June 2022, doi: 10.1109/TTS.2021.3108963.

[18] R. Khan, R. Khan, P. Shrivastava, A. Kapoor, A. Tiwari and A. Mittal, "Social media analysis with AI: Sentiment analysis techniques for the analysis of Twitter COVID-19 data", J. Crit. Rev., vol. 7, no. 9, pp. 2761-2774, 2020.

[19] A. S. Imran, S. M. Doudpota, Z. Kastrati and R. Bhatra, "Cross-cultural polarity and emotion detection using sentiment analysis and deep learning—A case study on COVID-19", IEEE Access, vol. 8, pp. 181074-181090, 2020.

[20] D. Antonakaki, P. Fragopoulou and S. Ioannidis, "A survey of Twitter research: Data model graph structure sentiment analysis and attacks", Expert Syst. Appl., vol. 164, Feb. 2021, [online] Available: https://doi.org/10.1016/j.eswa.2020.114006.

[21] H. Strobelt, S. Gehrmann, H. Pfister, And A. M. Rush, "Lstmvis: A Tool For Visual Analysis Of Hidden State Dynamics In Recurrent Neural Networks," Ieee Trans. Vis. Comput. Graph., 2018.

[22] S. Kumar, K. De and P. P. Roy, "Movie Recommendation System Using Sentiment Analysis From Microblogging Data," in IEEE Transactions on Computational Social Systems, vol. 7, no. 4, pp. 915-923, Aug. 2020, doi: 10.1109/TCSS.2020.2993585.

[23] S. Bhatia, "A Comparative Study of Opinion Summarization Techniques," in IEEE Transactions on Computational Social Systems, vol. 8, no. 1, pp. 110-117, Feb. 2021, doi: 10.1109/TCSS.2020.3033810.

[24] S. Davis and N. Tabrizi, "Customer Review Analysis: A Systematic Review," 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD), 2021, pp. 91-97, doi: 10.1109/BCD51206.2021.9581965.

[25] M. A. Tayal, M. M. Raghuwanshi and L. G. Malik, "ATSSC: Development of an approach based on soft computing for text summarization" in Comput. Speech Lang., vol. 41, pp. 214-235, Jan. 2017.

[26] E. Aslanian, M. Radmanesh and M. Jalili, "Hybrid recommender systems based on content feature relationship", IEEE Trans. Ind. Informat., Nov. 2016.

[27] E. Cambria, "Affective computing and sentiment analysis", IEEE Intell. Syst., vol. 31, no. 2, pp. 102-107, Mar./Apr. 2016.

[28] C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks", Int. J. Comput. Commun. Control, vol. 13, no. 1, pp. 50-61, 2018.

[29] Maria Giatsoglou, Manolis G. Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, Konstantinos Ch. Chatzisavvas, Sentiment analysis leveraging emotions and word embeddings, Expert Systems with Applications, Volume 69, 2017, Pages 214-224, https://doi.org/10.1016/j.eswa.2016.10.043.

[30] Wang H, Raj B, Xing E P. On the origin of deep learning. 2017.

[31] Penubaka, Balaji & Haritha, D.. (2018). Opinion Mining Based Reputational Trust Assessment in E-Commerce Applications. Journal of Advanced Research in Dynamical and Control Systems. 10.

[32] Donavalli, H. & Penubaka, Balaji. (2019). Identification of opinionated features extraction from unstructured textual reviews. International Journal of Recent Technology and Engineering. 7. 674-677.

Dynamic Programming Approach in Aggregate Production Planning Model under Uncertainty

Umi Marfuah[1], Mutmainah[2], Andreas Tri Panudju[3], Umar Mansyuri[4]
Department of Industrial Engineering-Faculty of Technology, Universitas Muhammadiyah Jakarta, Indonesia[1, 2]
Department of Industrial Engineering-Faculty of Science and Technology, Bina Bangsa University, Serang, Indonesia[3]
Department of Information System-Faculty of Science and Technology, Bina Bangsa University, Serang, Indonesia[4]

***Abstract*—In order to achieve a competitive edge in the market, one of the most essential components of effective operations management is aggregate production planning, abbreviated as APP. The sources of uncertainty discussed in the APP model include uncertainty in demand, uncertainty of production costs, and uncertainty of storage costs. The problem of APP usually involves many imprecise, conflicting and incommensurable objective functions. The application of APP in real conditions is often inaccurate, because some information is incomplete or cannot be obtained. The aim of this study is to develop APP model under uncertainty with a dynamic programming (DP) approach to meet consumer demand and minimize total costs during the planning period. The APP model includes several parameters including market demand, production costs, inventory costs, production levels and production capacity. After describing the problem, the optimal APP model is formulated using artificial neural network (ANN) techniques in the demand forecasting process and fuzzy logic (FL) in the DP framework. The ANN technique is used to forecast the input demand for APP and minimize the total cost during the planning period using the FL technique in the DP framework to accommodate uncertainties. The model input is historical data obtained through interviews. A case study was conducted on the the need for aluminum plates for the automotive industry. The results show that the ANN technique proposed for demand projection has a low error value in forecasting demand and FL in the DP framework is able to find minimal production costs in the APP model.**

***Keywords*—Aggregate production planning; artificial neural network; dynamic programming; fuzzy logic**

I. Introduction

A form of intermediate production planning known as aggregate production planning, or APP, has a time horizon of three to eighteen months and is used to establish the optimum solution level of production, stockpiles, and personnel management for each planning period within the form of limited factors of production and other constraints [1], [2]. The preferred APP strategy is capacity. Strategies for capacity choice include changing inventory levels[3]; varying the size of the workforce by hiring or firing[4]; varying production rates through overtime and idle time[5]; subcontract; using part time workers [6].

Aggregate planning is a complex issue mainly due to the need to coordinate the interacting variables so that the company can respond to requests in an effective manner [7]. The APP activity hierarchies is positioned somewhere in the between of long-term strategic alignment like new product development and short-term scheduling procedures on the factory floor [8]. The APP model is for operations managers with operations planning and sales teams.

Based on the number of objective functions, which are considered in the model, the APP model can be classified into two categories namely single objective function and multiple objective function [9]. The general purpose function in the APP model is to minimize the total system cost [7], [8], [10].

The nature of the data or input parameters in real-world APP issues, such as those involving demand, resources, costs, objective function coefficients, etc., is inherently imprecise due to the fact that some information cannot be retrieved or is unavailable in its whole [11]. In business practice, products usually have an uncertain demand and variable [12], customer preferences change, production capacity is limited [13], labor market conditions are unstable, subcontracting can incur higher costs[14], uncertainty of raw material supply [15], and an increase in backorders caused customer claim and led them to change the source of their purchases[8], [16]. This demonstrates the complex characteristics of APP and an appropriate APP model is needed.

The forecasts of future demand are the most important input for the creation of the APP strategy. A highly unpredictable demand results in frequent revisions of production planning from one planning period to the next [8], [15], [17]. This not only results in anxiety and nervousness within the production environment [4], but it is also one of the primary drivers of costs due to its adverse effects on labor and supply levels [5].

Artificial Neural Network (ANN) [18] algorithms have indeed been noticed to be effective methods for prediction due to their ability to facilitate non-linear data, to acquire delicate functional relationships among empirical data, even in cases where the underlying relationships are hard to explain or are unidentified. This is because ANN algorithms have the ability to accommodate non-linear data [19].

Conventional APP problem assumes market demand is crisp value [20], difficulty estimating crisp demand is overcome by using fuzzy demand which also increases estimation flexibility and results in better production plans that increase profits [13], [21].

Dynamic programming (DP) is a powerful optimization tool for dealing with complex problems involving sequential or multi-stage decision making in many fields [22].

As a result of the intrinsic subjectivity of people as well as the fuzziness with which they articulate their thoughts, there are a great deal of aspects that are imprecise and ambiguous. When applied in a context where there is uncertainty, doing an analysis of an issue involving multi-stage decision making using traditional DP can be challenging [23]. There are several reasons for this. DP is one of the earliest essential approaches in which fuzzy set theory is applied [24]. This is assuming that Zadeh's fuzzy set theory is the correct way to deal with uncertainty and imprecision in real-world issues [25], which leads to what is called fuzzy dynamic programming (FDP). One of the FDP applications has been used to find optimal routes with minimal costs on the problem of shipping goods from city one to city ten [26].

Key contribution of this paper are:

1) *Formulating the aggregate production planning (APP) problem as a Markov decision process (MDP) under uncertainty:* The authors developed a mathematical model to represent the APP problem in a stochastic environment. They formulated the problem as an MDP, which allowed them to take into account the uncertain variables that affect production planning decisions, such as demand and supply constraints.

2) *Applying a dynamic programming approach to solve the APP problem:* The authors used the value iteration algorithm to solve the MDP and determine the optimal production plan for each period. The dynamic programming approach allowed them to find the optimal solution for the APP problem by breaking it down into smaller subproblems and solving them recursively.

3) *Developing a scenario-based approach to model uncertainty:* The authors used a scenario-based approach to generate possible outcomes of uncertain variables, such as demand and supply constraints. This approach allowed them to create a set of scenarios that capture the uncertainty in the APP problem and formulate a stochastic optimization problem.

4) *Evaluating the proposed approach on a case study:* The authors evaluated the effectiveness of their proposed approach on a case study involving a manufacturing company. They compared the results of their approach with those obtained from a traditional linear programming model and found that the proposed approach was more effective in addressing uncertainty and generating optimal production plans.

Overall, the key contributions of the paper are the development of a dynamic programming approach to solve the APP problem under uncertainty and the application of a scenario-based approach to model uncertainty in the optimization problem. These contributions have the potential to improve production planning decisions for manufacturing companies facing uncertain demand and supply constraints.

Therefore, this study aims to formulate an optimal APP model with a DP framework that combines ANN and FL techniques. The ANN technique is used to forecast the input demand for APP and the preparation of APP using the FL technique in the DP framework.

This paper is divided into several sections that cover different aspects of the proposed approach: 1)Introduction: The introduction provides an overview of the problem of aggregate production planning (APP) under uncertainty and highlights the need for a dynamic programming approach to solve it. The authors introduce the concept of Markov decision processes (MDPs) and explain how they can be used to model the APP problem. 2) Methodology: The authors present the mathematical model that they developed to represent the APP problem as an MDP under uncertainty. They explain the variables and constraints that are included in the model and describe how it can be used to determine the optimal production plan for each period. 3) Result and Discussion. This section explains the value iteration algorithm that the authors used to solve the MDP and find the optimal production plan for each period. They describe the algorithm in detail and provide a step-by-step explanation of how it can be used to solve the APP problem. The authors introduce a scenario-based approach to model uncertainty in the APP problem. They explain how this approach can be used to generate possible outcomes of uncertain variables and describe how it can be used to formulate a stochastic optimization problem. The authors present a case study involving a manufacturing company to evaluate the effectiveness of their proposed approach. They compare the results obtained from their approach with those obtained from a traditional linear programming model and demonstrate the superiority of their approach in addressing uncertainty and generating optimal production plans. 4) Conclusion: The conclusion summarizes the key contributions of the paper and highlights the potential benefits of the proposed approach for manufacturing companies facing uncertain demand and supply constraints. The authors also suggest areas for future research and development in this field.

II. Methodology

A. Research Framework

The work step of this research is to first identify the affected factors in the APP system. Then build a demand forecasting model using the ANN technique which will be used as input to build the APP model. Finally, build the APP model using FL techniques within the DP framework. The research framework can be seen in Fig. 1.

B. Artificial Neural Networks (ANN)

The notion that underpins ANN is that the input, also known as the dependent variable, is passed through one or more hidden layers, each of which is composed of hidden units, or nodes, before it reaches the variable that is being measured as the output [27]. For the purposes of series data modeling and forecasting, the type of neural network model that sees the most widespread use is the single hidden layer feed - forward neural network [28].

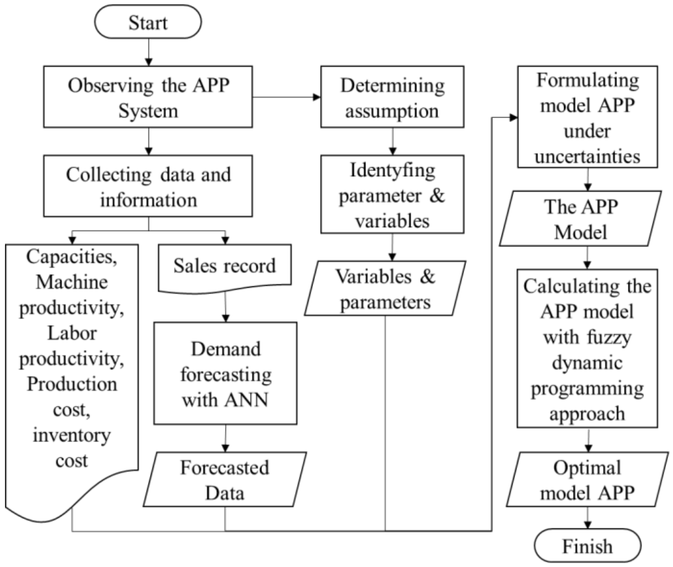

Start
Observing the APP System
Determining assumption
Formulating model APP under uncertainties
Collecting data and information
Identyfing parameter & variables
The APP Model
Capacities, Machine productivity, Labor productivity, Production cost, inventory cost
Sales record
Variables & parameters
Calculating the APP model with fuzzy dynamic programming approach
Demand forecasting with ANN
Forecasted Data
Optimal model APP
Finish

Fig. 1. Research framework.

According to the standard concept, the connection between outputs (y_t) and inputs ($y_{t-1},...,y_{t-p}$) is as follows:

$$y_t = w_0 + \sum_{j=1}^{q} w_j \,.\, g\left(w_{0j} + \sum_{i=1}^{p} w_{ij} \,.\, y_{t-1}\right) + \varepsilon_t$$

Where:

$w_{i,j}$ (i = 0,1,2, ..., p, j = 1,2, ..., q) and w_j (j = 0,1,2, ..., q) are the parameter that represents the model weight;

p represents the number of input nodes, and

q represents the number of hidden nodes.

In other words, the recurrent neural network receives the values y left as input, and it also has a hidden layer that is comprised of size q nodes [29]. In point of fact, then, the model executes a nonlinear functional mapping from historical data to those of the future:

$$y_t = f\left(y_{t-1}, \ldots\ldots, y_{t-p}, w\right) + \varepsilon_t$$

This considers that w is a vector of all parameters and f () is a function defined by the network structure and the weights for the connection. The ANN applied in this study is the default single hidden layer model using the feed-forward backpropagation algorithm [30], where the extension of the nodes number in one layer is equal to the input nodes number plus 1 (Fig. 2). The quantity of grids that were put in place, each having a starting weight that was chosen at random, and then averaged as estimates are calculated [31]. ANN technique is used in the demand forecasting process using Matlab R2017b software.

C. Dynamic Programming

Dynamic Programming is a strong formal instrument that may be utilized for the purpose of addressing a wide variety of multi-stage decision-making issues [32]. Since its origin in the middle of the 1950s by Bellman 1957, DP has developed into a common tool in a variety of fields [26], including but not limited to operations research, systems analysis, engineering, data analysis, control, and computer science, amongst others [22], [2], [33]. The fact that one only needs to solve a little fraction of each subproblem in order to complete DP successfully is one of its strengths [34]. This is because of Bellman's concept of optimality, which explains the situation. According to this, regardless of the decisions that were made in the stage before it, if the decisions that are going to be made in stage n are going to be a part of the overall optimal solution, then the decisions that are going to be made at stage n have to be optimal for all of the stages that come after it [34].

At each stage, n in the DP there are state variables, x_n, and optimum decision variables, d_n.

In stage n, there is a value returned by the function for each of the values xn and dn, $r_n(x_n,d_n)$. The result of the procedure once it has reached step n is x_{n-1}, the status variable for the stage n-1. The stage transformation function is responsible for calculating this result, $x_{n-1} = x_n + d_n - D_n$ which means inventory plus production minus demand (D_n). The optimum value function, denoted as $fn(xn)$, is the combined total return beginning at step n in the state xn and proceeding to stage 1 in accordance with the best possible strategy.

In general, the most effective way to tackle problems with dynamic programming is to begin at the end of the process and work your way backwards to the beginning. This is referred to as recursion in reverse. The recursive connection that is presented here may be utilized in order to implement the concept of optimal solutions in the context of achieving the lowest possible total cost:

$$f_n(x_n) = \min_{d_n}\{r_n(x_n, d_n) + f_{n-1}(x_n + d_n - D_n)\}$$

Where $r_n(x_n, d_n)$ is the total production and storage costs for the stage / month n. Production costs are the cost of production per unit multiplied by the number of units produced (d_n). Storage cost is the ending inventory for the month multiplied by the unit cost of storage. Mathematically, it is written as follows:

$$r_n(x_n, d_n) = C_n d_n + H_n(x_n + d_n - D_n)$$

Since there is no provision for backordering, it must fulfill the needs of the customers. That is, for the month n:

$$x_n + d_n \geq D_n$$

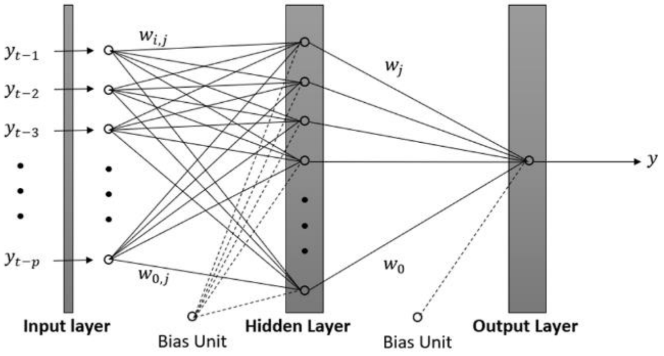


Fig. 2. Neural network structure.

Because there is a storage capacity limit of *Wn* at each and every stage *n*, the overall inventory at the conclusion of any given month cannot be greater than *Wn*. So that for every month it must be:

$$x_n + d_n - D_n \leq W_n, or$$

$$x_n + d_n \leq W_n + D_n$$

It is imperative that the quantity generated in any given month does not go over the capability of production for that month, or:

$$d_n \leq P_n$$

Beginning at stage 0 with the boundary conditions $f_0(x_0) = 0$, a problem can be solved by working backwards through the stages until reaching the final stage, *n*. It is assumed that there are no products stored in inventory at the beginning and end of the planning period.

D. Fuzzy Dynamic Programming

A fuzzy set may be identified by its one-of-a-kind membership function, which is responsible for mapping each component of the X discourse environment to the interval [0,1].

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)); x \in X\}$$

A function related to the degree of membership A˜ determined by $\mu_{\tilde{A}}(x): X \rightarrow [0,1]$. In a fuzzy set $\tilde{A}$, $\mu_{\tilde{A}}(x)$ is the degree of membership $x \in X$.

A fuzzy set that is defined on the uniform real numbers is denoted by the symbol A, is said to be a Triangular Fuzzy Number (TFN) if there are three arguments with the value $\tilde{A}$ (a_1, a_2, a_3) determined by the following membership functions:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2 \\ \dfrac{a_3 - x}{a_3 - a_2}, & a_2 \leq x \leq a_3 \\ 0, & x > a_3 \end{cases}$$

TFN may also be expressed in the form of an interval: [d1,d2], where $d_1 = (a_2 - a_1)\alpha + a_1$ and $d_2 = -(a_3 - a_2)\,\alpha + a_3$ [35]. Fuzzy numbers can be processed mathematically fuzzy according to the method of representation. Representation in the interval for example $\tilde{A} = [d_1, d_2]$ and $\tilde{B} = [e_1, e_2]$ be a representation of two TFN numbers, the arithmetic operation is as follows:

1) $\tilde{A} + \tilde{B} = [d_1 + e_1, d_2 + e_2]$
2) $\tilde{A} \times \tilde{B} = [d_1 \times e_1, d_2 \times e_2]$
3) $c \times \tilde{A} = [c \times d_1, c \times d_2]$

Fuzzy aggregate production planning problems are related to the uncertainty of demand, production costs and storage costs with the aim of producing minimal total costs during the planning period. Suppose an industry must produce goods from one period to another with different production costs and storage costs in each period. So a plan is needed to determine the amount of production per period with the lowest cost that takes into account the demand for each period.

In order to address fuzzy APP issues utilizing the fuzzy dynamic programming approach, one must first follow the methods that are mentioned below. Step 1: Determine that there is an issue with hazy choice variables and then to state that the fuzzy objective function is going to be optimized using definite bounds. Step 2: The problem that has to be addressed is then broken down into smaller subproblems or stages. Classify the fuzzy condition variables at each level, and then create the transformation function such that it is a function of both the fuzzy condition variables at the previous stage and the fuzzy decision variables at the stage after that. Step 3: Then using generalized fuzzy recursive relationships we get the optimal decision for the problem. $\widetilde{f_n}(x_n, d_n)$ is the lowest possible sum of money spent on the previous *n* stages. $\widetilde{f_n^*}(x_n)$ is the optimal value (minimum cost) when the product is in the state x_n with *n* stage again to reach the final stage. The optimal value equation of $\widetilde{f_n}$ on condition x_n can be obtained by selecting the appropriate decision on the decision variable d_n that is:

$$\tilde{f}_n^*(x_n) = \min_{d_n}\{\tilde{r}_n(x_n, d_n) + \tilde{f}_{n-1}(x_{n-1})\}$$

Range d_n determined by x_n, but x_n defined by the events that took place in the stage before it. The return function will then assume its final shape in the subsequent stage:

$$\tilde{r}_n(x_n, d_n) = \tilde{C}_n d_n + \widetilde{H}_n\big(x_n + d_n - \widetilde{D}_n\big)$$

So that the fuzzy recursive equation is obtained as follows:

$$\tilde{f}_n^*(x_n) = \min_{d_n}\{(\tilde{C}_n d_n + \widetilde{H}_n\big(x_n + d_n - \widetilde{D}_n\big) + \tilde{f}_{n-1}(x_{n-1})\}$$

Step 4: Create a suitable table to show the importance of the return function at each stage. Step 5: Determine the overall optimal decision and its value.

The case study reported in this research is aluminum industry which processes aluminum plates for the automotive industry.

III. Results and Discussion

A. Demand Forecasting

Demand forecasting is done using the backpropagation feed-forward algorithm in the Matlab R2017b software. The input data used consisted of data on sales results, selling prices, total stock of goods, and prices for complementary products. The activation function in the hidden layer uses sigmoid and in the output layer uses linear (Fig. 3). The learning process uses a scaled conjugate gradient (trainscg) with parameters epochs 5000, sigma 5e-05, lambda 5e-07, goal 0.001 and the rest is default. From the training results obtained the mean squared error (MSE) of 0.00014347 at epoch 18 (Fig. 4) and the overall R value is 0.99074 (Fig. 5).

Demand forecasting stage with the backpropagation algorithm on ANN with input data obtained from historical data from 2016 to 2019 consisting of sales data (X1), selling price (X2), product stock (X3), and complementary product prices (X4) . Data normalization was carried out using the *X/Xmax* formula (Table I).

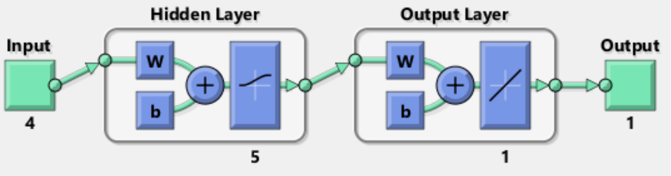


Fig. 3. Network structure.

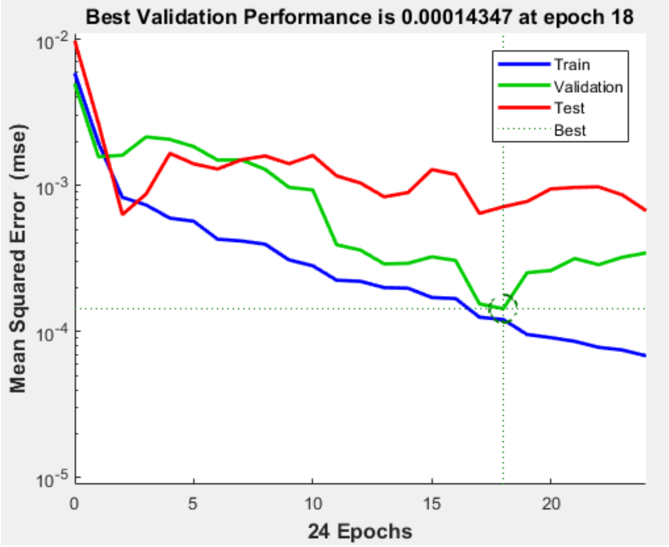


Fig. 4. MSE value.

TABLE I. DATA INPUT, TARGET, AND OUTPUT

Variable Data	
X1	... 0.63 0.64 0,67 0.60 0.64 0.75 0.67 0.64 0.67 0.70
X2	... 0.98 0.98 0.95 0.97 0.97 1.00 0.95 0.97 0.97 1.00
X3	... 0.68 0.76 0.76 0.64 0.91 0.98 0.68 0.71 0.88 0.65
X4	... 0.81 0.92 0.90 0.84 0.86 0.80 0.84 0.84 0.89 0.86
Target	... 0.63 0.64 0,67 0.60 0.64 0.75 0.67 0.64 0.67 0.70
Output	... 0.64 0.63 0.67 0.60 0.64 0.73 0.67 0.64 0.67 0.69

After the training and data testing were carried out, a simulation was carried out. This demand forecasting uses ANN backpropagation with a two-layer architecture consisting of one hidden layer with five neurons and one output layer, the sigmoid activation function (logsig) on the hidden layer and linear (purelin) on the output layer. The comparison of simulation results with actual data can be seen in Fig. 6. The comparison of simulation results and actual data shows that the simulation results by ANN backpropagation are close to the actual data, there is only a slight difference which is not too significant.

B. APP Model under Uncertainty

The simulation results of demand forecasting are used as input for the APP model. This paper sets out for a period of six months, from July to December which is completed by beginning at the end of the process and working one's way back to the beginning (backwards recursion). Stage 1 is December, stage 2 is November and beyond. Units are in tonnes and costs are in million rupiah. Production capacity (P_n) each month the same, namely 600 tons and storage capacity (W_n) each month is 900 tons. Demand data, production costs and storage costs in the TFN are shown in Table II and the fuzzy representation data in the confidence interval (α) is 0.5 in Table III.

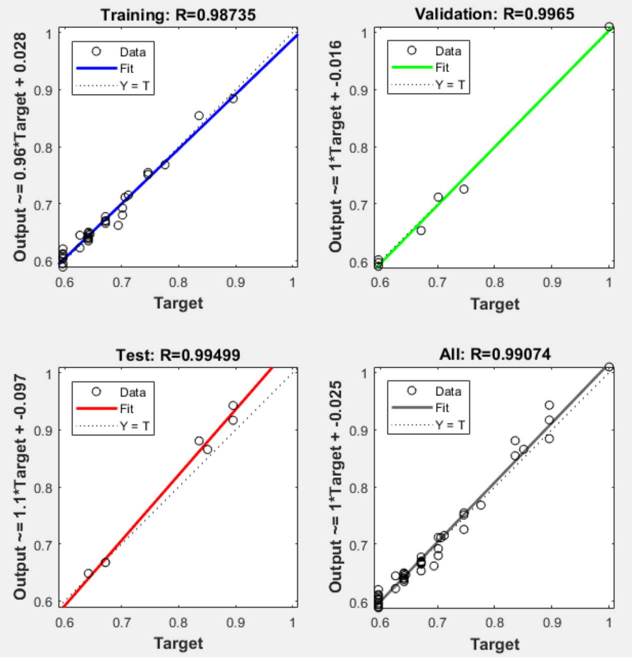


Fig. 5. R value.

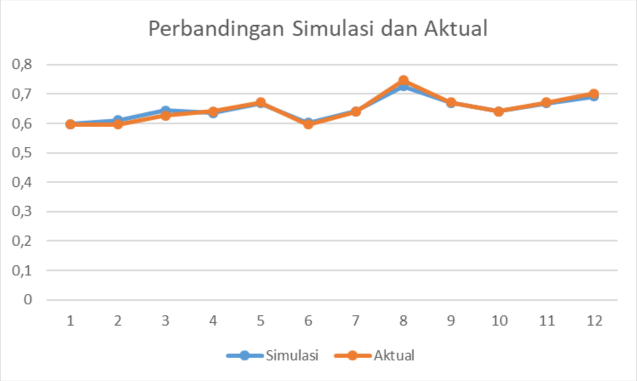


Fig. 6. Comparison of ANN backpropagation simulation results with actual data.

TABLE II. FUZZY DATA

Stage n	Fuzzy Demand $\tilde{D}_n$	Product Cost per unit $\tilde{C}_n$	Holding Cost per unit $\tilde{H}_n$
1	(400, 500, 600)	(5.1, 5.2, 5.3)	(0.077, 0.078, 0.080)
2	(300, 400, 500)	(5.3, 5.4, 5.5)	(0.080, 0.081, 0.082)
3	(300, 400, 500)	(4.9, 5.0, 5.1)	(0.074, 0.075, 0.077)
4	(300, 400, 500)	(5.0, 5.1, 5.2)	(0.075, 0.077, 0.078)
5	(400, 500, 600)	(5.4, 5.5, 5.6)	(0.081, 0.083, 0.084)
6	(300, 400, 500)	(5.0, 5.1, 5.2)	(0.075, 0.077, 0.078)

TABLE III. FUZZY DATA IN INTERVAL REPRESENTATION

Stage	**Fuzzy Demand**	**Product Cost per unit**	**Holding Cost per unit**
n	$\tilde{D}_n$	$\tilde{C}_n$	$\tilde{H}_n$
1	[450, 550]	[5.15, 5.25]	[0.077, 0.079]
2	[350, 450]	[5.35, 5.45]	[0.080, 0.082]
3	[350, 450]	[4.95, 5.05]	[0.074, 0,076]
4	[350, 450]	[5.05, 5.15]	[0.076, 0.077]
5	[450, 550]	[5.45, 5.55]	[0.082, 0.083]
6	[350, 450]	[5.05, 5.15]	[0.076, 0.077]

APP problem solving is divided into six stage according to a planning period of six months. Its objective function is to minimize total costs which include production costs and inventory storage costs during the planning period. The minimal costs at each stage are solved by equations:

$$\tilde{f}_n^*(x_n) = \min_{d_n}\{(\tilde{C}_n d_n + \tilde{H}_n(x_n + d_n - \tilde{D}_n) + \tilde{f}_{n-1}(x_{n-1})\}$$

$$\tilde{f}_1^*(x_1) = \min_{d_n}\{[5.227d_{1l} + 0.077x_{1l} - 34.76\,; 5.329d_{1u} + 0.079x_{1u} - 43.31]\}$$

$$\tilde{f}_2^*(x_2) = \min_{d_n}\{[5.43d_{2l} + 0.08x_{2l} - 28.09 + \ f_{1l}(x_{1l}); 5.532d_{2u} + 0.082x_{2u} - 36.79 + \ f_{1u}(x_{1u})]\}$$

$$\tilde{f}_3^*(x_3) = \min_{d_n}\{[5.024d_{3l} + 0.074x_{3l} - 25.99 + \ f_{2l}(x_{2l}); 5.126d_{3u} + 0.076x_{3u} - 34.09 + \ f_{2u}(x_{2u})]\}$$

$$\tilde{f}_4^*(x_4) = \min_{d_n}\{[5.126d_{4l} + 0.076x_{4l} - 26.51 + \ f_{3l}(x_{3l}); 5.227d_{4u} + 0.077x_{4u} - 34.76 + \ f_{3u}(x_{3u})]\}$$

$$\tilde{f}_5^*(x_5) = \min_{d_n}\{[5.532d_{5l} + 0.082x_{5l} - 36.79 + \ f_{4l}(x_{4l}); 5.633d_{5u} + 0.083x_{5u} - 45.79 + \ f_{4u}(x_{4u})]\}$$

$$\tilde{f}_6^*(x_6) = \min_{d_n}\{[5.126d_{6l} + 0.076x_{6l} - 26.51 + \ f_{5l}(x_{5l}); 5.227d_{6u} + 0.077x_{6u} - 34.76 + \ f_{5u}(x_{5u})]\}$$

The calculation for each stage from steps 1 to 6 is shown in Tables IV to IX.

TABLE IV. STAGE 1

x_{1l}	d_{1l}*	$f_{1l}(x_{1l})$	x_{1u}	d_{1u}*	$f_{1u}(x_{1u})$
0	450	2317.39	0	550	2887.64
50	400	2059.89	50	500	2625.14
100	350	1802.39	100	450	2362.64
150	300	1544.89	150	400	2100.14
200	250	1287.39	200	350	1837.64
250	200	1029.89	250	300	1575.14
300	150	772.39	300	250	1312.64
350	100	514.89	350	200	1050.14
400	50	257.39	400	150	787.64
450	0	-0.11	450	100	525.14
			500	50	262.64
			550	0	0.14

TABLE V. STAGE 2

x_2	d_{2l}*	$f_{2l}(x_{2l})$	x_{1l}	d_{2u}*	$f_{2u}(x_{2u})$	x_{1u}
50	300	3922.30	0	400	5067.75	0
100	250	3397.30	50	350	4532.75	50
150	200	2872.30	100	300	3997.75	100
200	150	2347.30	150	250	3462.75	150
250	100	1822.30	200	200	2927.75	200
300	50	1297.30	250	150	2392.75	250
350	0	772.30	300	100	1857.75	300
400	0	518.80	350	50	1322.75	350
450	0	265.30	400	0	787.75	400
500	0	11.80	450	0	529.35	450
550	0	15.91	450	0	270.95	500
600	0	19.91	450	0	12.55	550
650	0	23.91	450	0	16.51	550
700	0	27.91	450	0	20.61	550
750	0	31.91	450	0	24.71	550
800	0	35.91	450	0	28.81	550
850	0	39.91	450	0	32.91	550
900	0	43.91	450	0	37.01	550

TABLE VI. STAGE 3

x_3	d_{3l}*	$f_{3l}(x_{3l})$	x_{2l}	d_{3u}*	$f_{3u}(x_{3u})$	x_{2u}
50	300	5407.21	50	400	7087.86	50
100	250	4634.71	100	350	6300.36	100
150	200	3862.21	150	300	5512.86	150
200	150	3089.71	200	250	4725.36	200
250	100	2317.21	250	200	3937.86	250
300	50	1544.71	300	150	3150.36	300
350	0	772.21	350	100	2362.86	350
400	0	522.41	400	50	1575.36	400
450	0	272.61	450	0	787.86	450
500	0	22.81	500	0	533.26	500
550	0	30.62	550	0	278.66	550
600	0	38.32	600	0	24.06	600
650	0	46.02	650	0	31.82	650
700	0	53.72	700	0	39.72	700
750	0	61.42	750	0	47.62	750
800	0	69.12	800	0	55.52	800
850	0	76.82	850	0	63.42	850
900	0	84.52	900	0	71.32	900

TABLE VII. STAGE 4

x_4	d_{4l}*	$f_{4l}(x_{4l})$	x_{3l}	d_{4u}*	$f_{4u}(x_{4u})$	x_{3u}
50	300	6922.30	50	400	9147.75	50
100	250	5897.30	100	350	8102.75	100
150	200	4872.30	150	300	7057.75	150
200	150	3847.30	200	250	6012.75	200
250	100	2822.30	250	200	4967.75	250
300	50	1797.30	300	150	3922.75	300
350	0	772.30	350	100	2877.75	350
400	0	526.30	400	50	1832.75	400
450	0	280.30	450	0	787.75	450
500	0	34.30	500	0	537.00	500
550	0	45.91	550	0	286.25	550
600	0	57.41	600	0	35.50	600
650	0	68.91	650	0	47.11	650
700	0	80.41	700	0	58.86	700
750	0	91.91	750	0	70.61	750
800	0	103.41	800	0	82.36	800
850	0	114.91	850	0	94.11	850
900	0	126.41	900	0	105.86	900

TABLE VIII. STAGE 5

x_5	d_{5l}*	$f_{5l}(x_{5l})$	x_{4l}	d_{5u}*	$f_{5u}(x_{5u})$	x_{4u}
0	450	9374.91	50	550	12200.11	50
50	400	8077.41	100	500	10877.61	100
100	350	6779.91	150	450	9555.11	150
150	300	5482.41	200	400	8232.61	200
200	250	4184.91	250	350	6910.11	250
250	200	2887.41	300	300	5587.61	300
300	150	1589.91	350	250	4265.11	350
350	100	1071.41	400	200	2942.61	400
400	50	552.91	450	150	1620.11	450
450	0	34.41	500	100	1091.86	500
500	0	50.12	550	50	563.61	550
550	0	65.72	600	0	35.36	600
600	0	81.32	650	0	51.12	650
650	0	96.92	700	0	67.02	700
700	0	112.52	750	0	82.92	750
750	0	128.12	800	0	98.82	800
800	0	143.72	850	0	114.72	850
850	0	159.32	900	0	130.62	900
900	0	37.01	900	0	28.91	900

TABLE IX. STAGE 6

x_6	d_{6l}*	$f_{6l}(x_{6l})$	x_{5l}	d_{6u}*	$f_{6u}(x_{6u})$	x_{5u}
0	350	11142.5	0	450	14517.5	0

The results of calculations using fuzzy dynamic programming obtained a minimum total cost in the last stage $\tilde{f}_6^*(x_6)$ is [11142.5; 14517.5] with a mean of 12830 or twelve billion eight hundred and thirty million rupiah. Details of the amount of production and inventory per month can be seen in Table X.

TABLE X. DETAILED AMOUNT OF PRODUCTION

Month	Stage (n)	Amount of production (d_n^*)	Inventory on-hand (x_{n-1})
Jul	6	[350, 450]	0
Agt	5	[450, 550]	50
Sep	4	[300, 400]	50
Okt	3	[300, 400]	50
Nov	2	[300, 400]	0
Des	1	[450, 550]	0

IV. CONCLUSION

This paper presents an application of the use of backpropagation artificial neural network (ANN) to predict demand as input to an Aggregate Production Planning model that is compiled using a fuzzy dynamic programming (FDP) framework. Demand uncertainty, production costs, and storage costs are accommodated in the FDP framework. The prediction result of the number of requests using ANN backpropagation produces a prediction with an MSE value of 0.00014347, which means that the prediction generated by the ANN model is very close to the actual value.

The minimum total cost during the six months of the planning period calculated using the FDP framework is [11142.5; 14517.5] with a middle value of 12830 or twelve billion eight hundred and thirty million rupiah. The lowest production was in September, October and November with the amount between 300 and 400 tonnes. In July the production is between 350 and 450 tonnes. August and December with the same amount of production between 450 to 550 tons. Total product held in inventory was 50 tonnes each at the end of August, September and October.

REFERENCES

[1] J. Khalili and A. Alinezhad, "Performance Evaluation in Aggregate Production Planning Using Integrated RED-SWARA Method under Uncertain Condition," Sci. Iran., no. January, pp. 1–10, 2020.

[2] Z. Liu, Y. Zhou, G. Huang, and B. Luo, "Risk aversion sbased inexact stochastic dynamic programming approach for water resources management planning under uncertainty," Sustain., vol. 11, no. 24, 2019, doi: 10.3390/SU11246926.

[3] C. Martínez-Costa, M. Mas-Machuca, and A. Lusa, "Integration of marketing and production decisions in aggregate planning: A review and prospects," Eur. J. Ind. Eng., vol. 7, no. 6, pp. 755–776, 2013, doi: 10.1504/EJIE.2013.058395.

[4] R. C. Wang and H. H. Fang, "Aggregate production planning with multiple objectives in a fuzzy environment," Eur. J. Oper. Res., vol. 133, no. 3, pp. 521–536, 2001, doi: 10.1016/S0377-2217(00)00196-X.

[5] E. Demirel, E. C. Özelkan, and C. Lim, "Aggregate planning with Flexibility Requirements Profile," Int. J. Prod. Econ., vol. 202, pp. 45–58, 2018, doi: 10.1016/j.ijpe.2018.05.001.

[6] J. Heizer, B. Render, and C. Munson, Operations management: sustainability and supply chain management, Twelfth Ed. United States of America: Pearson Education Inc., 2017.

[7] E. Noegraheni and H. Nuradli, "Aggregate Planning to Minimize Cost of Production in Manufacturing Company," Binus Bus. Rev., vol. 7, no. 1, p. 39, 2016, doi: 10.21512/bbr.v7i1.1448.

[8] A. Jamalnia, J. B. Yang, A. Feili, D. L. Xu, and G. Jamali, "Aggregate production planning under uncertainty: a comprehensive literature survey and future research directions," Int. J. Adv. Manuf. Technol., vol. 102, no. 1–4, pp. 159–181, 2019, doi: 10.1007/s00170-018-3151-y.

[9] J. Jang and B. Do Chung, "Aggregate production planning considering implementation error: A robust optimization approach using bi-level particle swarm optimization," Comput. Ind. Eng., vol. 142, no. February 2019, p. 106367, 2020, doi: 10.1016/j.cie.2020.106367.

[10] M. Gansterer, "Aggregate planning and forecasting in make-to-order production systems," Int. J. Prod. Econ., vol. 170, pp. 521–528, 2015, doi: 10.1016/j.ijpe.2015.06.001.

[11] R. C. Wang and T. F. Liang, "Application of fuzzy multi-objective linear programming to aggregate production planning," Comput. Ind. Eng., vol. 46, no. 1, pp. 17–41, 2004, doi: 10.1016/j.cie.2003.09.009.

[12] E. B. Tirkolaee, A. Goli, and G. W. Weber, "Multi-objective aggregate production planning model considering overtime and outsourcing options under fuzzy seasonal demand," Lect. Notes Mech. Eng., pp. 81–96, 2019, doi: 10.1007/978-3-030-18789-7_8.

[13] A. Goli, E. B. Tirkolaee, B. Malmir, G. Bin Bian, and A. K. Sangaiah, "A multi-objective invasive weed optimization algorithm for robust aggregate production planning under uncertain seasonal demand," Computing, vol. 101, no. 6, pp. 499–529, 2019, doi: 10.1007/s00607-018-00692-2.

[14] B. Zhu, J. Hui, F. Zhang, and L. He, "An Interval Programming Approach for Multi-period and Multi-product Aggregate Production Planning by Considering the Decision Maker's Preference," Int. J. Fuzzy Syst., vol. 20, no. 3, pp. 1015–1026, 2018, doi: 10.1007/s40815-017-0341-y.

[15] A. Jamalnia, J. B. Yang, D. L. Xu, A. Feili, and G. Jamali, "Evaluating the performance of aggregate production planning strategies under uncertainty in soft drink industry," J. Manuf. Syst., vol. 50, pp. 146–162, 2019, doi: 10.1016/j.jmsy.2018.12.009.

[16] A. A. Demirkan and Z. D. Unutmaz Durmuşoğlu, "Evaluation of the Production Planning Policy Alternatives in a Pet Resin Production Plant: a Case Study From Turkey," Brazilian J. Oper. Prod. Manag., vol. 17, no. 2, pp. 1–24, 2020, doi: 10.14488/bjopm.2020.024.

[17] A. Cheraghalikhani, F. Khoshalhan, and H. Mokhtari, "Aggregate production planning: A literature review and future research directions," Int. J. Ind. Eng. Comput., vol. 10, no. 2, pp. 309–330, 2019, doi: 10.5267/j.ijiec.2018.6.002.

[18] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of Predictive Algorithms: Backpropagation, SVM, LSTM and Kalman Filter for Stock Market," Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019. IEEE, pp. 228–234, 2019, doi: 10.1109/AICAI.2019.8701258.

[19] A. Kochak and S. Suman, "Demand Forecasting Using Neural Network for Supply Chain Management," Int. J. Mech. Eng. Robot. Res., vol. 4, no. 1, pp. 96–104, 2015.

[20] J. Chudoung, "Iterative dynamic programming," Automatica, vol. 39, no. 7, pp. 1315–1316, 2003, doi: 10.1016/s0005-1098(03)00079-7.

[21] B. Phruksaphanrat, A. Ohsato, and P. Yenradee, "Aggregate Production Planning With Fuzzy Demand and Variable System Capacity Based on Theory of Constraints Measures," Int. J. Ind. Eng., vol. 18, no. 5, pp. 219–231, 2011.

[22] L. Li and K. K. Lai, "Fuzzy dynamic programming approach to hybrid multiobjective multistage decision-making problems," Fuzzy Sets Syst., vol. 117, pp. 13–25, 2001, doi: 10.1016/S0165-0114(98)00423-0.

[23] W. B. Powell, "Perspectives of approximate dynamic programming," Ann. Oper. Res., vol. 241, no. 1–2, pp. 319–356, 2016, doi: 10.1007/s10479-012-1077-6.

[24] A. Ishak and P. Nababan, "The fuzzy goal programming approach to production planning of intermediate gear spare parts: a case study," J. Sist. dan Manaj. Ind., vol. 4, no. 2, pp. 137–143, 2020, doi: 10.30656/jsmi.v4i2.2143.

[25] R. E. Bellman and L. A. Zadeh, "Decision-Making in a Fuzzy Environment," Manage. Sci., vol. 17, no. 4, pp. B141–B164, 1970.

[26] S. Mohanaselvi and S. Suparna Mondal, "A Fuzzy Dynamic Programming Approach to Fuzzy Least Cost Route Problem," J. Phys. Conf. Ser., vol. 1377, no. 1, 2019, doi: 10.1088/1742-6596/1377/1/012042.

[27] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," Int. J. Forecast., vol. 22, pp. 443–473, 2006, doi: 10.1016/j.ijforecast.2006.01.001.

[28] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks : The state of the art," Int. J. Forecast., vol. 14, pp. 35–62, 1998.

[29] A. M. Rather, A. Agarwal, and V. N. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," Expert Syst. Appl., vol. 42, no. 6, pp. 3234–3241, Apr. 2015, doi: 10.1016/j.eswa.2014.12.003.

[30] S. S. Fayaed et al., "Improving dam and reservoir operation rules using stochastic dynamic programming and artificial neural network integration model," Sustain., vol. 11, no. 19, 2019, doi: 10.3390/su11195367.

[31] H. R. Yazgan, "Selection of dispatching rules with fuzzy ANP approach," Int. J. Adv. Manuf. Technol., vol. 52, no. 5–8, pp. 651–667, 2011, doi: 10.1007/s00170-010-2739-7.

[32] A. Eshragh, "Surprise Maximization: A Dynamic Programming Approach," pp. 1–5, 2020, [Online]. Available: http://arxiv.org/abs/2012.14933.

[33] A. O. Esogbue and J. Kacprzyk, "Fuzzy Dynamic Programming," in Fuzzy Sets in Decision Analysis, Operations Research and Statistics, The Handbo., R. Słowiński, Ed. Boston, MA.: Springer, 1998, pp. 281–307.

[34] A. Rathke et al., "Dynamic Pricing Using Thompson Sampling with Fuzzy Events," Commun. Comput. Inf. Sci., vol. 1237 CCIS, no. 1, pp. 653–666, 2020, doi: 10.1007/978-3-030-50146-4_48.

[35] M. Marimin et al., Teknik dan Analisis Pengambilan Keputusan Fuzzy Dalam Manajemen Rantai Pasok, no. April. 2013.

A Predictive Approach to Improving Agricultural Productivity in Morocco through Crop Recommendations

Rachid Ed-daoudi[1], Altaf Alaoui[2], Badia Ettaki[3], Jamal Zerouaoui[4]

Laboratory of Engineering Sciences and Modeling-Faculty of Sciences, Ibn Tofail University, Campus Universitaire, BP 133, Kenitra, Morocco[1, 2, 3, 4]

LyRICA: Laboratory of Research in Computer Science-Data Sciences and Knowledge Engineering, School of Information Sciences, Rabat, Morocco[3]

***Abstract*—Agricultural productivity is a critical component of sustainable economic growth, particularly in developing countries. Morocco, with its vast agricultural potential, is in need of advanced technologies to optimize crop productivity. Precision farming is one such technology, which incorporates the use of artificial intelligence and machine learning to analyze data from various sources and make informed decisions about crop management. In this study, we propose a web-based crop recommendation system that leverages ML algorithms to predict the most suitable crop to harvest based on environmental factors such as soil nutrient levels, temperature, and precipitations. We evaluated the performance of five ML algorithms (Decision Tree, Naïve Bayes, Random Forest, Logistic Regression, and Support Vector Machine) and identified Random Forest as the best-performing algorithm. Despite the promising results, we faced several challenges, including limited availability of data and the need for field validation of the results. Nonetheless, our platform aims to provide free and open-source precision farming solutions to Moroccan farmers to improve agricultural productivity and contribute to sustainable economic growth in the country.**

Keywords—Precision agriculture; artificial intelligence; machine learning; crop recommendation; Morocco

I. Introduction

Agriculture is a crucial sector in Morocco, accounting for a significant portion of the country's GDP and employing a large percentage of the population. However, the sector faces several challenges, including water scarcity, unpredictable weather patterns, and a lack of access to information and resources [1]. Recently, an increasing interest has been observed for developing crop recommendation systems to address these challenges and improve agricultural productivity in the country [2].

Crop recommendation systems use predictive techniques to suggest the most suitable crops for a given location and set of conditions. These systems rely on data from various sources, such as satellite imagery, weather forecasts, soil analyses, and historical crop yields, to make accurate predictions [3]. By providing farmers with personalized recommendations, crop recommendation systems can help them make informed decisions about crop selection and improve their yields and profits [4].

The objective of this paper is to present a predictive approach to improving agricultural productivity in Morocco through crop recommendations. Specifically, we will develop a crop recommendation system that utilizes prediction techniques to suggest the most suitable crops for a given location and set of conditions.

In order to achieve this objective, we will initially examine the current body of literature related to crop recommendation systems as well as the difficulties encountered by the agricultural sector in Morocco. After this, we will outline the approach we employed to construct our system, which including the dataset, prediction techniques, and evaluation metrics. Following this, we will detail the outcomes of our study, including the precision of our forecasts and the efficacy of our crop recommendation system. Lastly, we will analyze the implications of our research on enhancing agricultural productivity in Morocco and suggest potential avenues for future research.

The structure of the paper is as follows. Firstly, an introduction is provided, followed by a succinct AI in precision agriculture summary, focusing primarily on Machine Learning (ML) as a component of Artificial Intelligence (AI). The paper also includes a concise literature review that highlights recent developments in the field. The research methodology is then presented, which involves an experimental evaluation of our research using various ML algorithms such as Decision Tree, Naïve Bayes, Random Forest, Logistic Regression, and Support Vector Machine. A discussion comes after this section. Finally, the key findings of the study are summed up as a conclusion of this paper.

II. Precision Farming Overview

Precision farming is an agricultural management approach that uses technology to optimize crop yields and reduce waste. Its objective is to provide farmers with real-time data and information on their farms and livestock, in order to take accurate actions that can result in maximized crop yields and limited losses.

AI helps in analyzing data from various sources and making informed decisions about crop management [5]. ML, a subfield of AI, has proven to be particularly useful in precision farming due to its ability to automatically learn patterns from large datasets and make predictions based on those patterns [6]. Fig. 1 presents the key component of precision farming.

Precision farming involves the use of a range of IoT sensors that gather various indicators associated with farming [7]. These indicators can include various environmental parameters such as soil moisture, nutrient levels, pH, temperature, humidity, light, and weather conditions. Other indicators may also relate to livestock, such as veterinary well-being, feed intake, and weight gain. These indicators can be used to optimize various aspects of crop and livestock management, including irrigation, fertilization, disease prevention, and pest control [8].

The data collected by these sensors is transmitted to the technical staff. Various data analytic methods are then applied to interpret and derive useful insights from the collected data. The resulting information is exploited to take accurate and timely actions. To provide a better understanding, Fig. 2 illustrates the various stages involved, including the automated collection of data and the processing operations by agribots according to the insights derived from the processed data.

One area where ML has been applied in precision farming is crop yield prediction. ML algorithms can analyze data from sensors, such as satellite imagery, weather stations, and soil sensors, to make accurate predictions about crop yields [9]. For example, a study [10] used ML to predict winter wheat yields in China based on satellite imagery and weather data.

Another application of ML in precision farming is crop disease detection. By analyzing images of crops, ML algorithms can identify signs of disease or stress and alert farmers to take action [11]. For example, a study [12] used ML to identify cassava disease in Tanzania based on images captured by drones. The authors found that their ML model was able to accurately identify diseased plants, even in cases where the symptoms were not visible to the human eye.

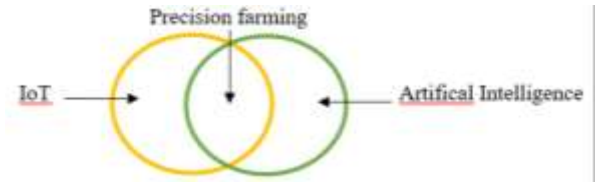


Fig. 1. Core components of precision agriculture.

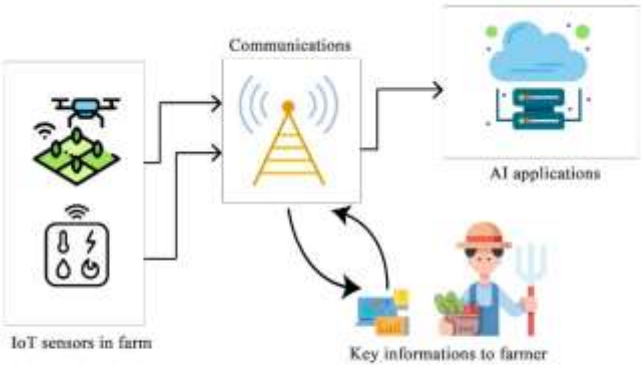


Fig. 2. Overview of the general process involved in precision agriculture.

ML can also be used to optimize crop management decisions, such as irrigation and fertilization. By analyzing data on soil moisture, nutrient levels, and weather patterns, ML algorithms can suggest the optimal amount of water to apply to a crop [13]. For example, a study by [14] demonstrates that the integration of active canopy sensing and machine learning improves the prediction accuracy of the corn nitrogen nutrition index by accounting for genetic, environmental, and management factors.

In conclusion, ML has significant potential in precision farming due to its ability to analyze large datasets and make accurate predictions. By applying ML to tasks such as crop yield prediction, disease detection, and crop management optimization, farmers can improve their yields and reduce waste.

III. Literature Review

Nowadays, as ML has been increasingly employed in precision agriculture, numerous studies have been conducted across various areas of farming. Therefore, to provide a clearer picture and distinguish this work from others, we present a summary of recent research in Table I.

TABLE I. Overview of the Latest Research and their Significant Contributions

Reference	Field	Techniques Used	Results
[15]	Crop yield prediction	Machine learning	Improved accuracy in crop yield prediction with the use of multiple data sources and ML
[16]	Disease detection	Transfer learning, Convolutional neural networks	Accurate detection of grapes and tomatoes leaf diseases using transfer learning and CNNs
[17]	Crop manageme nt	Machine learning	Accurate prediction of crop yield and nutrient deficiency using aerial images and ML
[18]	Soil mapping	Machine learning	Accurate mapping of soil properties using ML and sensor data
[19]	Pest detection	Deep convolutional neural networks	Accurate detection of tomato pests using deep learning and image analysis
[20]	Irrigation optimizati on	Reinforcement learning	Efficient water use and yield improvement in precision irrigation using reinforcement learning
[21]	Crop growth modeling	Deep learning	Accurate modeling of crop growth and development using deep learning with satellite and climate data
[22]	Crop yield prediction	Convolutional neural networks, gradient boosting	Accurate wheat yield prediction UAV-based multi-sensor data fusion approach and a machine learning algorithm
[23]	Nitrogen manageme nt	Machine learning algorithms	Accurate estimation of plant nitrogen status using machine learning and hyperspectral imaging
[24]	Crop health monitoring	Deep learning	Accurate monitoring of banana health using aerial images and deep learning

[25]	Seed selection	Deep learning, machine vision	Efficient maize kernel selection using deep learning and machine vision techniques
[26]	Crop yield prediction	Convolutional neural networks	Accurate wheat yield prediction using UAV images and ML algorithms
[27]	Weed detection	Convolutional neural networks	Accurate detection of weeds in rice using deep learning and image analysis
[28]	Crop manageme nt	Random forests, GIS approaches	Accurate prediction of nutrient deficiency using aerial images and ML
[29]	Disease detection	Convolutional neural networks	Accurate detection of potato diseases using deep learning and image analysis

Table I displays that numerous scientists have created Artificial Intelligence AI-based solutions suitable for different agriculture areas including crop selection and cultivation techniques. These solutions are often presented in the form of predictive systems and recommendation platforms, using state-of-the-art techniques. While these approaches have potential, they do not necessarily address the need for real solutions that can be implemented in the field at low cost and can help farmers and decision-makers optimize their resources and maximize profits.

It is interesting to note that few studies have focused on this issue in recent years, with most studies providing only theoretical overviews and practical implementations. The challenge is to find a way to make these solutions available for all, and make them more visible to a larger audience, such as farmers. Our study provides a step-by-step explanation of how to address these problems and offers examples of how to make these technologies available for free.

IV. Methods and Results

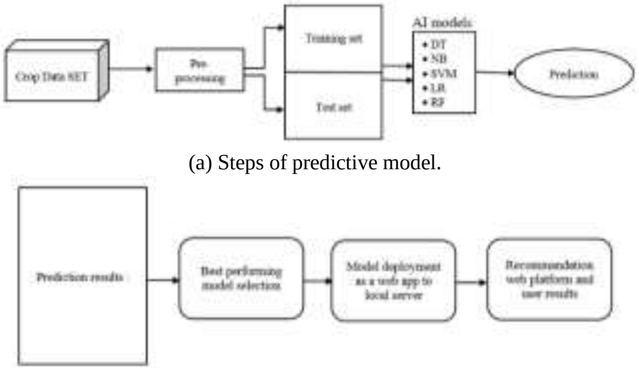
A. System Conception

To design our crop recommendation system, we initiated the first step by preparing a crop recommendation dataset. We used Moroccan climate, fertilizer and precipitations data [30] then preprocessed it in order to make extensive explorations. We worked with a dataset consisting of 1800 entries and eight variables. Next, we conducted an exploration of the data to understand its nature. Finally, we used ML techniques counting NB, DT, RF, SVM, and LR to extract the best features and build our ML models.

Fig. 3 provides a detailed illustration of the steps taken in designing the architecture and developing the crop recommendation system.

After completing the models building step, we proceeded to evaluate their performance. Once the evaluation was complete, we then moved on to deploy the model with highest performance as a web application, which forms a basis of our crop recommendation system.

After completing the evaluation, the next step was to deploy the best selected model as a web application, which serves as the foundation for our crop recommendation system.


Crop Data SET
Pre-processing
Training set
Test set
AI models
• DT
• NB
• SVM
• LR
• RF
Prediction
Prediction results
Best performing model selection
Model deployment as a web app to local server
Recommendation web platform and user results

(a) Steps of predictive model.

(b) Steps of recommendation system building.

Fig. 3. Design process of the crop recommendation system.

B. Dataset Preparation and Feature Selection

The crop recommendation dataset we selected comprised several features, including content in soil of Phosphorus (P), Potassium (K) and Nitrogen (N), as well as soil pH value and moisture, precipitations, temperature and plant variety. The data collection consisted of 1800 entries, and an overview of the dataset statistics is presented in Table II.

In terms of the relationship between the input parameters and target, we observed that certain parameters, such as Potassium, phosphorus and Nitrogen exhibit strong correlations of 0.6. This is a promising sign prior to the commencement of the prediction. There were also some phew negative correlation values between the indicators. This shows that the dataset is rich in information.

From a biological perspective, the P K N levels play a vital role as important macro-nutrients that support plant growth. Potassium, is responsible for the plant's natural functioning, while phosphorus aids in the maturation of fruits. Nitrogen contributes mainly to the growth of leaves.

Fertilizer containing elements such as P K N and can be tailored to different crops based on their concentrations. These nutrients are essential for the growth and productivity of crops, and using fertilizer can make up for any deficiencies in the soil. By understanding the PKN ratios required for optimal plant development, farmers can manage fertilization and achieve better yields [31].

Other parameters in our data, including temperature, soil moisture and pH, and precipitations play important roles in crop selection. The pH value affects the existence of vital substances, while precipitations level is crucial in plant survival. Air temperature impacts photosynthesis, and soil moisture is essential for growth, but too much or too little can be detrimental. Our dataset includes several crop categories, such as apples, dates, corn, grape, pepper, orange, peach, potatoes, onions, tomatoes, olives, and watermelon, that can be predicted based on these features. The prediction of these crops yields is dependent on the parameters mentioned.

TABLE II. STATISTICAL DESCRIPTION OF THE DATASET

Parameters	P	K	N	Temperature	Soil Moisture	pH	Precipitations
Records	1800	1800	1800	1800	1800	1800	1800
Mean	53.36	48.14	50.55	25.61	71.48	6.47	103.50
Standard Deviation	32.98	150.64	36.9	5.06	22.26	0.77	54.95
Minimum	5.00	5.00	0.00	8.82	14.25	3.50	20.21
Maximum	145.00	205.00	140.00	43.67	99.98	9.93	298.56

C. Predictive Models Development

To develop our predictive ML models, we utilized the Python programming language. The data preprocessing stage involved importing the necessary libraries, including NumPy, Pandas, Scikit-learn, and Matplotlib.

To train and test the models, we split the dataset into training and testing datasets, with a ratio of 70:30. We then employed five ML models (DT, NB, LR, RF, and SVM).

To evaluate the accuracy of the models, we used precision, recall, F1 and accuracy metrics, which were calculated using the following formulas:

$$Precision \ = \frac{TP}{TP+FP} \quad (1)$$

$$Recall \ = \frac{TP}{TP+FN} \quad (2)$$

$$Accuracy \ = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$F1\ score \ = \ 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

Where TP (true positive) is the number of correctly predicted positive values, FP (false positive) is the number of falsely predicted positive values, TN (true negative) is the number of correctly predicted negative values, and FN (false negative) is the number of falsely predicted negative values.

Additionally, we used k-fold cross-validation scores to evaluate the effectiveness of the different algorithms used in the predictive analysis, which provided a more robust assessment of the models' performance [32].

These scores helped us evaluate the effectiveness of the different algorithms used in the predictive analysis.

V. RESULTS AND DISCUSSION

A. Results of Prediction

This section presents a discussion of the experimental results obtained for the performance of the ML algorithms used in our study. Table III provides a detailed summary of the precision, recall, accuracy, and k-fold cross-validation score results.

Based on our results, the ML algorithms employed in our study exhibit varying levels of predictive capabilities in determining profitable crop based on inputs. Our analysis shows that RF has the highest performance in terms of accuracy, with a score of 97.18%. In comparison, other algorithms such as NB (96.36%), LR (95.62%), SVM (87.38%), and DT (86.64%) have lower accuracy scores.

Fig. 4 illustrates a comparison of the accuracy for all ML models adopted in our analysis.

The precision value assesses the ratio of accurate positive label predictions, indicates that NB and RF exhibit excellent performance. Conversely, DT has the lowest performance in this category. Regarding recall value, which assesses the ability of the ML models to correctly predict actual positives, RF once again outperforms the other models, while SVM and DT have the lowest scores.

Regarding the F1 score, which is the harmonic mean of precision and recall, RF attains the highest score of 97%, followed by NB at 96%, LR at 96%, SVM at 87%, and DT at 83%.

To estimate the overall performance of the models, we used K-fold cross-validation with a value of K=10. The results indicate that RF has the highest 10-fold cross-validation vale, while DT has the lowest performance. Although accuracy alone is not a reliable performance metric, considering other evaluation measures, it is apparent that RF surpasses the used ML techniques in forecasting the most appropriate crop.

TABLE III. PERFORMANCE METRICS SCORES FOR THE ML MODELS

Algorithm	Accuracy	F1	Recall	Precision	10-Fold Cross Validation
SVM	87.38%	87%	87%	87%	88.50%
NB	96.36%	96%	96%	97%	97%
RF	97.18%	97%	97%	97%	97.40%
LR	95.62%	96%	96%	96%	96.31%
DT	86.64%	83%	87%	82%	92%

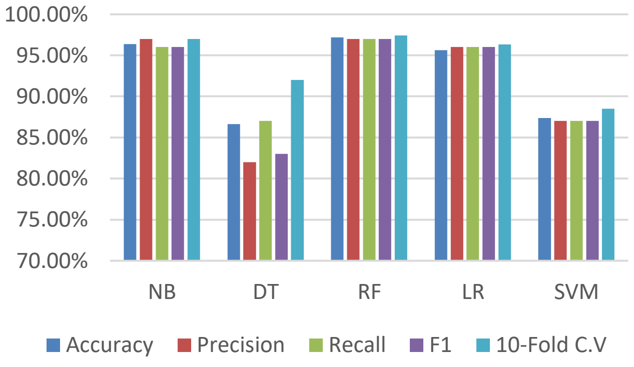


Fig. 4. Results comparison of the ML algorithms.

B. Crop Recommendation Solution

In our opinion, the market for precision agriculture solutions that are available at no cost and released under an open-source license is continuing to evolve, and considerable progress is needed for these solutions to compete with closed-source solutions. However, the involvement of a network of IT specialists and interested parties, including farmers, has aided expansion of open-source alternatives. These communities can address any issues or doubts related to technology integration with ease [33-35]. Bearing this in mind, our main objective was to propose and design a crop recommendation system.

Relying on the performance metrics mentioned earlier, RF demonstrated superior performance in comparison to the other ML models we used. Therefore, we plan to employ RF to predict the optimal crop based on the user's input parameters, such as P K N, soil moisture and pH, temperature, and precipitations. After selecting the best-performing model, we serialized/saved it Python’s built-in persistence model, namely pickle, for developing of the crop recommendation system. The various steps involved in designing and deploying the crop recommendation system are outlined in Fig. 5.

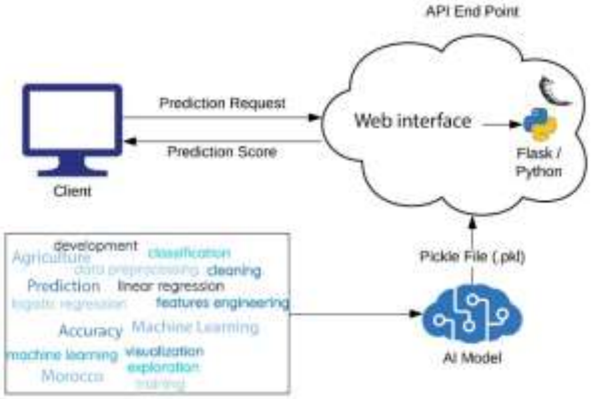


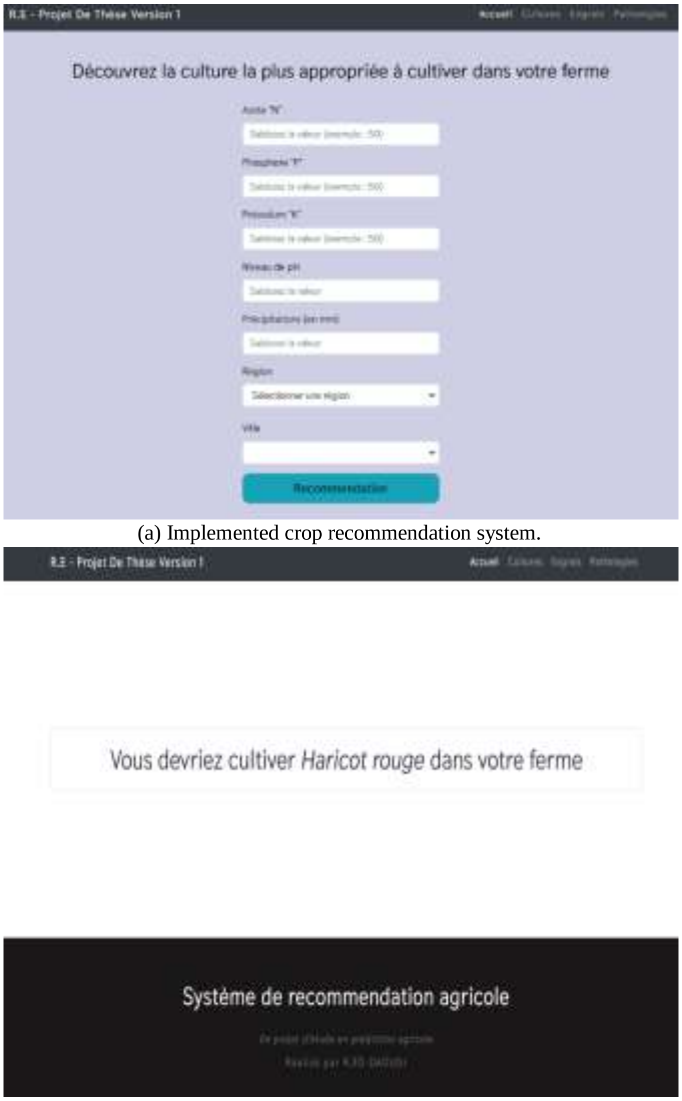
Fig. 5. Stages included in creating and launching the crop suggestion platform.

To create the system, we utilized Flask as a Python micro framework used in applications development, which enables programmers to conveniently build Application Programming Interfaces (APIs) in Python language. To create interactive web pages for users, we used HTML (Hypertext Markup Language) and designed them using CSS. After that, we implemented the platform Flask micro web framework. Fig. 6 depicts the fully functional recommendation platform hosted locally.

The working system along with the development environment are available on an uploaded directory online [36]. It consists of all code files used for the training and prediction, as well as the app code of the recommendation system and the frontend source code.

In Fig. 6(a), the collected field data is entered manually at the Front end.

In Fig. 6(b), the Result is displayed after processing the given data through the trained M.L Model.



Découvrez la culture la plus appropriée à cultiver dans votre ferme

(a) Implemented crop recommendation system.

Vous devriez cultiver *Haricot rouge* dans votre ferme

Systéme de recommendation agricole

(b) Implemented crop recommendation system.

Fig. 6. Implemented crop recommendation system.

VI. Discussion

After evaluating the performance results of the ML model training, we found that RF has the best performance among all the models. Despite the small training dataset of only 1800 entries, RF still outperforms other models in terms of recall, precision, and other performance criteria, apart from accuracy. Therefore, we chose RF as the ML technique to use in building the recommendation system.

Once the required inputs are submitted, the platform verifies for missing values and if the entries fall within acceptable thresholds before proceeding. The system then predicts the most appropriate crop to plant based on the input parameters. This enables farmers to make informed decisions for higher returns on investment and reduced wastage. This enables farmers to opt for knowledgeable choices that lead to increased profits and reduced waste.

During the development of our crop recommendation system, we postulated that users would utilize pre-existing meteorological data and the abundant IoT farming tools that are already deployed to obtain necessary information for submitting parameters to our platform. Since our platform is web-based, it is accessible from anywhere and from any device, providing a high level of convenience for the users.

However, we acknowledge that the dataset we used for developing the system is constituted of Moroccan data, and crop growth can vary based on the context and circumstances of different territories. To provide our system to farmers globally, more data from various geographic locations may be required. As our system continues to develop, it can have the capability to compile information from various geographical territories, allowing users to enter their position along with the relevant data to predict suitable crops for planting. This paper displays the potential of using ML to design smart techniques at the service of agriculture, and it may set the stage for emerging studies to develop predictive platforms in conjunction with robotic technology.

Investing in farming is critical for any economy, and selecting profitable crops is an important task that can guarantee the best production. Our proposed solution aims to help farmers select the best crop for their land and environmental context. With further improvements, like integrating IoT in automated data collection and using more parameters and key indicators from different locations, we can offer this technology to more farmers, especially those who want to implement technology-driven precision farming.

While many organizations and startups are designing precision farming solutions, most of them are subscription-based with high prices, free and open-source alternatives can be accessible for farmers to. It is crucial for farmers to be knowledgeable about these alternatives that can enable them to utilize technology at no cost, instead of making significant upfront investments.

VII. Conclusion and Perspectives

Agriculture is a vital industry that feeds billions of people worldwide, and the use of technology has transformed traditional farming practices, resulting in improved yields and a higher quality harvest while reducing manual labor. Precision farming, powered by AI, is becoming increasingly popular due to its ability to collect field data and to assist farmers in taking informed actions to manage their cultivated lands.

This study presents a web-based ML-driven crop recommendation system to assist users choose the most suitable crops to plant based on local environmental conditions. Although there are many commercial solutions available, alternatives also exist at freely and at no coast.

The study demonstrates the combination of ML and precision agriculture and outlines the architecture of a web-based platform. The platform could be improved by adding real-time monitoring capabilities by interfacing with existing data aggregation platforms from sensors and IoT. This would result in a more accurate and suitable prediction framework, making precision farming more accessible to farmers worldwide, ultimately helping to address the challenges of feeding the global population.

References

[1] Laamari, A., Boughlala, M., Herzenni, A., Karrou, M., & Bahri, A. Water policies in Morocco–Current situation and future perspectives. Improving water and land productivities in rainfed systems. Community-Based Optimization of the Management of Scarce Water Resources in Agriculture in CWANA, 2011, 8, 103.

[2] El Hachimi, C., Belaqziz, S., Khabba, S., & Chehbouni, A. Towards precision agriculture in Morocco: A machine learning approach for recommending crops and forecasting weather. 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), 2021, 88-95.

[3] Sartore, L., Rosales, A. N., Johnson, D. M., & Spiegelman, C. H. Assessing machine learning algorithms on crop yield forecasts using functional covariates derived from remotely sensed data. Computers and Electronics in Agriculture, 2022, 194, 106704.

[4] Pande, S. M., Ramesh, P. K., Anmol, A., Aishwarya, B. R., Rohilla, K., & Shaurya, K. Crop Recommender System Using Machine Learning Approach. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1066-1071. IEEE.

[5] Gebbers, R., & Adamchuk, V. I. Precision agriculture and food security. Science, 2010, 327(5967), 828-831.

[6] Shahi, T.B., Xu, C.-Y., Neupane, A., & Guo, W. Machine learning methods for precision agriculture with UAV imagery: a review. Electronic Research Archive, 2022, 30(12), 4277-4317.

[7] Rokade, A., Singh, M., Malik, P. K., Singh, R., & Alsuwian, T. Intelligent Data Analytics Framework for Precision Farming Using IoT and Regressor Machine Learning Algorithms. Applied Sciences, 2022, 12(19), 9992.

[8] Vuran, M. C., Salam, A., Wong, R., & Irmak, S. Internet of underground things in precision agriculture: Architecture and technology aspects. Ad Hoc Networks, 2018, 81, 160-173.

[9] Benos, L., Tagarakis, A. C., Dolias, G., Berruto, R., Kateris, D., & Bochtis, D. Machine Learning in Agriculture: A Comprehensive Updated Review. Sensors, 2021, 21(11), 3758.

[10] Cao, J., Zhang, Z., Tao, F., Li, Z., Li, W., Xu, X., & Zhang, Y. Identifying the Contributions of Multi-Source Data for Winter Wheat Yield Prediction in China. Remote Sensing, 2020, 12(5), 750.

[11] Zhang, N., Yang, G., Pan, Y., Yang, X., Chen, L., & Zhao, C. A Review of Advanced Technologies and Development for Hyperspectral-Based Plant Disease Detection in the Past Three Decades. Remote Sensing, 2020, 12(19),3188.

[12] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. Deep learning for image-based cassava disease detection. Frontiers in Plant Science, 2017, 8, 1852.

[13] Chen, Y.-A., Hsieh, W.-H., Ko, Y.-S., & Huang, N.-F. An Ensemble Learning Model for Agricultural Irrigation Prediction. In 2021 International Conference on Information Networking (ICOIN), 2021, pp. 311-316. IEEE.

[14] Li, D., Miao, Y., Ransom, C.J., Bean, G.M., Kitchen, N.R., Fernández, F.G., Sawyer, J.E., Camberato, J.J., Carter, P.R., Ferguson, R.B., Franzen, D.W., Laboski, C.A.M., Nafziger, E.D., & Shanahan, J.F. Corn nitrogen nutrition index prediction improved by integrating genetic, environmental, and management factors with active canopy sensing using machine learning. Remote Sensing, 2022, 14(2), 394.

[15] Cedric, L. S., Adoni, W. Y. H., Aworka, R., Zoueu, J. T., Mutombo, F. K., Krichen, M., & Kimpolo, C. L. M. Crops yield prediction based on machine learning models: Case of West African countries. Smart Agricultural Technology, 2022, 2, 100049.

[16] Paymode, A. S., & Malode, V. B. Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG. Artificial Intelligence in Agriculture, 2022, 6, 23-33.

[17] Wang, S., Guan, K., Wang, Z., Ainsworth, E. A., Zheng, T., Townsend, P. A., Liu, N., Nafziger, E., Masters, M. D., Li, K., Wu, G., & Jiang, C. Airborne hyperspectral imaging of nitrogen deficiency on crop traits and yield of maize by machine learning and radiative transfer modeling. International Journal of Applied Earth Observation and Geoinformation, 2021, 105, 102617.

[18] Forkuor, G., Hounkpatin, O.K.L., Welp, G., & Thiel, M. High resolution mapping of soil properties using remote sensing variables in South-Western Burkina Faso: A comparison of machine learning and multiple linear regression models. PloS one, 2017, 12(1), e0170478.

[19] Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. Sensors 2017, 17, 2022.

[20] L. Sun, Y. Yang, J. Hu, D. Porter, T. Marek and C. Hillyer. Reinforcement Learning Control for Water-Efficient Agricultural Irrigation. IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, China, 2017, pp. 1334-1341

[21] Cai, Y., Guan, K., Lobell, D., Potgieter, A. B., Wang, S., Peng, J., Xu, T., Asseng, S., Zhang, Y., You, L., & Peng, B. Integrating satellite and climate data to predict wheat yield in Australia using machine learning approaches. Agricultural and Forest Meteorology, 2019, 274, 144-159.

[22] Fei, S., Hassan, M.A., Xiao, Y. et al. UAV-based multi-sensor data fusion and machine learning algorithm for yield prediction in wheat. Precision Agric, 2023, 24, 187–212.

[23] Jiang, J., Atkinson, P. M., Zhang, J., Lu, R., Zhou, Y., Cao, Q., Tian, Y., Zhu, Y., Cao, W., & Liu, X. Combining fixed-wing UAV multispectral imagery and machine learning to diagnose winter wheat nitrogen status at the farm scale. European Journal of Agronomy, 2022, 138, 126537.

[24] Selvaraj, M. G., Vergara, A., Montenegro, F., Ruiz, H. A., Safari, N., Raymaekers, D., Ocimati, W., Ntamwira, J., Tits, L., Omondi, A. B., & Blomme, G. Detection of banana plants and their major diseases through aerial images and machine learning methods: A case study in DR Congo and Republic of Benin. ISPRS Journal of Photogrammetry and Remote Sensing, 2020, 169, 110-124.

[25] Xu P, Tan Q, Zhang Y, Zha X, Yang S, Yang R. Research on Maize Seed Classification and Recognition Based on Machine Vision and Deep Learning. Agriculture, 2022, 12(2):232.

[26] Tanabe, R., Matsui, T., & Tanaka, T. S. Winter wheat yield prediction using convolutional neural networks and UAV-based multispectral imagery. Field Crops Research, 2023, 291, 108786.

[27] Huang, H., Lan, Y., Yang, A., Zhang, Y., Wen, S., & Deng, J. Deep learning versus Object-based Image Analysis (OBIA) in weed mapping of UAV imagery. International Journal of Remote Sensing, 2020, 41(9), 3446-3479.

[28] Gasmi A, Gomez C, Chehbouni A, Dhiba D, El Gharous M. Using PRISMA Hyperspectral Satellite Imagery and GIS Approaches for Soil Fertility Mapping (FertiMap) in Northern Morocco. Remote Sensing, 2022, 14(16):4080.

[29] Arshaghi A, Ashourian M, Ghabeli L. Potato diseases detection and classification using deep learning methods. Multimedia Tools and Applications, 2022, 30:1-8.

[30] FAOSTAT. Food and Agriculture Data. Available from: http://www.fao.org/faostat/en/#home. Accessed January 2023.

[31] P. Krasilnikov, M.A. Taboada, and Amanullah, "Fertilizer Use, Soil Health and Agricultural Sustainability," Agriculture, 2022, vol. 12, no. 4, p. 462.

[32] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management, 2009, 45(4):427–437.

[33] A. Puspaningrum, A. Sumarudin and W. P. Putra, "Irrigation Prediction using Machine Learning in Precision Agriculture," 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 2022, pp. 204-208.

[34] Gebbers, R., & Adamchuk, V. I. Precision agriculture and food security. Science (New York, N.Y.), 2010, 327(5967), 828–831.

[35] Abu, N., Bukhari, W., Ong, C., Kassim, A., Izzuddin, T., Sukhaimie, M., Norasikin, M., & Rasid, A. Internet of Things Applications in Precision Agriculture: A Review. Journal of Robotics and Control (JRC), 2022, 3(3), 338-347.

[36] The directory for the working system and development environment: https://drive.google.com/drive/folders/1eBtx4g4VGOg4VHWlMyZjnPExNCGUjav3?usp=share_link. Accessed January 2023.

The Research on the Motion Control of the Sorting Manipulator based on Machine Vision

Kuandong Peng[1], Zufeng Wang[2]
School of Intelligent Manufacturing, Hangzhou Polytechnic, Hangzhou, Zhejiang 311402, China[1]
Information Technology Center, Zhejiang University City College, Hangzhou, Zhejiang 310015, China[2]

***Abstract*—With the development of production technology, manipulators are gradually introduced in advanced production manufacturing industries to complete some tasks such as picking and sorting. However, the traditional manipulator has a complicated sorting process and low production efficiency. In order to improve the accuracy of sorting and reduce the labor intensity of workers, this paper studied the motion control of the sorting manipulator with machine vision. After placing four kinds of objects of different shapes on the conveyor belt, experiments were conducted on the catching and sorting process of the manipulator under different experimental environments, different conveyor belt speeds, and with or without machine vision. It was found that the overall success rate of the sorting robotic arm using machine vision for catching objects of different shapes was as high as 96%, and the sorting accuracy was as high as 97.91%. Therefore, it is concluded that the manipulator can achieve high accuracy in catching and sorting objects with the guidance of machine vision, and the adoption of machine vision has a positive impact on the motion control of the sorting manipulator.**

***Keywords*—Machine vision; manipulator; motion control; camera calibration; item sorting**

I. Introduction

In today’s rapid development of the intelligent technology industry, industrial automation has become the development direction of the production manufacturing industry, and manipulators have been widely used in industrial operations [1], for example, manipulator sorting technology. Traditional robot sorting is mainly carried out by means of demonstration and can only work in a fixed environment, and the manipulator does not work efficiently because it cannot recognize and process objects without a vision system. The manipulator using machine vision system can take camera shots of the work site to collect images and obtain information such as the location and size of the object to realize catching and sorting. Therefore, the combination of machine vision and manipulators allows the arms to identify objects independently, which has important practical significance for improving efficiency and reducing labor costs. Radcliffe et al. found that the use of machine vision allowed small vehicle platform systems to navigate autonomously and reduce errors through laboratory field tests [2]. Min et al. found that a portable machine-based visual inspection system for track defects could replace manual labor to some extent after field experiments [3]. Liu proposed that robots equipped with machine vision-based manipulators could efficiently perform tasks such as logistics courier sorting and fruit picking in orchards [4]. Abad et al. found through field experiments that machine vision was able to achieve at least 95% accuracy in color discrimination of stacked colored objects [5]. Nair et al. found that the application of machine vision for image analysis helped in classifying flood zones and had an accuracy of 83.1% [6]. Mohamed et al. applied a real-time machine vision manipulator through Python software and found that the arm was very accurate in detecting external defects in agricultural products [7]. He et al. verified that a machine vision-based sorting method was feasible by obtaining the contours, features and dimensions of mechanical parts with a camera and filtering out defective products through image analysis [8]. The article first described the machine vision of the sorting manipulator through camera calibration and image processing methods; then, the inverse kinematic equation in the D-H parameters was used to solve different joint angles of the manipulator to achieve motion control. Finally, after placing four objects of different shapes onto the conveyor belt in turn, the experiments were conducted to investigate whether the application of machine vision is helpful for the motion control of the sorting manipulator by adjusting the experimental environment and the speed of the conveyor belt and using machine vision or not. This work provides a theoretical basis for future research on the motion control of the sorting manipulator using machine vision.

II. Machine Vision for Sorting Manipulators

Machine vision technology involves several disciplines, including image processing [9], artificial intelligence [10], pattern recognition [11], etc. Computers are used to simulate human visual ability, and their combinations with manipulator catching technology allow arms discriminate and grasp like humans. This paper established a manipulator vision system to realize the camera calibration of the machine vision part and carried out image pre-processing [12] and feature extraction to match and recognize objects. The system determined the catching coordinate position by the image edge contour information of objects, and then positioned, grasped, and sorted objects.

A. Camera Calibration

The purpose of camera calibration [13] is to correct the distorted image and construct a three-dimensional scene based on the obtained image. It is assumed that there is a point named P in the space [14]. Point P is converted into a camera image plane to obtain the image pixel coordinates to realize manipulator catching. The specific conversion process is as follows.

$$S\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & u_0\\0 & f_y & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}R & T\\0 & 1\end{bmatrix}\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix}, \quad (1)$$

where S is a scale factor, R is a 3*3 rotation matrix, T is a 3*1 translational transformation vector, $\mathrm{x_w}$, $\mathrm{y_w}$, and $\mathrm{z_w}$ are the homogeneous coordinates of a point in the space under the world coordinate system and camera coordinates, f_x and f_y are the focal lengths of the camera in the X/Y directions, respectively, and u_0 and v_0 are the position of the image coordinate system origin O in the pixel coordinate system.

$$M_1=\begin{bmatrix}f_x & 0 & u_0\\0 & f_y & v_0\\0 & 0 & 1\end{bmatrix}, \quad (2)$$

$$M_2=\begin{bmatrix}R & T\\0 & 1\end{bmatrix}, \quad (3)$$

where M_1 and M_2 represent the internal and external parameter matrices of the camera, respectively. The internal parameters of the camera include the internal information such as the focal length, optical axis, and focus position of the camera, while the external parameter of the camera is the conversion relationship between the world coordinate system and the camera coordinate system. The overall conversion relationship is as follows:

$$S\begin{bmatrix}u\\v\\1\end{bmatrix}=M_1M_2\begin{bmatrix}x_w\\y_w\\z_w\\1\end{bmatrix} \quad (4)$$

B. Image Pre-processing

1) Image filtering process: Median filtering [15] can eliminate isolated noise points in an image and effectively protect the boundary information of images, which will not cause severe blur to images like mean filter. The specific method is described below. First, sequence $\mathrm{f_1}$, $\mathrm{f_2}$,⋯, and $\mathrm{f_n}$ is defined, and the length of the window corresponding to the median filter is odd number L, L = 2n+1, where n is a positive integer. It is assumed that at a time point, $\mathrm{f_i}$ is the value of the signal sample located in the center of the window, and the signal sample inside the window is $\mathrm{f_{i-n}}$, ⋯, $\mathrm{f_i}$, ⋯, $\mathrm{f_{i+n}}$. These L signal sample values are ranked from large to small, and the i value in the middle is taken as the output value of median filtering:

$$X_i = med\{f_{i-n}, \cdots, f_i, \cdots,\ f_{i+n}\} \quad (5)$$

To obtain better image processing result, the two-dimensional median filtering is used:

$$\mathrm{g(x,y) = med\{f(x-k, y-l), (k,l) \in W\}} \quad (6)$$

where f(x,y) is the original image and g(x,y) is the image after median processing. W is a two-dimensional template, usually 3*3 or 5*5, and in this paper, a 3*3 two-dimensional region is chosen.

2) Image grayscale: Regarding image grayscale [16] in the RGB model, the value of R=G=B is called the grayscale value, and the grayscale range is [0,255]. When the gray scale is 255, it is the brightest (pure white); when the gray scale is 0, it is the darkest (pure black). There are two main implications of converting the captured sorter image to grayscale image, one is that the grayscale image takes up less memory and has faster computing speed compared to the color image; the second is that the contrast is more obvious visually after conversion to grayscale image, highlighting the object location. After grayscale processing of the image of the sorter captured by the camera as in Fig. 1, the next step of feature extraction operation can be performed.

Fig. 1. Comparison of before and after grayscale processing of the sorter.

3) Image segmentation: Image segmentation [17] is a key step from processing an image to analyzing it, and is an indispensable pre-processing for recognition images and computer vision. In the acquired image of the sorter, the identified object is found to be only a small part of the overall image, and the excess makes the system slow in processing the image, so the image is segmented to highlight the object. In this paper, Matlab code [18] will be used to segment the sorter images, using the functions shown in Table I.

TABLE I. MATLAB IMAGE SEGMENTATION FUNCTIONS AND THEIR FUNCTIONS

Function Name	Function
imshow	Show images
imfinfo	Read information about the image file
imhist	Calculate and display the histogram of an image
Imadjust	Contrast enhancement
edge	Detect image edges
imcrop	Cut image (x: the width of the cut image, y: the height of the cut image)

III. MANIPULATOR MOTION CONTROL

The manipulator is an artificial intelligence device, and its motion control is the basis for ensuring the stability of the arm's posture [19]. The kinematic analysis of the manipulator can be divided into forward kinematics and inverse kinematics. Forward kinematics means deducing the pose of the end-effector of the manipulator relative to the reference coordinate system when the geometric parameters of the connecting rod joint and the angular value of the joint angle have been known. Inverse kinematics means calculating the joint angle of the manipulator when the geometric parameters of the connecting rod joint have been known and the pose of the end-effector of the manipulator relative to the base coordinate system has been given. D-H parameters [20] determine the coordinate change relationship between adjacent joints by assigning a coordinate system to every joint of the manipulator (Fig. 2), then the joint transformation relationship are combined through a mathematical formula to determine the overall change relationship between the end-effector and the base to obtain the related kinematic equation. D-H parameters contain four basic parameters, as shown in Table II.

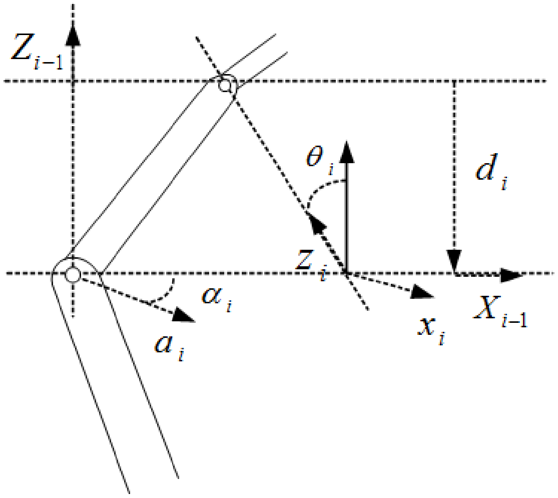


Fig. 2. Description of standard D-H parameters.

TABLE II. DESCRIPTION OF STANDARD D-H PARAMETERS

Name	Symbol	Meaning
joint angle (°)	α_i	The angle rotating from x_{i-1} to x_i around the z_{i-1} axis
connecting rod torsion angle (°)	θ_i	The angle rotating from Z_{i-1} to Z_i around the X_i axis
connecting rod length (mm)	a_i	The distance from Z_{i-1} to Z_i around the X_i axis
offset distance (mm)	d_i	The distance from X_{i-1} to X_i around the Z_{i-1} axis

In this paper, the equation of the inverse kinematics [21] is used: when the pose of the manipulator end coordinates has been known, joint variable θθ_i is calculated. The specific equation is:

$$ {}^0_iT = {}^0_1T(\alpha_1){}^1_2T(\alpha_2)\cdots{}^{i-1}_{\;\;\;i}T(\alpha_i) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (7)$$

where P is the translation vector used for determining the coordinates of the end of the manipulator in the space, n, o, and a denote the rotation vectors used for determining the attitude information of the end of the manipulator. The equation is solved by isolating the joint variables. Joint variable α_i can be obtained by left multiplication of the unknown inverse transformation of connecting rod and the two sides of (7).

IV. EXAMPLE ANALYSIS

A. Experimental Principle

The working principle of the sorting manipulator system used in this experiment is shown in Fig. 3. The image of an object is captured by an industrial-grade 3D smart camera [22]. The image data captured by the camera is transmitted to the input port of the image capture card. The image capture card transforms the analog video signal into a digital image. The computer program calculates the specific position of the object through the image and sends the position coordinates to the control system of the manipulator. Finally, the manipulator grasps and sorts the object.

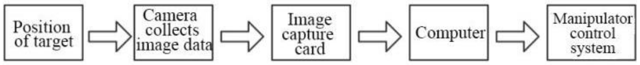


Fig. 3. Manipulator working principle diagram.

B. Experimental Design

The machine vision-based manipulator first scans the object with bilateral cameras, then pre-processes and recognizes the collected object image, and finally catches and sorts the object. Since the objects on the conveyor belt are dynamic and the background of the images captured by the cameras is variable, it is also important to study the manipulator's catching and sorting for dynamic objects under different experimental environments. The objects were placed under three different conditions, including different experimental environments, different object movement speeds, and with or without machine vision, 100 times each condition. Four objects of different shapes [23] were placed onto the conveyor belt in turn, and the conveyor belt and manipulator were turned on. The original environment of the experiment was an empty laboratory without any shade and with no shadow on the conveyor belt under the light. The main change of the experimental environment was to add large potted plants to the original environment and place them under the light to make their shadows cast the conveyor belt. The specific change before and after the experimental environment is shown in Fig. 4. The number of times the manipulator successfully caught and the number of times it correctly sorted under different experimental environments and movement speeds of the four objects were recorded, and the success rate was calculated. Finally, the catching and sorting results of the manipulator with and without machine vision were presented in the form of a table.

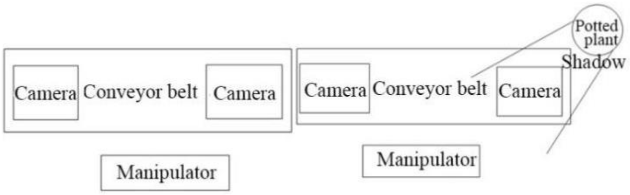


Fig. 4. Simple diagram before and after the change of experimental environment.

C. Experimental Results

Sorting accuracy formula: sorting accuracy = number of correct sporting/number of successful catch.

According to the experimental data displayed in Table III, it was seen that the manipulator completed the catching and sorting of objects of four shapes in the 100 times of catching and sorting under every condition. After calculation, the overall catching and sorting rates of the manipulator with machine vision were 96% and 97.91%, respectively, while the overall rates of the manipulator without machine vision were 84% and 57.99%, respectively. It was also found that the catching rates for the objects of four shapes showed decreasing trends, but the decreasing trend in the sorting rate was more significant than the catching rate. Thus, it was concluded that the manipulator with machine vision had a huge improvement for both object catching and sorting, i.e., machine vision had a positive influence on the motion control of the manipulator.

It was seen from Table IV that when the speed of the conveyor belt rose to 150 mm/s, the success rates of catching and sorting clothes, ball sports equipment and ring-pull cans gradually decreased, but the sorting success rate of the cardboard box remained at 100%. Overall it was seen that the sorting rate using the machine vision-based manipulator remained above 90%. This showed that although the number of successful catching and correct sorting decreased as the speed of the object moved faster, but the manipulator with machine vision could still effectively catch and sort objects. The results fully demonstrated that machine vision was beneficial to the motion control of the manipulator.

In order to compare the effect of the machine vision-based manipulator for object catching and sorting under different experimental environments, large potted plants were placed next to the conveyor belt to create a shadow influence. It was seen from the data in Table V that the catching and sorting rates of objects of four shapes in the complex environment decreased, and the catching rate of cardboard box decreased from 96% to 80%, which was the most significant. However, from an overall perspective, the manipulator with machine vision performed better than the manipulator without machine vision. This showed that the application of machine vision had a very significant impact on the sorting service of the manipulator and machine vision helped the manipulator to sort objects.

TABLE III. THE CATCHING AND SORTING RESULTS OF THE MANIPULATOR WITH AND WITHOUT MACHINE VISION AT THE SAME SPEED

	Clothes		**Ball Sports Equipment**		**Ring-pull Can**		**Cardboard Box**	
	Catching rate	***Sorting rate***	***Catching rate***	***Sorting rate***	***Catching rate***	***Sorting rate***	***Catching rate***	***Sorting rate***
with machine vision	92%	95.65%	96%	100%	100%	96%	96%	100%
without machine vision	84%	52.94%	80%	62.50%	80%	60%	92%	56.52%

TABLE IV. CATCHING AND SORTING TEST RESULTS OF THE MACHINE VISION-BASED MANIPULATOR AT DIFFERENT SPEEDS

	Clothes		**Ball Sports Equipment**		**Ring-pull Can**		**Cardboard Box**	
	Catching rate	***Sorting rate***	***Catching rate***	***Sorting rate***	***Catching rate***	***Sorting rate***	***Catching rate***	***Sorting rate***
100 mm/s	92%	95.65%	96%	100%	100%	96%	96%	100%
150 mm/s	84%	90.48%	92%	86.96%	80%	90%	96%	100%

TABLE V. CATCHING AND SORTING TEST RESULTS OF THE MACHINE VISION-BASED MANIPULATOR UNDER DIFFERENT EXPERIMENTAL ENVIRONMENTS

	Clothes		**Ball Sports Equipment**		**Ring-pull Can**		**Cardboard Box**	
	Catching rate	***Sorting rate***	***Catching rate***	***Sorting rate***	***Catching rate***	***Sorting rate***	***Catching rate***	***Sorting rate***
original environment	92%	95.65%	96%	100%	100%	96%	96%	100%
complex environment	92%	86.96%	92%	82.61%	92%	82.61%	80%	90%

V. DISCUSSION

The manipulator plays an important role in the field of automation engineering as an important component of industrial systems [24]. A study has developed an improved machine vision system that is capable of identifying and classifying items with different geometric shapes and colors and manipulating and separating them using a computer-controlled manipulator [25]. The experimental parameters in this paper differed from those in existing studies in the following aspects: (1) existing experiments use three cameras, while this paper used two cameras; (2) the sorted objects in existing research are in a condition of rest, while the sorted objects in this paper are in a state of motion; (3) existing research on manipulators under machine vision is oriented to grasp and sort items according to different geometric shapes and colors, but this paper took into account not only the different geometric shapes of the items, but also the different running speeds of the items on the conveyor belt and the complexity of the environment in which the manipulator performs sorting. In this paper, by reviewing a large number of literature on machine vision and robotic arm, the authors further deepened the research on the use of machine vision-based sorting manipulator on the basis of existing research and achieve good experimental results. However, there is still room for further research.

1) This paper showed that the correct sorting rate of manipulators using machine vision decreased to a certain extent in the case of high speed motion of sorted items and in the case of complex sorting environment, and subsequent studies can be conducted to analyze these problems.

2) To improve the sorting speed of the manipulator using machine vision, speeding up the image acquisition or transmission process can be considered.

3) Whether classification algorithms can be added for sorting items, and whether classification algorithms can play a role in improving the correct rate of robotic arm sorting should be studied.

VI. Conclusion

This paper mainly introduced the machine vision and the motion control of the sorting manipulator. After putting four kinds of objects of different shapes on the conveyor belt in turn, the cameras on both sides scanned the objects, the collected images were fed back to the manipulator control system after processing, and the catching and sorting behaviors of the manipulator were tested by adjusting the experimental environment and the speed of the conveyor belt and applying machine vision or not. The experiment found that among the 300 catching and sorting tests, the overall catching and sorting success rates of the manipulator were 96% and 97.91%, respectively, while the manipulator without machine vision had an overall catching rate of 84% and an overall sorting success rate of 57.99%. The experiment verified that the manipulator guided by machine vision could improve the ability to perceive and recognize the external world and achieve high accuracy in catching and sorting objects. Therefore, the application of machine vision has a positive impact on the motion control of the sorting manipulator, laying a foundation for the future application of machine vision in sorting service in practice.

References

[1] R. Li, R. Wang, and X. Tian, “Binocular vision mechanical arm system based on salient region target recognition,” J. Phys. Conf. Ser., vol. 1550, pp. 1-5, May 2020.

[2] J. Radcliffe, J. Cox, and D. M. Bulanon, “Machine vision for orchard navigation,” Comput. Ind., vol. 98, pp. 165-171, June 2018.

[3] Y. Min, B. Xiao, J. Dang, B. Yue, and T. Cheng, “Real time detection system for rail surface defects based on machine vision,” EURASIP J. Image and Vide., vol. 2018, pp. 1-11, January 2018.

[4] A. Liu, H. Zhao, T. Song, Z. Liu, H. Wang, and D. Sun, “Adaptive control of manipulator based on neural network,” Neural Comput. Appl., vol. 33, pp. 4077-4085, November 2020.

[5] A. C. Abad, D. D. Ligutan, E. P. Dadios, L. J. S. Cruz, M. C. D. P. Del Rosario, and J. N. S. Kudhal, “Fuzzy logic-controlled 6-DOF robotic arm color-based sorter with machine vision feedback,” Int. J. Adv. Comput. Sci. Appl., vol. 9, pp. 21-31, January 2018.

[6] B. B. Nair, S. Krishnamoorthy, M. Geetha, S. N. Rao, “Machine vision based flood monitoring system using deep learning techniques and fuzzy logic on crowdsourced image data,” Intell. Decis. Technol., vol. 15, pp. 357-370, September 2021.

[7] A. R. Mohamed, R. A. Elgamal, G. Elmasry, and S. Radwan, “Development of a real-time machine vision prototype to detect external defects in some agricultural products,” J. Soil Sci. Agric. Eng., vol. 11, pp. 317-325, May 2021.

[8] Y. F. He, and G. O. Tirian, “A method of detecting the feature of cylindrical pin based on machine vision,” J. Phys. Conf. Ser., vol. 1781, pp. 1-9, February 2021.

[9] G. Veerendra, R. Swaroop, D. S. Dattu, C. A. Jyothi, and M. K. Singh, “Detecting plant Diseases, quantifying and classifying digital image processing techniques,” Mater. Today Proc., vol. 51, pp. 837-841, June 2021.

[10] J. Gao, Y. Chen, and F. Li, “Kinect-based motion recognition tracking robotic arm platform,” Intell. Control Autom., vol, 10, pp. 79-89, January 2019.

[11] J. E. Miranda-Vega, M. Rivas-Lopez, and W. F. Fuentes, “k-nearest neighbor classification for pattern recognition of a reference source light for machine vision system,” IEEE Sens. J., vol. 21, pp. 11514-11521, September 2021.

[12] X. J. Wang, X. Hou, W. Niu, and S. Jiang, “Research on embedded navigation system of agricultural robot based on ARM,” J. Phys. Conf. Ser., vol. 1955, pp. 1-6, June 2021.

[13] Q. C. Hsu, N. Ngoc-Vu, and R. H. Ni, “Development of a faster classification system for metal parts using machine vision under different lighting environments,” Int. J. Adv. Manuf. Tech., vol. 100, pp. 1-17, February 2019.

[14] Q. Zhang, and Q. Wang, “Common self-polar triangle of separate circles for light field camera calibration,” Int. J. Pattern Recogn., vol. 39, pp. 521-528, June 2021.

[15] G. Wang, “Design and implementation of english text recognition system under robot vision,” J. Phys. Conf. Ser., vol. 1621, pp. 1-7, August 2020.

[16] S. Hu, “Research on data acquisition algorithms based on image processing and artificial intelligence,” IOP Conf. Ser. Mater. Sci. Eng., vol. 34, pp. 1-22, July 2020.

[17] A. Sardar, N. Mehrshad, and S. M. Razavi, “Efficient image segmentation method based on an adaptive selection of Gabor filters,” IET Image Process., vol. 14, pp. 1-12, December 2020.

[18] O. S. Osman, “MATLAB image processing tool-based GUI for high-throughput image segmentation and analysis to study structure and morphology of skin H&E stained sections,” J. Northw. Polytechn. Univ., vol. 737, pp. 1-8, March 2020.

[19] Q. Shi, and J. Zhang, “Research on manipulator motion control based on neural network algorithms,” Int. J. Biometrics, vol. 12, pp. 81-90, May 2020.

[20] M. Krämer, C. Rösmann, F. Hoffmann, and T. Bertram, “Model predictive control of a collaborative manipulator considering dynamic obstacles: NA,” Optim. Contr. Appl. Met., vol. 41, pp. 1211-1232, April 2020.

[21] G. Vladimirov, and S. Koceski, “Inverse kinematics solution of a robot arm based on adaptive neuro fuzzy interface system,” Int. J. Comput. Appl., vol. 178, pp. 10-14, August 2019.

[22] A. Fahruzi, B. S. Agomo, and Y. A. Prabowo, “Design of 4DOF 3D robotic arm to separate the objects using a camera,” Int. J. Artif. Intell. Robot., vol. 3, pp. 27-35, May 2021.

[23] T. O. Mon, and N. Zaraung, “Vision based volume estimation method for automatic mango grading system,” Biosyst. Eng., vol. 198, pp. 338-349, October 2020.

[24] H. W. Lee, “Study of a mechanical arm and intelligent robot,” IEEE Access, Vol. 8, pp. 119624-119634, June 2020.

[25] N. V. Ngo, G. A. Porter, and Q. C. Hsu, “Development of a color object classification and measurement system using machine vision,” Sensors Mater., vol. 31, pp. 4135-4154, October 2019.

Frequency Domain Improvements to Texture Discrimination Algorithms

Ibrahim Cem Baykal
Adana Alparslan Turkes Science and Technology University
Computer Engineering Department
Sarıçam, Adana, Turkiye

***Abstract*—As the production speeds of factories increase, it becomes more and more challenging to inspect products in real time. The goal of this article is to come up with a computationally efficient texture discrimination algorithm by first testing their ability to localize defects and then increase their efficiency by removing less effective parts of them. Therefore, abilities of the most popular texture classification algorithms such as the GLCM, the LBP and the SDH to localize defects are tested on different datasets. These tests reveal that, on small windows GLCM and SDH perform better. Frequency properties of the textures are used to fine-tune the parameters of these algorithms. Further experiments on three different datasets prove that the accuracy of the algorithms are increased almost twice while decreasing the processing time considerably.**

***Keywords*—Machine vision; ANN; SVM; pattern recognition; co-occurrence; texture feature extraction**

I. Introduction

Despite being old, based on the yearly citations they receive, Haralick’s grey level co-occurrence matrices (GLCM)[1], Unser’s sum and difference histograms (SDH)[2] and Local Binary Patterns (LBP)[3] are the most popular texture classifiers in the literature. As shown in Fig. 1, great majority of machine vision systems inspect a product simultaneously while it is being produced. Usually, imaging sensors are stationary and the product moves under them. This causes the viewing angle of the cameras to always be static with respect to the product. The lighting conditions in a factory are usually under some degree of control and do not change radically. When used for such purpose, these algorithms are usually applied to the texture towards the same alignment because in a factory the camera is usually at a fixed position with respect to the production line or the conveyor belt. Therefore, throughout the experiments performed in this article, texture alignment is fixed.

Another difference between the texture recognition and the discrimination is that the recognition applications process the entire image while the discrimination applications process smaller windows to localize where the defect is. Unfortunately, there are not many experimental studies in the literature that measures the effectiveness of algorithms on different (non-overlapping) window sizes. Texture segmentation [4] applications use sliding windows which makes them very slow.

Because of the rapid progress of technology, materials in factories are produced faster and faster. Flat surfaced products such as wood [5], fabric [6] and metal [7] require machine vision systems to detect production defects. Unfortunately, high-speed production of these goods combined with the availability of high-resolution cameras put great strain on the network and the computers that process those images in real time. A typical line-scan camera, employed in web inspection tasks for products such as film, paper and fabric, generates between 15 and 150 MB of data per second and half a dozen cameras are required to cover the entire width of the web, meaning that gigabytes of data must be reliably transferred and processed. One approach to this problem is to pre-process images inside the cameras and send the image data to the main computer only when the camera suspects that there might be a defect [8]. In this study, Brodatz texture set is experimented on with popular texture classification algorithms to see which one performs better on small windows. Based on these results, improvements to these existing algorithms are proposed. These modified algorithms are tested on three different datasets.

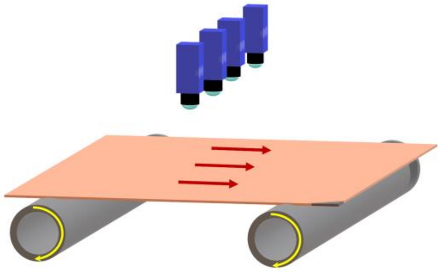


Fig. 1. Simultaneous manufacturing and inspection of a flat product inside a factory.

II. Related Work

GLCM are calculated by counting the relative pixel value occurrences that are away from each other (d) pixels. The result is a histogram matrix of size N×N. N is defined by the maximum number of gray levels in the image. Therefore, for greyscale images it is usually 256×256=64K elements. Haralick proposes 14 features that can be calculated from the GLCM, which requires processing of those 64K elements 14 times. This not only requires considerable amount of processing power but also consumes several 64 KB of memory. Considering the fact that the GLCM was invented in 1973, this was a problem for the computers back then. One solution to overcome this problem was to use 16 grey levels for the image. That solution reduces the matrix size to 16×16.

Unser proposed another optimization to this problem. He mathematically proved that almost the same results could be achieved by using the sum and difference histograms (SDH) of those pixel pairs. Instead of a matrix, his algorithm generates two vectors, one for the sum of the pixels and one for the difference for every given distance ‘d.’ An image with 256 gray levels produces two vectors with 512 elements. Calculating those 14 features that Haralick suggested becomes computationally simpler. This method's main advantage is the fact that it reduces the amount of memory. However, based on our experiments (see Section VIII) it increases the computational complexity compared to the GLCM if reduced number of gray levels are used. In the literature, SDH has been tested on many texture classification problems and it has been proved that it is a quite successful texture classifier.

Texture analysis has been the focus of numerous studies over the years and as a result, over two dozen algorithms have been invented to classify texture. In volume II of “Handbook of Computer Vision and Applications,” Wagner [9] experimented on seven different texture sets using 18 different texture classification algorithms. His experiments showed that Unser's SDH is not only the most successful among the 18 algorithms, but also the second fastest. Based on the measurements of that study, the only algorithm faster than the SDH is the Local Textural Features [10]. However, that algorithm had the second lowest success rate. Therefore, according to that study, SDH is computationally the most efficient texture classifier. However, it should be noted that the GLCM was probably calculated using the full 256 gray levels making it much slower compared to the SDH. The second most successful algorithm is Chen’s geometric features [11] which is the slowest among the 18 algorithms (30 times slower than the SDH). The third successful algorithm is the Gabor Filters [12], which is the second most popular algorithm used for fabric defect detection in the literature. Gabor Filters compare energy of the specific frequency bands on fabric images to detect defects. Our method to select pixel distance and the orientation is mathematically related to Gabor Filters. Another popular algorithm in the literature is Law's Masks [13]. Laws has suggested five 1-D convolution masks for feature extraction. From these masks 25 2-D filter masks can be constructed and features are calculated from them. According to Wagner [9], Law's Masks performed slightly worse than the SDH, but is eight times slower. Galloway’s Run-Length Matrices [14] should also be mentioned because of their speed. His technique calculates characteristic textural features from gray-level run lengths in different image directions and then calculates five features from them for each direction. However his algorithm did not score as high as the previously mentioned algorithms according to Wagner.

The literature still lacks a mathematical definition of the texture. Therefore, humans are the only referee when it comes to judging which texture is the same or different. That is why an interesting experiment conducted by Petrou [15], comparing human perception to the texture classification algorithms is so important. In this study, classification performance of numerous algorithms were compared to a group of humans. Variants of both the GLCM and the SDH, with different gray levels were experimented with. These experiments showed that direct use of GLCM or SDH as input to a classifier resembles to human perception much closer than using only the features derived from them. SDH was measured as one of the fastest. SDH using16 gray levels is closer to human perception than the GLCM as well. In his exhaustive experiments, Gonzales-Rufino [16] shows that GLCM and SDH perform better against the Local Binary Patterns, and the Neighboring Grey Level Dependence Statistics [17]. In the literature, there is abundant evidence that second order pixel statistics based texture classification is one of the most successful and the most efficient method. All this evidence suggests that, further improving SDH might give us computationally the most efficient texture discrimination method.

III. Performance Comparison on Small Windows

In the literature, mathematical definitions of the texture is very vague: “A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic” [18]. Because of the fact that a texture is defined in terms of “statistics” and “periodicity,” a texture can exist only as a group of pixels, or in other words as a window. As mentioned before, texture discrimination requires localization of texture aberrance. Therefore, we need to measure the performance of texture classifiers on varying size of windows.

For this experiment, four different size windows were used on 13 different Brodatz textures. The size of the windows are 240×240, 120×120, 60×60 and 30×30. As shown in Fig. 2, the right half of the images are used for training and the left halves are used for testing. Overlapping blue squares represent training locations and green squares represent non-overlapping test locations. The ones at the top use 120x120 pixel windows and the ones at the bottom use 30x30 pixel windows. Green boxes indicate correctly classified windows and red ones indicate falsely classified windows. The numbers inside the red boxes indicate which of the 13 textures that window was misclassified as.

For this experiment, six different feature sets were used. The first two feature set is the four features obtained from the GLCM. In the literature GLCM is mostly used in either 256 color mode or 16 color mode. Therefore, both versions were tried. The third feature set is the 32 gray level histogram plus Unser’s difference histograms with 16 colors (H-32+DH-16). The final three feature sets are LBP with (r=1;P=8), (r=1,2;P=8) and (r,P)={(1,8);(2,8)};(3,12)}. Either Artificial Neural Networks (ANN) or Support Vector Machine (SVM) were used as classifiers. Table I shows that ANN is slightly more successful than the SVM. While LBP is more successful on larger windows, 16 color GLCM and (H-32+DH-16) are more successful on smaller windows. In other words, these algorithms are more suitable for defect localization or texture segmentation applications. However, readers must keep in mind that “window size” is a relative value, which depends on the image resolution. If the image of the texture is high resolution, than these window sizes must be increased. More detailed results of this experiment were shared in a conference article [19].

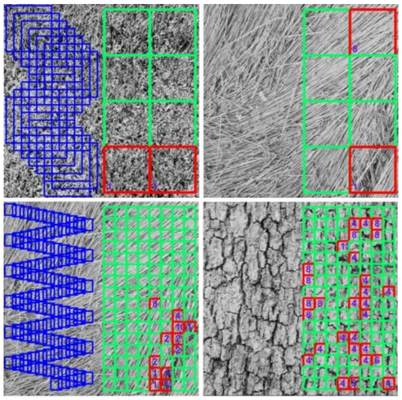


Fig. 2. Top: 120×120 pixel size window test. Bottom: 30×30 pixel size window test. Left half of the images are used for training. Overlapping blue boxes are training areas. Non-overlapping green windows on the right half of images are correctly classified windows.

TABLE I. NUMBER OF CORRECTLY CLASSIFIED WINDOWS

Window Size (pixels) (total test windows)	**GLCM-256**		**GLCM-16**		**H-32+DH-16**		**LBP(r=1;P=8)**		**LBP(r=1;P=8)**		**LBP(r=1;P=8)**	
	ANN	***SVM***	***ANN***	***SVM***	***ANN***	***SVM***	***ANN***	***SVM***	***ANN***	***SVM***	***ANN***	***SVM***
240×240 (**26**)	**26**	17	**26**	18	**26**	19	**26**	**26**	**26**	**26**	**26**	**26**
120×120 (**104**)	102	73	102	94	101	74	**104**	99	**104**	99	**104**	99
60×60 (**416**)	393	367	**408**	378	402	253	401	399	396	400	375	400
30×30 (**1664**)	1558	1480	**1575**	1536	1552	1556	1459	1483	1503	1519	1431	1526
TOTAL (2444)	2079	1937	**2111**	2026	2081	1902	1990	2006	2029	2044	1936	2051

IV. PROPOSED MODIFICATIONS

Despite the growing popularity of the LBP, the previous test shows that on smaller windows (patches), performance of GLCM and "histogram plus difference histograms" are superior. In his article, Unser proves that his sum and difference histograms (SDH) are mathematically equivalent of the GLCM. The reason why he proposes them is because SDH requires less memory and processing power. Since one of the main objectives of this study is to come up with a texture discrimination (defect detection) algorithm that can fit inside a camera, it is natural that we chose to continue with Unser's sum and difference algorithms. Unser also states that %99 of the texture information comes from the difference histogram rather than the sum histograms.

Another reason why difference histograms are used is that they are immune against illumination changes. Despite the fact that illumination conditions are under control in factory settings, we would still want our algorithm to be as robust as possible to illumination changes. This can be explained using the mathematical model of the light absorption. Let's assume that a light source is sending photons to a textured surface at an illumination strength of 200. Let the first area corresponding to the pixel-1 has a reflectivity of %90 and the second area corresponding to the pixel-2 has reflectivity of %50. Then the camera would detect the photons coming from these two areas at a strength of 180 and 100 respectively. The sum of these pixels would be 280, and their difference would be 80. If the illumination is increased by %20, then these pixels would be registered as gray levels of 240×0.9=216 and 240×0.5=120. Their sum would be 346 and their difference would be 96. The fluctuation for the difference histogram would be much lower because the increase in the intensities would cancel each other out. If we use 16 gray levels instead of 256, then the difference values for those two different illumination conditions would be 5 and 6. This is only one level shift for %20 change in the illumination, which means that for lower illumination changes or for pixels with less contrast difference, it would not be noticeable at all. Nevertheless, applying the histogram equalization would be enough for this algorithm to withstand any reasonable illumination fluctuations. On the other hand, even with the histogram equalization, sum vectors would be very sensitive to any illumination change.

A. Frequency Domain Fine Tuning

There are dozens of articles in the literature that present successful results of both Haralick's and Unser's method on texture classification and discrimination problems. As Haralick and Unser both did, almost every one of those studies set the pixel distance "d" to 1. There is no explanation in Haralick's paper why he used d=1. Using 1 for the pixel distance has become so common that some articles don't even mention what distance they used [20]. The rest of them either use 1 or use 1 along with some other pixel distance, such as 5s. For example, a fabric defect detection experiment done by Latif-Amet [21] also set pixel distance to 1. A defect detection study on aluminium surfaces conducted by Chondronasios [22] selects d=1 and Georgieva [23] uses two different pixel distances, d=1 and d=5 to classify the cork tiles.

Because the texture pixels are defined as "a region slowly varying, or approximately periodic" [6], we will build our mathematical model based on this. If we scan a texture through a straight line at any given angle, the one dimensional signal obtained should be:

$$f(x) = a_0 + \sum_{n=1}^{N} An\,cos\left(2\pi\frac{n}{k}+\theta_n\right) + \varphi(x) \qquad (1)$$

Where $\varphi(x)$ is the noise. Since we want to calculate histograms of pixels that are "d" pixels apart, and assuming that the texture has frequencies that are integer multiple of themselves, the values of the two pixels that are "d" pixels apart would be:

$$f(p_{x1}) = a_0 + \sum_{n=1}^{N} An\ cos\left(2\pi \frac{nx1}{k} + \theta_n\right) + \varphi(x1)(2)$$

$$f(p_{x2}) = a_0 + \sum_{n=1}^{N} An\ cos\left(2\pi \frac{n(x1+d)}{k} + \theta_n\right) + \varphi(x1 + d)\ (3)$$

If our eyes can detect obvious periodicity in a texture, such as the ones created by periodic weavings of a fabric, then we can assume that most of the cosine terms have frequencies that are integer multiple of the main repetition. This means that, when the pixel distance "d" is equal to exactly one period of the texture, the difference of these two pixels will cancel out all of the cosine terms:

$$f(p_{x1}) - f(p_{x2}) = \varphi(x_1) - \varphi(x_1 + d) \qquad (4)$$

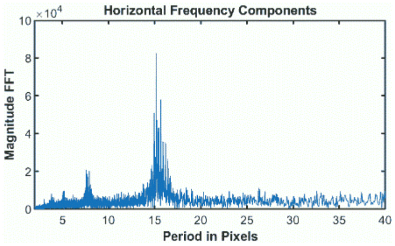
Naturally, we assume that the biases cancel each other out as well and all that remains are the two random variables. If the pixel distance does not equal to one period of the sinusoid, there will always be cosine terms in (4), meaning that the signal's variance would be much higher. High variance is not a desired property for a feature. This is the main reason why we use difference histograms (DH).

As shown in Fig. 3, if we scan through the texture image in four different orientations and calculate the magnitude of the Fourier Transform (FT) of that signal, the highest peaks in the magnitude of the FT will tell us which direction and distance must be used for the calculation of difference histogram.

We can use another method to verify our approach. If we have large number of defective samples (or areas), we can perform Fisher's Discriminant Analysis (FDA). The difference histogram vectors must be calculated on non-defective and defective windows for a range of "d" (let's say from 1 to 25) and for all of the orientations. Then the FDA of each element must be calculated and summed. The pixel distance and the orientation with the highest FDA sum should give us the optimum pixel distance "d". Fisher's Discriminant Ratio (FDR) is defined as:

$$FDR = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2}\ (5)$$

In this equation, m1 and m2 are the respective mean values, and σ_1 and σ_2 are the respective deviations associated with the values of a feature in defective and non-defective classes. Collectively these modifications will be referred to as "Diftogram." The following sections will show how this is done through a few examples.



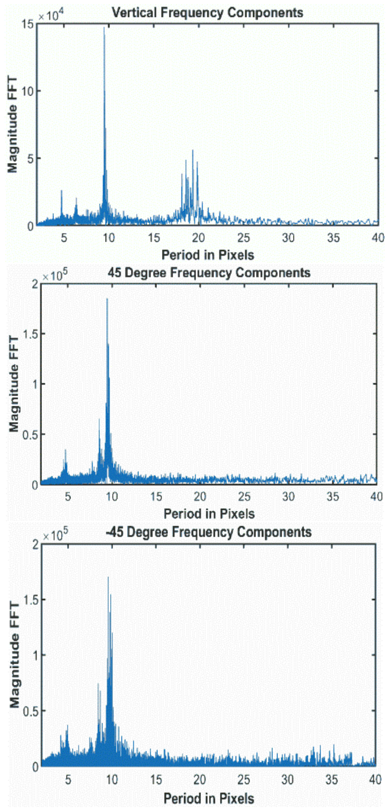


Fig. 3. Magnitude FFTs of the signals obtained by scanning the fabric image horizontally, vertically, at +45 and -45 degree angle.

V. Dataset-I

A typical fabric sample is used for the first experiment. Fabric defect detection is one of the areas of industrial machine vision where texture discrimination is widely used. In this experiment, 12 images with natural fabric defects such as broken thread, missing thread, stuck thread, weaving error, punctures, and lubricant oil stains are used. In order to compare the effects of modifications, the co-occurrence matrices are used as in the conventional way. In order to increase robustness against illumination changes, histogram equalization is applied to the area of interest. This can be noticed by the slight increase of contrast around the perimeter of the white boxes in all of the fabric images. GLCM are calculated for each of the 80×80 pixel windows on 1024×720 pixel images with pixel distance set as d=1. Co-occurrence matrices are calculated for orientations of 0, +45, +90 and -45 degrees using reduced gray levels of 16. Alongside the 14 features proposed by Haralick, the two features proposed by Clausi [24] and the three features proposed by Soh [25] are also calculated. This is necessary to prevent possible criticism that GLCM method is incomplete without the newly added features and that the results could have been better. In Section 3, the ANN performed better than the SVM. Therefore, a feed forward ANN with 25 neurons in hidden layer is used for all of the experiments.

Total of 19 features, 14 from Haralick, 2 from Clausi and 3 from Soh are calculated from every matrix. Since there is one matrix per each of the four directions, a total of 76 features are calculated to be used as input to the neural network. The training and the test sets are the same. The Neural Network toolbox of the Matlab software was used to train the ANN. The ANN training process is not a deterministic one. Every time a neural net is trained, even with the same training set and the parameters, results will be slightly different. Because of that, the experiment was conducted more than 30 times and the most successful ANN among them was selected. Results of this experiment are shown in Table II and Fig. 4. The white windows represent 80×80 pixel squares that are defined as "normal" by the ground truth and are also correctly classified as non-defective by the algorithm. The green boxes are drawn if both the ground truth and the algorithm classifies them as defective. In other words, correctly classified areas are marked as either white or green boxes. The black color window represents that the window was marked as defective but the algorithm failed to detect it. If the window is red, this means that a non-defective area was incorrectly classified by the algorithm as defective. Table II reveals that the GLCM algorithm using d=1 and using four orientations was able to detect 14 defects out of 42, and misclassified 1 or 2 areas as defective.

In order to see the effect of using Unser's difference matrices as direct input only, without applying the method to find the most effective pixel distance and orientation, the experiment was repeated using Unser's difference histograms with d=1 and in the four orientations. It was observed that the ANN had a very low margin to distinguish defective and normal areas. Because of this, two different outcomes were observed. In some of the trials, ANN recognized 37 of the 42 defective regions with 27 misclassification of normal areas whereas in the other trials it recognized 28 defective areas with 19 misclassification of non-defective areas. Both of these results are given in Table II.

To measure the success rate of the proposed method against other algorithms, the most effective pixel distance 'd' needs to be calculated. In order to achieve that, a fabric image without any defects must be scanned through four directions to obtain one dimensional signals. Afterwards, the magnitude of the Fourier transform of these signals should be calculated as shown in Fig. 3. Each signal has one dominant peak. The pixel period corresponding to those peaks should be used as the pixel distance.

To verify, or perhaps to get a second opinion, Fisher's Discriminant Analysis should also be used. As the first step, we calculate all of the DH for each of the 80×80 pixel windows for all of the twelve fabric images, using all possible pixel distances between one and twenty five. Afterwards we calculate the FDR value for each element in the difference vector using (5), and then sum these FDR values. The pixel distance and orientation with the highest FDR sum value is the one that should be used for texture discrimination. Below are the FDR sums for the elements of the vectors obtained for 25 different distances and four orientations. For 0-degree orientation:

1 2 3 4 5 6 7 8
FDR(Θ = 0)=[0.39 0.26 0.76 1.12 2.20 3.02 3.94 4.23
9 10 11 12 13 14 **15** **16** 17 18
19
3.99 3.93 3.75 4.07 7.22 13.81 **21.88** **25.34** 17.65 10.89
6.66
20 21 22 23 24 25
4.48 3.93 4.48 5.23 5.63 5.33]

The largest FDR sum values are for d=15 and d=16 for Θ equals 0 degrees (horizontal direction). When we look at the Fourier transform of the horizontal signals in Fig. 3, we see that the peak magnitude is exactly at d=16. Proceeding with the vertical direction,

1 2 3 **4** **5** 6 7 8
FDR (Θ = 90) = [2.76 4.65 5.43 **6.02** **5.51** 4.58 3.44 3.00
9 10 11 12 13 14 **15** **16** 17 18 19 20
21
3.21 3.51 3.53 2.47 2.70 3.75 **5.02** **5.66** 4.54 2.62 2.87 3.68
1.74 1.15 2.49 3.03 2.82]

FDR sum values reach their maximum at d=(4,5) and d=(15,16). As seen in the Fourier analysis of the vertical signal, there is a small peak at d=5 but the most important peak at d=10 has no effect on the FDR values and instead of the flat peak around d=19 we the values at d=15 and 16 of the FDR are higher. It is possible that the very narrow peak at d=10 is not detected and the defects have their own characteristics that is altering the values of the FDR. Because there are more than one significant cosine components in the vertical signal, the FDR values in the vertical direction are far smaller than that of the horizontal. This increases the variance of the signal and as a result, reduces the FDR values. For +45 orientation we get d=9, and for -45 degree orientation we get d=10.

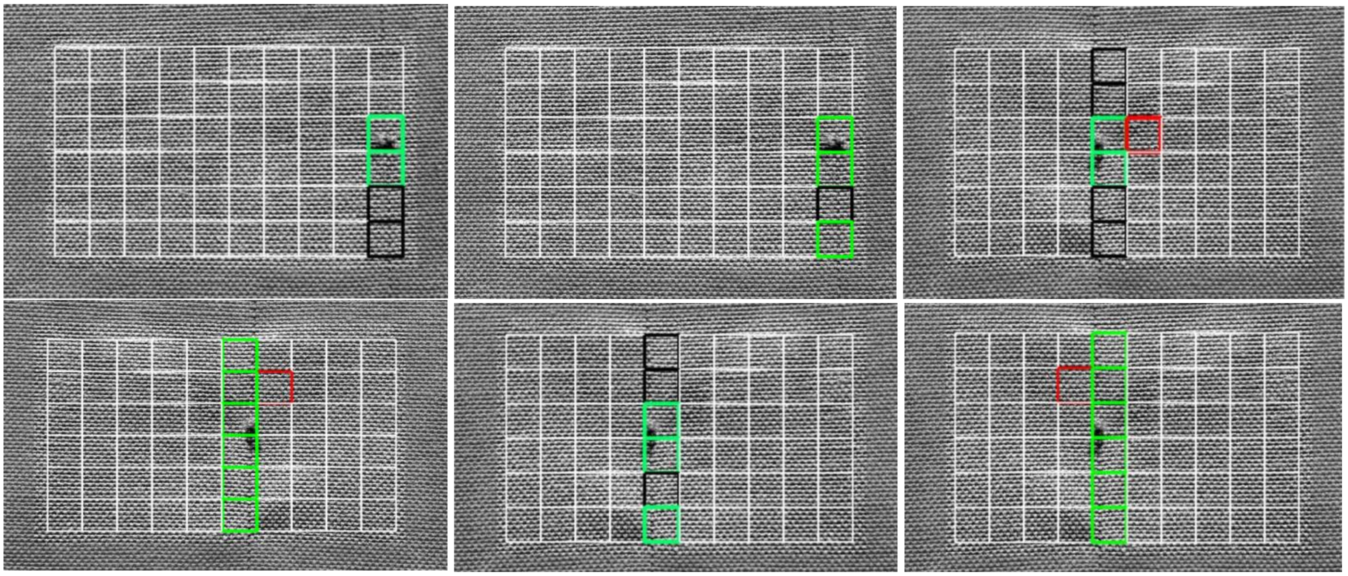
1 2 3 4 5 6 7 8
FDR (Θ = + 45)=[2.70 4.28 5.48 6.00 5.70 5.52 5.47 5.78
9 10 11 12 13…
10.49 6.78 1.26 1.92 3.31 4.25 5.03 5.43 3.92 3.61 6.10
2.38 0.76 1.36 2.81 3.55 3.56]
1 2 3 4 5 6 7 8
FDR (Θ = - 45)=[1.57 3.01 4.39 5.80 7.10 8.08 7.38 5.70
9 **10** 11 12 13…
5.62 **7.98** 3.24 0.61 1.59 3.88 5.32 5.74 4.69 2.67 2.39 4.05
3.29 1.47 1.68 3.45 4.12]

The FDR values for 0 degree orientation clearly indicate that difference histogram should be applied in the horizontal direction and the pixel distances should be set as d=15 and d=16. Using these parameters Diftogram's success rate is shown in Table II. The algorithm classified 37 defective areas out of 42 correctly and misclassified 6 areas as defective. That is almost 2.5 times (37/14=%264) higher defect recognition rate than the GLCM experiment.

As shown in Table II and Fig. 4, the results are considerably superior against the LBP. Some readers might be confused that the training and test sets are the same. The purpose of this study is not to measure the actual success rate of a single algorithm but to measure relative success rate of several algorithms with respect to each other. As long as the training conditions are the same, these results will give us which algorithm performs relatively better. This will be proven in Section VII (Dataset III) where we perform training on different training and test sets and the relative success rates remain similar.

TABLE II. DISCRİMİNATİON RESULTS FOR DATASET-I

Algorithm	Defects	False +	Accuracy	Precision
	Max 42	*Max 678*		
GLCM-256 d=1; 4 orientations	14	2	0.9583	0.8750
GLCM-256 d=1; 4 orientations	15	1	0.9611	0.9375
LBP (r=1,p=8) (r=2, p=8)	20	1	0.9681	**0.9524**
LBPri	25	2	0.9736	0.9259
DH d=1; 4 orientations	**37**	27	0.9556	0.5781
DH d=1; 4 orientations	28	19	0.9542	0.5957
DH d=**15,16** Horizontal	**37**	6	**0.9847**	0.8605



GLCM-256 with 4 orientations d=1. LBPri DH, horizontal, d=15 &16

Fig. 4. Two of the defective fabric samples: Broken thread (top) and stuck thread (bottom). White boxes are correctly classified normal areas and green ones are correctly discriminated defective ones. Black ones indicate the regions that the algorithm failed to discriminate and red ones are incorrectly classified defects.

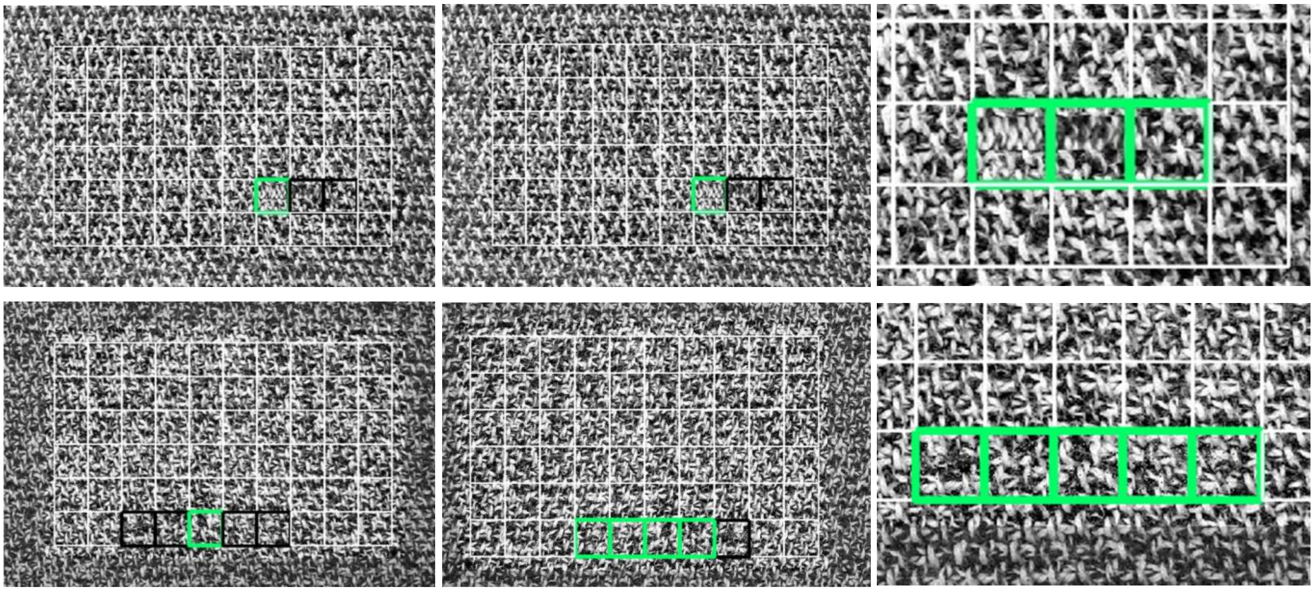
VI. DATASET-II

A slightly more challenging fabric sample is selected for this experiment. The thread used for weaving this fabric has an alternating colour making the noise level of the texture extremely high. The lighting conditions were also altered. For this experiment, multiple light sources were used. The first three fabric samples in the set were taken under %20 higher illumination then the rest to demonstrate illumination change's effect. Another challenging part of this experiment is that the defects were artificially created to make them extra difficult to detect. The defects are so mild that even the human eye can barely notice them. In order to do so, 15 different images of the fabric were taken and then using the GNU Image Manipulation Software, GIMP (https://www.gimp.org), different kind of defects were created using blurring, Gaussian and salt and pepper noise, contrast altering, spatial compression, pixel smearing and etc.

Table III and Fig. 5 shows the results of the GLCM algorithm using 256 and 16 grey levels, four orientations and 19 features calculated from each orientation. Pixel distance was set to d=1, similar to almost every GLCM experiment published in the literature. The classifier is the same ANN used for the previous experiments, a feed forward network with 25 neurons in hidden layers. The training set and the test set are the same, the training was repeated more than a dozen times and the most successful one is shown here. This combination of algorithms classified 20 defects correctly out of 64. Zero or two normal areas were misclassified as defective.

TABLE III. DISCRİMİNATİON RESULTS FOR DATASET-II

Algorithm	Defects	False +	Accuracy	Precision
	Max 64	*Max 836*		
GLCM-256 d=1; 4 orientations	20	0	0.9511	**1.00**
GLCM-16 d=1; 4 orientations	22	2	0.9511	0.9167
LBP (r=1,p=8) (r=2, p=8)	28	0	0.96	**1.00**
LBPri	28	1	0.9589	0.9655
DH d=1; 0 & 90 degrees	**59**	1	**0.9933**	0.9833



GLCM-256 with 4 orientations d=1. LBPri DH d=1, 0 & 90 degrees

Fig. 5. Samples from dataset II. Above: A different kind of weaving was cut and paste on it as defect. Below: Added salt and pepper noise. Green boxes are correctly discriminated defective areas and black boxes are the ones that the algorithm failed. Images on the right shows zoomed defects.

Before proceeding to the second part of the experiment, the Fourier analysis of the fabric image must be performed. The absolute value of the FFT obtained by combining many lines from several images shows that there isn't any salient sinusoid in this signal. The FFT results are similar in other directions therefore they are not shown. Under such circumstances, the highest peak is usually obtained at zero frequency. This means that the FDR analysis should give us d=1. When we perform the FDR analysis that is exactly what we get:

FDR(Θ = 0)=[**5.97** 2.37 1.41 1.39 1.68 2.00 2.17 2.19 2.18 2.05 1.97 1.99 2.03 1.91 1.71 1.64 1.50 1.34 1.25 1.26 1.32 1.29 1.24 1.22 1.12]

FDR(Θ = 90)=[**5.50** 1.89 0.57 0.52 0.77 1.25 1.58 1.94 2.18 2.33 2.43 2.47 2.29 2.01 1.80 1.81 2.03 2.24 2.63 2.97 3.42 3.58 3.67 3.58 3.43]

Similar results are obtained for 45 and -45-degree orientations. It seems that selection of pixel distance as 1 is not a bad choice if the texture has a lot of noise or there are more than a few high amplitude periodic components in the texture.

For the next part of the experiment, Unser's difference histograms are directly used as the texture features. The grey levels are reduced to 16, pixel distance is set to one, and only two orientations, horizontal and vertical are selected. In the first part of the experiment where GLCM was used, same parameters were used except for the fact that, 45 and -45-degree orientations were also used with the GLCM. These two directions, 45 and -45 are skipped to demonstrate the power of the proposed method and also to make the algorithm twice as fast. The elements of the two difference vectors are directly fed to the same ANN used for the previous experiments. As shown in Table III, this algorithm recognized 59 defective regions out of 64 and misclassified one normal area as defective. Compared to the defect detection rate of the GLCM algorithm, this is almost %300 higher. Against the LBP variants, it is almost %200 more successful.

VII. Dataset-III

For the third experiment, a fabric sample with a patterned weaving is used. The fabric is weaved to display a repeating pattern, which creates a secondary pattern on top of the pattern created by the threads of the fabric. Lighting conditions were slightly altered for many of the 45 image samples of this fabric. The defects were created using the GIMP software, but this time with increased variety and larger surface area. These defects vary in strength; about half of them are easily noticeable by the human eye. The frequency analysis in Fig 6 shows that this fabric has a major component at a period of around 27 pixels in the horizontal direction. At 45 degrees, the major component is at 13 pixels (not shown) and for the vertical direction, there are three peaks at 9, 14 and 25 pixels. The FDR analysis show consistent results with these findings:

FDR(Θ = 0)=[3.63 2.11 1.60 0.56 2.12 0.46 1.55 0.76 0.54 1.03 0.79 1.35 2.07 1.79 1.66 1.13 1.09 0.43 0.6 0.53 0.30 1.24 1.29 0.86 1.12 2.18 **6.10** 4.92 1.89 0.95]

FDR(Θ = 45)=[3.75 2.57 2.60 0.75 0.51 1.51 2.66 0.75 0.47 0.55 1.32 1.97 **6.41** 4.25 1.21 1.55 1.17 0.74 0.69 0.90 0.56 0.3814 0.73 1.03 1.66 2.30 2.06 0.95 0.50 0.51]

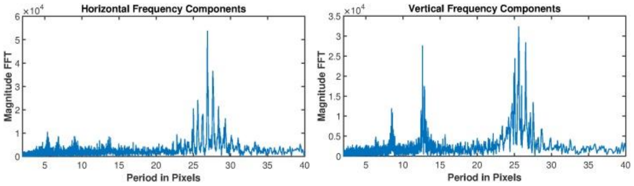
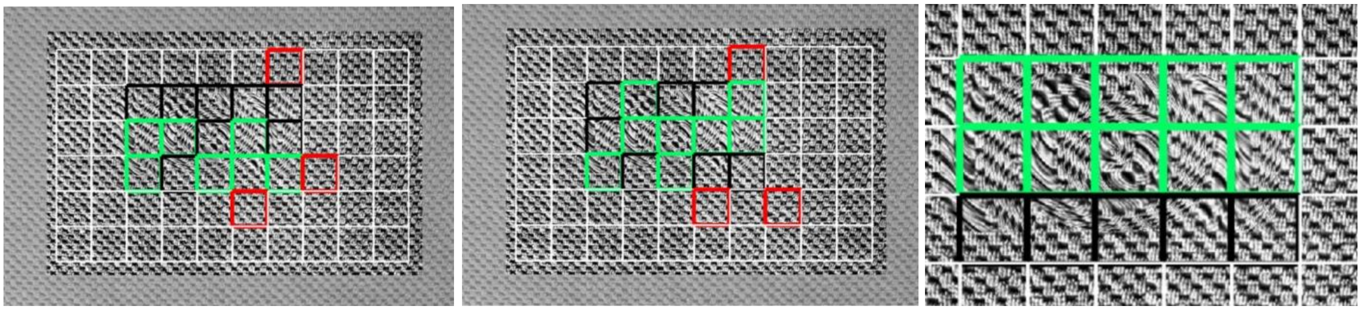


Fig. 6. Magnitude of the FFT of the signal obtained by scanning the fabric of dataset III, horizontally and vertically.

FDR(Θ = 90)=[**6.16** 2.65 1.21 0.86 0.82 1.08 1.18 1.57 1.85 1.56 1.40 1.19 1.07 1.54 2.01 1.91 1.68 1.18 0.87 0.72 0.64 1.15 2.11 3.14 3.87 2.83 1.52 0.78 0.76 1.17]

Because of the abundance of distinct frequency components in the vertical direction, (Θ = 90), the FDR analysis show that there is no advantageous pixel distance in the vertical direction other than d=1. For the 0-degree orientation, d=27 and for the 45-degree orientation d=13. Fig. 7 and Table IV shows the results of discrimination tests for the GLCM with d=1 in four directions, Diftogram with d=27 for the horizontal and d=one for the vertical orientations and the LBP. Diftogram is approximately %250 more successful than the GLCMs and %200 more successful than the LBPs. Notice also that the number of false positives are smaller than both the GLCMs and the LBPs. This result is slightly lower than the previous examples. The reason for that is because the fabric has two different patterns on it and the defects are stronger. Much higher results could be obtained by adding more orientations and pixel distances to the Diftogram if computational efficiency is not essential.

Because we are testing the reliability of different kind of features against each other, there is nothing wrong about using the same images for both training the ANN and testing it as long as the ANN is trained the same amount for each feature set and the ANN is not trained excessively to the point where it memorizes the samples. In other words, if the ANN has enough generalization capability, it will measure how well the features separate the defects from the background. The default parameters of the ANN trainer in Matlab has very good balance between memorizing and generalizing therefore, the experiments conducted so far has measured the reliability of the features quite well. Unfortunately, there might be people who do not trust such findings and demand that the training and the test sets should be different. For those people the third experiment is conducted again, this time using the odd numbered 23 images (1, 3, 5, ... 45) for training and the other 22 for testing. Table V shows that the Diftogram is still about two times more successful than both the GLCMs and the LBPs.



GLCM-256 with 4 orient. d=1.

LBPri

DH d=27, horizontal & d=1 vertical

Fig. 7. A sample from dataset III, with optical distortions on it. Black boxes indicate the regions that the algorithm failed to discriminate. White boxes are correctly classified normal areas and green ones are correctly discriminated defective ones. Red boxes are false positives.

TABLE IV. DISCRİMİNATİON RESULTS FOR DATASET-III

Algorithm	Defects	False +	Accuracy	Precision
	Max 361	*Max 2339*		
GLCM-256 d=1; 4 orientations	124	16	0.9063	0.8857
GLCM-16 d=1; 4 orientations	118	17	0.9037	0.8741
LBP (r=1,p=8) (r=2, p=8)	159	33	0.9130	0.8281
LBPri	150	29	0.9111	0.8380
DH d=27; 0 deg. d=1; 90 degree	**292**	**6**	**0.9722**	**0.9799**

TABLE V. DISCRİMİNATİON RESULTS FOR DATASET-III (SEPARATE TRAİNİNG SET)

Algorithm	Defects	False +	Accuracy	Precision
	Max 361	*Max 2339*		
GLCM-256 d=1; 4 orientations	67	21	0.8871	0.7614
GLCM-16 d=1; 4 orientations	68	20	0.8886	0.7727
LBP (r=1,p=8) (r=2, p=8)	66	22	0.8856	0.7500
LBPri	69	25	0.8856	0.7340
DH d=27; 0 deg. d=1; 90 degree	**141**	**11**	**0.9508**	**0.9276**

VIII. Complexity Measurements

In order to measure the computational complexity of each method, a C++ program was written. Measuring the processing time per one image is not possible on today's fast computers. To get an accurate measurement of how long each method takes to compute, 10 images were processed 500 times and then the total duration was divided to 5000. The computer was programmed to run a loop 5000 times, processing one of the 10 images in each iteration. This was done to prevent the CPU from copying an entire image in its cache. Processing the same image from the cache repeatedly would yield wrong results. Using The Microsoft Visual Studio compiler, with full optimization, measurements show that the calculation of the GLCM in one direction using 16 grey levels takes approximately the same amount of time with calculating the difference histogram in one direction with the same grey levels. Calculation of the difference histogram takes %98.5 of that of the GLCM. A few of the features that Haralick proposed were also implemented. The time required for their completion was negligible compared to the time required for the computation of a 16×16 GLCM. Over all, it seems that it takes %3-5 percent longer for the GLCM and its features to be calculated. However, for 256 grey levels, the GLCM becomes 256×256, making the calculation time for the features far longer. In Matlab, overall time for the calculation of 256 grey level GLCM takes 32 times longer compared to 16 grey level GLCM.

In order to compute the speed of LBP, a pure Matlab code is used. LBP (R=1, 2 p=8) took %43 longer to calculate than the GLCM-16 (Matlab code) in one direction. Therefore, it is computationally more efficient than the GLCM because it is more successful than the GLCM calculated in four directions. LBPri is % 419 slower than the GLCM-16 in one direction therefore it is only slightly more efficient than the GLCM. However, it should be noted that the LBP is cheating by using a gigantic; 16 MB look up table to speed up its calculations. It is essentially creating a processor inside the memory. Based on these numbers, Diftogram is computationally more efficient than both of the LBP algorithms are. According to a recent comparative study [26], among the LBP algorithms, LBPri is the fastest among all of them and it is only %15 less successful than the most successful variants such as the MRELBP [27]. Considering that the most successful ones are at least five times slower than the LBPri, (MRELBP is 9 times slower) it is seems that Diftogram is still computationally more efficient, even if it is not as reliable as the newest versions of the LBP.

In both of the previous experiments, GLCM was computed in four directions. On the other hand, based on our proposed method, the difference matrix (DM) was calculated in only two directions. Therefore, the proposed method is not only more successful at recognizing the defects, but also twice faster against GLCM-16.

IX. Results and Conclusion

In this article, two new approaches were proposed. The first one suggests that the use of Unser's difference histograms as input to a classifier yields better results at discriminating the texture than Haralick's method. Usually texture discrimination is a far more challenging problem then the texture classification; therefore, by using this method, similar superior results should be expected for classification as well. The second approach is that the orientation and the pixel distance to calculate the second order histograms must be selected based on the frequency properties of the texture. Two methods are proposed for the selection of these parameters. Another suggestion is the use of pixel average as an extra feature for more challenging patterns. Based on this design, three experiments are conducted.

Many of the defects used in the experiments are so faint that the human eye can barely notice them. In each of the experiments, the proposed set of methods performed approximately %250-%300 better at discriminating the texture, compared to the GLCM method with 19 features. The C++ implementation of the proposed algorithm, based on the parameters used for these experiments, should be approximately twice faster than the GLCM-16, which means, at least %500 improvement in efficiency. Against the LBP variants, the Diftogram is almost %200 more successful. The computational efficiency of the rotation variant plain LBP is comparable to that of the Diftogram while both the success rate and the efficiency of the LBPri is less.

Majority of the recent articles study rotation invariant texture recognition and discrimination algorithms. Therefore, the improvements discussed here might seem not very important for many academicians. However, engineering is performed to solve the problems of humanity, not the ones that we create in our own labs. Considering the fact that, every second, quality inspection is being performed on thousands of factories across the planet, hopefully, the improvements proposed in this article will find much greater use than many others published in the literature.

Acknowledgment

This study was financially supported by Adana Alparslan Türkeş Science and Technology University Scientific Research Coordination Unit. Project number: 16103010.

References

[1] R. M. Haralick, K. Shanmugam, I. Dinstein, “Textural Features for Image Classification,” IEEE Trans. Syst., Man, Cybern., vol. SMC-3, No 6, pp. 610-621, 1973.

[2] M. Unser, “Sum and Difference Histograms for Texture Classification,” in IEEE Trans. Pattern Anal. Machine Intell. Vol. 8, No.1 1986, pp. 118-125.

[3] T. Ojala, M. Piatikainen, T. Maenpaa, “Multiresolution grey-scale and rotation invariant texture classification with local binary patterns,” IEEE Transactions on Pattern Analysis and Machine Intelligence vol.24, no:7, 2002 pp 971–987.

[4] Abuhussein, Mohammed, and Aaron Robinson. “Obscurant Segmentation in Long Wave Infrared Images Using GLCM Textures.” Journal of Imaging 8.10 (2022): 266.

[5] Teo, Hong Chun, et al. “A review of the automated timber defect identification approach.” International Journal of Electrical and Computer Engineering 13.2 (2023): 2156.

[6] Meeradevi, T., et al. “An analytical survey of textile fabric defect and shade variation detection system using image processing.” Multimedia Tools and Applications (2022): 1-30.

[7] Tang, Bo, et al. “Review of surface defect detection of steel products based on machine vision.” IET Image Processing (2022).

[8] I. C. Baykal, G. A. Jullien "In-Camera Detection Of Fabric Defects" IEEE International Symposium on Circuits and Systems (ISCAS) 2004.

[9] Wagner T. (1999) Chapter 12 Texture Analysis. In: B. Jahne, H. Hauβecker, P. Geiβler(ed), Handbook of Computer Vision and Applications, vol 2, Academic Press, pp 276-307.

[10] Schramm, U., (1994). Automatische Oberflachenprufung mit neuronalen Netzen. Stuttgart: IRB-Verlag.

[11] Chen, Y., Nixon, M., and Thomas, D., "Statistical geometrical features for texture classification." Pattern Recognition, 28(4): 1995, pp.537-552.

[12] Fogel, I. and Sagi, D., "Gabor Filters as texture discriminator," Biological Cybernetics, no: 61, 1989, pp103-113.

[13] Laws, K. I., Textured Image Segmentation. PhD thesis, Faculty of the Graduate School, University of Southern California. 1980.

[14] Galloway, M. M., Texture analysis using gray level run lengths, Computer Graphics and Image Processing, no: 4, 1975, pp. 172-179.

[15] Maria Petrou, Alireza Talebpour, Alexander Kadyrov. "Reverse engineering the way humans rank textures" Pattern Anal. Applic. Vol. 10, No 2, 2007, pp. 101–114.

[16] E. Gonzalez-Rufino, P.Carrion, E.Cernadas, M.Fernandez-Delgado, R.Dominguez-Petit, "Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fishovary," Pattern Recogniton, no 46, 2013, pp: 2391-2407.

[17] L.H. Siew, R.M. Hodgson, E.J. Wood, "Texture measures for carpet wear assessment," IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (1) 1988, pp: 92–104.

[18] Sklansky, J, "Image segmentation and feature extraction," IEEE transactions on Systems, Man, and Cybernetics, SMC-8, 1978, pp 237-247.

[19] I. C. Baykal, "Performance Comparison of Texture Classifiers on Small Windows" IEEE Internatioanal Atificial Intelligence and Data Processing Symposium (IDAP) 2019.

[20] M. A. Selver, V. Avşar, H. Ozdemir, "Textural fabric defect detection using statistical texture transformations and gradient search," in The Journal of The Textile Institute, 2014, 105:9, pp. 998-1007.

[21] A. Latif-Amet, A. Ertuzun, A. Ercil, "An efficient method for texture defect detection: sub-band domain co-occurrence matrices," Image and Vision Computing, Elsevier, no:18 2000, pp: 543-553.

[22] A. Chondronasios, I. Popov, I. Jordanov, "Feature selection for surface defect classification of extruded aluminum profiles," in Int. J. Adv. Manuf. Technol. No 83, 2016, pp 33-41.

[23] A. Georgieva, I. Jordanov. "Intelligent Visual Recognition and Classification of Cork Tiles with Neural Networks," IEEE Transactions On Neural Networks, Vol. 20, No. 4, 2009, pp. 675-685.

[24] D A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," Can. J. Remote Sensing, vol. 28, no. 1, pp. 45-62, 2002.

[25] L. Soh and C. Tsatsoulis, "Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-Occurrence Matrices," IEEE Transactions on Geoscience and Remote Sensing, vol. 37, no. 2, 1999.

[26] L. Liu, P. Fieguth, Y. Guo, X. Wang, M. Pietikainen, "Local binary features for texture classification: Taxonomy and experimental study," Pattern Recognition, no:62 pp 135-160, 2017.

[27] L. Liu,S.Lao,P.Fieguth,Y.Guo,X.Wang,M.Pietikainen, "Median robust extended local binary pattern for texture classification," IEEETrans. Image Process. 25(3) pp:1368–1381, 2016.

An Autonomous Multi-agent Framework using Quality of Service to Prevent Service Level Agreement Violations in Cloud Environment

Jaspal Singh*, Major Singh Goraya
Department of Computer Science and Engineering-Sant Longowal Institute of Engineering and Technology,
Deemed University, Longowal, Sangrur, Punjab, India

Abstract—Cloud is a specialized computing technology accommodating several million users to provide seamless services via the internet. The extension of this reverenced technology is growing abruptly with the increase in the number of users. One of the major issues with the cloud is that it receives a huge volume of workloads requesting resources to complete their executions. While executing these workloads, the cloud suffers from the issue of service level agreement (SLA) violations which impacts the performance and reputation of the cloud. Therefore, there is a requirement for an effective design that supports faster and optimal execution of workloads without any violation of SLA. To fill this gap, this article proposes an automatic multi-agent framework that ensures the minimization of the SLA violation rate in workload execution. The proposed framework includes seven major agents such as user agent, system agent, negotiator agent, coordinator agent, monitoring agent, arbitrator agent and the history agent. All these agents work cooperatively to enable the effective execution of workloads irrespective of their dynamic nature. With effective execution of workloads, the proposed model also resulted in an advantage of minimized energy consumption in data centres. The inclusion of a history agent within the framework enabled the model to predict future requirements based on the records of resource utilization. The proposed model followed the Poisson distribution to generate random numbers that are further used for evaluation purposes. The simulations of the model proved that model is more reliable in reducing SLA violations compared to the existing works. The proposed method resulted in an average SLA violation rate of 55.71% for 1200 workloads and resulted in an average energy consumption of 47.84kWh for 1500 workloads.

Keywords—Cloud computing; multi-agent framework; SLA violations; energy consumption; history agent; Poisson distribution

I. INTRODUCTION

Cloud computing is a well-established paradigm that offers computing resources and services in a pay-as-you-go fashion to all the users connected to it [1]. It also provides resources to users that can be fully controlled by the users themselves through the virtualization of resources [2]. The cloud paradigm can be generally categorized as a scalable architecture that supports the inheritance of a wide range of technologies including utility computing, service-oriented architecture (SOA), and virtualization [3]. This paradigm also provides a shared pool of resources that offers services to workloads belonging to diverse forms of applications. Virtualized IT resources offer services of three types including software as a service (SaaS), Platform as a service (PaaS), and infrastructure as a service (IaaS) [4, 5]. With deep-spread data centers, the cloud paradigm ensures seamless services to its end users. Most of the popular organizations and companies are currently rendering cloud services to their customers and some of them include Google, Amazon, and Microsoft [6, 7]. The main acceptance of this paradigm is due to the flexible services offered where the users are requested to pay only for the services they have used [8].

The daily needs of the general community are satisfied with the cloud computing service which stays at a basic level of the computing paradigm [9]. Such a computing facility is specifically introduced to provide quality of service (QoS) aware services to a market of users to meet their objectives and requirements [10]. Thus, the service level agreement (SLA) oriented resource management is a crucial need for the users that negotiate a pile of virtualized and inter-connected systems between the users and cloud service providers or between the resource providers and brokers [11]. Due to the widespread availability of business models, it becomes a highly complex issue to select the appropriate service provider that can fulfil the execution of an application by meeting its QoS requirements [12]. A system-centric resource management framework is usually employed by cloud providers to offer computing services and resources [13]. A market-oriented resource management framework is of utmost need to enable the supply and demand of resources thereby offering feedback to both providers and consumers in terms of economic incentives [14]. Also, based on the usage of resources and services, the service requests are distinguished through QoS-based resource allocation [15].

Currently, the cloud paradigm provides only limited support for dynamic SLA negotiations between the associated participants such as cloud service providers and consumers [16]. Also, there are no reliable mechanisms that can offer automatic resource allocation to multiple competing requests [17]. The existing frameworks are unable to completely support customer-driven service management with the requested service requirements and customer profiles [18]. The SLAs that are signed between the cloud customers and cloud service providers are required to be maintained on each call of request processing and executions. Generally, market-based resource management strategies are more focused on customer satisfaction and service provider profits [19, 20]. Therefore, the development of a framework that can satisfy both the service providers and customers is of utmost need [21]. In most of the research works conducted, it has been concluded that it is almost impossible to extract appropriate market-based resource management schemes that can encompass both computational risk management and user-driven service management to sustain the SLA-aware allocation of resources [22, 23].

The SLA-oriented schemes are required to offer personalized attention to customers to help them meet their SLA-aware objectives [24]. One of the most important factors to be considered while designing such a solution is that the demands of the users fluctuate with time for the changes encountered in the operating environment and business operations [25]. SLA can be defined as a formal agreement that provides information regarding the quality of every non-functional requirement (NFR) of a service [26]. A formal procedure is followed in cloud computing that if there is any SLA violation encountered in the workload execution process, then penalties are provided to the service providers [27]. When there are no violations of SLA for different workload executions, then rewards are provided either to the customers or service providers after evaluations [28]. One of the major problems arising here is with the dynamic execution of workloads where there are a huge number of workloads arriving in the cloud for executions. At this point, the QoS cannot be assured in every circumstance and there is a requirement for an automated system that can accurately monitor the violations occurring within the environment [29, 30]. Therefore, there is a leading requirement for an automatic system that can control and monitor the QoS of the workloads within the negotiated terms.

A. Motivation

There are several techniques encountered to automate the process of resource management via SLA negotiation. Generally, those methodologies integrate virtualization and market-based allocation policies for allocating the cloud resources to workloads to complete executions. Several efforts have been made to automate the process of SLA-aware resource allocation to the workloads. Some methods focused on framing SLA to workload execution through the negotiation process whereas others focused on automating the entire process. But, only a few methods explored the benefits of multiple agents in the cloud to enable the automatic management of resources to support SLA-aware workload execution. Therefore, there is a need for such a technique to be enforced to avoid SLA violations while executing the workloads. Therefore, this paper presents an automatic multi-agent framework that supports the execution of workloads without any violation of SLA. Moreover, the proposed framework also optimizes energy consumption in data centers to enhance overall performance.

B. Contribution

The major contributions of the proposed work include the following:

- A new and efficient multi-agent system is proposed in this work to enable seamless services to its users by satisfying their fluctuating demands and enabling SLA-aware executions of workloads.

- Presenting the agent-based cloud framework where each of the agents is incorporated to provide timely execution of workloads without disturbing the SLAs. Moreover, the framework is designed in a unique way to satisfy both the service providers and the customers involved.

- Introducing an additional history agent within the agent-based framework to keep track of the resources used and the requests processed. The aim of adding this agent is to enable the prediction of future demands so that the overall efficiency and reputation of the system can be enhanced.

- Introducing the Poisson distribution function (PDF) model to generate random numbers based on the input to form the dataset. The generated dataset is then provided to the proposed model to evaluate and compare the model extensively.

C. Organization

The remainder of the paper is structured as per the following: Section II presents the literary works established by other researchers working in the same field, Section III provides the proposed methodology with architectures and explanations, Section IV provides the results and discussion with comparative analysis and Section V concludes the paper with future scopes.

II. Background on Resource Provisioning in Cloud Computing and QoS Constraints

Some of the recent works established for controlling SLA violations in the cloud are reviewed below:

Cloud computing technology faces several challenges among which SLA violation is one of the most common and tiring problems affecting its overall performance. In the cloud-based e-commerce negotiation framework, the optimization of broker negotiation strategy is a cumbersome task. Generally, long-term or pre-request optimizations are followed to resolve the task. The pre-request strategies focus on the usage of various utility functions and are followed in most research works. The long-term strategies are less focused and most of them are unable to guarantee negotiation and state-of-art to minimize SLA. Such limitation was addressed by Rajavel and Thangarathanam [31] effectively through the stochastic behavioral learning negotiation (SBLN) technique. The main intention of the technique was to maximize the success rate and utility value to a maximum level. The increase in the desired values was attained by increasing the count of negotiation rounds. The performance of the method was implemented and compared with other techniques and the outcomes proved its efficacy.

The Multiple agent-based systems were developed by Azhagu and Gnanasekar [32] to deal with the SLA violations in the cloud computing infrastructure. Violations of SLA affect the business operations of both the cloud service providers and customers as compensation is required to be provided by the service providers (CSP) for their customers. The agent-based model enhanced the trust of every stakeholder through the automatic minimization of SLA violations. The framework included a total of six agents a user agent, a system agent, a negotiation agent, a coordinating agent, a monitoring agent, and an arbitrator agent. The monitoring agent was responsible to monitor the cloud environment and indicated SLA violations. The arbitrator agent observed and identified the cause of the violation and posted penalties or rewards based on the performance. After evaluations of the entire framework, the outcomes suggested that the method was effective in controlling SLA violations in the cloud with the maximization of performance in workload executions.

The Discovery of cloud services is a highly challenging issue due to the increase in complexities and network size. With the dynamic increase of these two factors, the effective discovery of services is hampered making it an NP-hard problem. The popular cloud service discovery method based on ant colony optimization (ACO) suffered from load balancing issues. To resolve the issue and enable effective usage of resources, Heidari and Navimipour [33] introduced the inverted ACO (IACO) method that promised load-aware service discovery to the cloud. In the inverted algorithm, the attractive behavior of pheromones was replaced with the repulsive behavior. The model was simulated using the Cloudsim tool and the numerical results of the model proved its efficiency over the other compared methods. Also, the model provided several other benefits including energy efficiency, response time mitigation, and control of SLA violations.

Cloud computing supports large-scale processing in a distributed fashion with higher flexibility. SLA violations in the cloud occur due to several facts and it is important to control these violations to attain performance improvement. VM allocation is one of the common and challenging problems in the cloud resulting in SLA violations. Other problems associated with VM allocation include problems in asset utilization and energy consumption. An SLA-aware strategy to allocate the VMs in the cloud using an intelligent algorithm was introduced by Samriya et al. [34]. To attain the objective, the method utilized the multi-objective emperor penguin optimization (EPO) algorithm that allocated the VMs in a heterogeneous cloud environment. Further, simulations were conducted to prove the performance of the method compared with other multi-objective metaheuristic optimization algorithms. The outcomes proved that the model effectively reduced SLA violations and energy consumption in the cloud environment.

Another strategy based on resource allocation was introduced by Belgacem et al. [35] based on the exploration of properties of multiple agents in the cloud. Cloud infrastructure face challenges in resource allocation due to its heterogeneous nature, volatile resource usage, and accommodation of VMs with diverse specifications. The method introduced the combination of an intelligent multi-agent system with the reinforcement learning method (IMARM) to attain the objective of optimal resource allocation. The Q-learning process was combined with the properties of multiple agents to gain performance enhancement in resource allocation accordingly. IMARM method responded well to the fluctuating customer demands through dynamic allocation and release of resources accordingly. Moreover, the VMs were moved to the best state concerning the current state environment through the learning model. Finally, simulations were conducted to prove the performance improvement attained by the model compared to previous models in terms of various metrics.

In both cloud and utility-based computing platforms, SLA emerge as a chief aspect while providing personalized services to the users. In order to offer flexible establishment of SLAs and to prevent SLA violations, Son and Jun [36] presented a proactive resource allocation (PRA) scheme. The presented scheme optimally selected a suitable datacenter among the available globally distributed datacenters to enhance resource allocation to the workloads. The method also provided time slots and price negotiations for flexible SLAs. The effectiveness of the method was proved through experiments.

TABLE I. COMPARATIVE ANALYSIS OF THE EXISTING LITERARY WORKS

Authors	Methods	Advantages	Drawbacks
Rajavel and Thangarathanam [31]	SBLN	Obtained drastic increase in success rate and utility value	The unwanted conflicts among the participants are required to be addressed
Azhagu and Gnanasekar [32]	Multi-agent system	Automatically controlled SLA violations in the cloud through continuous monitoring	More QoS parameters are required to be considered to attain optimal performance
Heidari and Navimipour [33]	IACO	Efficient discovery of cloud services with enhanced utilization of resources	The repulsive behavior of pheromones is required to be evaluated deeper to prove its advantages
Samaria et al. [34]	Multi-objective EPO	To enable effective VM allocation in the cloud with minimized SLA violation and energy consumption	Other important performance objectives such as wastage of resources are required to be focused
Belgacem et al. [35]	IMARM	Dynamic allocation and release of resources and providing better responses to the customers' changing demands	More metrics are required to be considered and more analyses are required to prove its reliability
Son and Jun [36]	PRA	The overall efficiency of negotiation and utility has been increased with the trade-off algorithm	Limited SLA options are provided by the framework based on enforced SLA strategies
Wu et al. [37]	PURS	Facilitated intelligent bilateral bargaining of SLAs and provided maximum profit for the brokers through enhanced customer satisfaction levels	The penalty for the failure of negotiation from the user's perspective is not considered

Another SLA negotiation framework was introduced by Wu et al. [37] to accomplish profit with higher customer satisfaction. The process of negotiation establishment becomes tough with the existence of multiple CSPs. The introduced framework considered SaaS broker as a one-stop-shop for the customers and negotiation was performed with multiple CSPs. The automated framework supported bilateral bargaining of SLAs and helped in maximizing the profit of brokers. Extensive evaluations with real CSP proved the efficacy of the method. Table I presents a comparative analysis of the existing literary works.

A. Problem Statement

On reviewing the existing works, it has been identified that the multi-agent system in the cloud is highly advantageous and helps to offer numerous reliable services to its customers. The Multi-agent-based framework is one of the effective methods to enable the execution of workloads without any violation of the SLA constraints. The existing methodologies are unable to completely enable the execution of workloads within the defined deadlines. Other agent-based frameworks are merely unstable as the failure of negotiation is not given importance or considered that may result in performance degradation. Moreover, the negotiations terms and conditions are not well-established in most of the existing works. Apart from these, the demands of the future workloads are unidentified which delays processing of workloads. Because of looking forward to attaining optimal resource provisioning using QoS in cloud computing and higher performance by satisfying the QoS constraints of users, a very few techniques are formulated based on the self-management of cloud services using multiple agents. Moreover, it is of utmost need to optimize the violations of SLA with the help of negotiation before the deployment of services in the cloud. To overcome the existing drawbacks and to fill the gaps, a new multi-agent-based framework is introduced based on the accommodation of multiple agents to monitor and complete the execution of workloads within the defined SLA. The proposed framework also utilizes an additional agent to back up the details regarding executions in order to identify the future demands for resources. By this way, the profit and rewards from both the ends can be considered and the effectiveness of negotiations can also be improved.

III. Proposed Methodology

Execution of cloud workloads within the defined deadlines is a complex task and requires appropriate algorithms and techniques. Efficient workload execution in cloud is highly crucial as it has wide range of applications supporting companies associated with it. The agent-based frameworks are faster in approaching the requests from users compared to other agentless frameworks. This helps to complete the workload execution within the deadlines. Moreover, these frameworks are capable of constantly monitoring the environment and collect data at real time. Due to these advantages, a new Autonomous Multi-agent-based framework based upon Probability and History (AMAPH) is designed in this work to prevent SLA violations and to attain higher performance in workload executions. The proposed multi-agent system monitors the cloud environment and checks for SLA violations. When there is no violation encountered in a workload execution, rewards are provided to the service provider or customer and when there is a violation, penalties are provided and the reason for the violation is determined. The proposed mechanism works deliberately to avoid any kind of SLA violation within the cloud environment and assures proper execution of workloads that are succeeded respectfully. Moreover, the History agent keeps a record of either successful or failed requests, respectfully. The overall architecture of the proposed work is displayed in Fig. 1.

The proposed multi-agent framework includes seven agents a user agent, a system agent, a negotiation agent, a history agent, a coordinating agent, a monitoring agent, and an arbitrator agent. The requests reach the user agent at the initial stage and then based on the type of request; it is forwarded to the system agent.

The type of service required for processing the request is determined and the request is forwarded to the negotiation agent where a negotiation process is initiated between the user agent and service provider. A service is selected for the request and the details are then forwarded to the coordinating agent. The history agent is responsible to track the services offered to the requests. SLA is established by the coordinator and the monitoring agent dynamically monitors the environment for any violation and each violation, an indication is sent to the arbitrator agent.

Finally, penalties are laid by the arbitrator to the service provider and the type and reasons for the violation are determined. The Poisson distribution function (PDF) component is included in the framework to test the performance of the proposed system (AMAPH) and assuring for the different number of workloads accordingly in comparison to the base paper [32], at a glance.

A. User Agent

The user agent is the initial agent of the proposed framework, and the role of the agent is to receive the requests provided by the associated cloud users. Thus, the cloud users directly request the services via the user agent to the cloud. This agent is responsible for dealing with the user registration processes for new users. Each user is linked with a single user agent to attain the cloud services.

For any kind of additional services requested by the user, multiple user agents are not created in the proposed work and the additional requested services are handled by the same agent. For registration of new users, the user agent gathers the required information such as the personal information of users via a registration form. In the case of service requests from the user side, the user agent determines the type of service being requested by the user. All the details regarding the service type requested are collected and analyzed and then the requests are forwarded to the system agent for further processing.

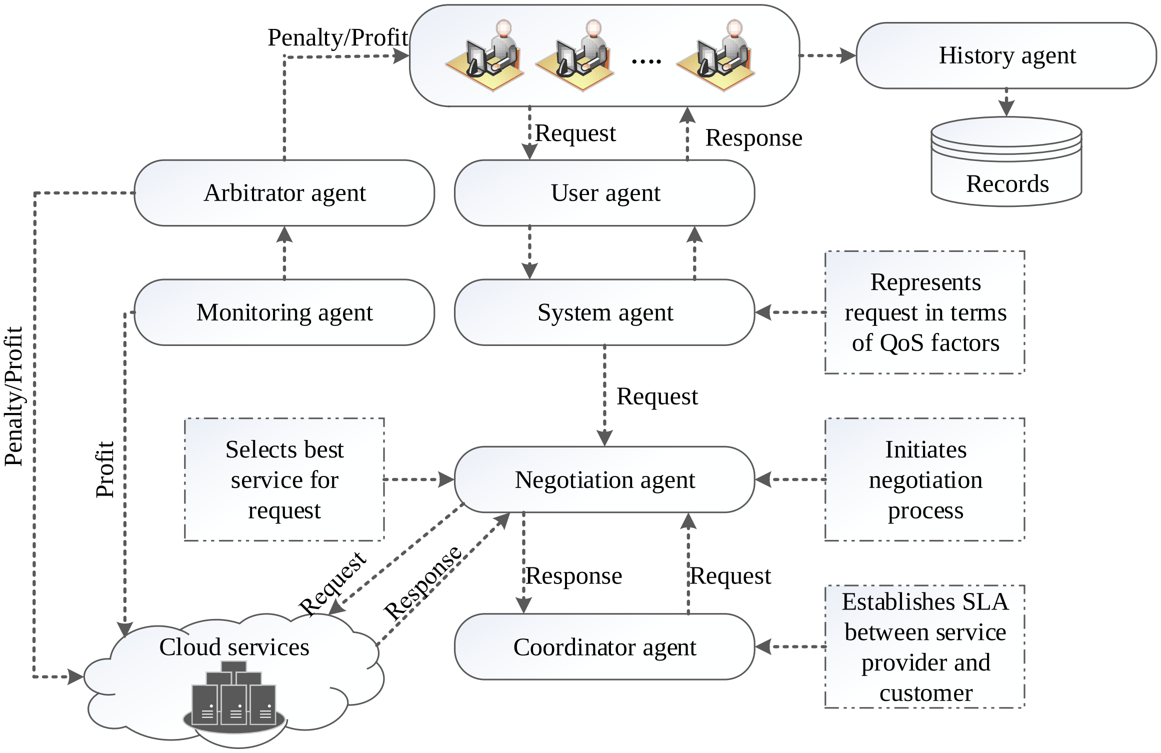


Fig. 1. Architecture of the proposed Multi-Agent framework (AMAPH).

B. *System Agent*

The system agent receives the requests from the user agent and determines the actions to be taken further. The details regarding the requests are obtained and then the requests are represented in technical terms including the quality of service (QoS) factors such as account type, number of accounts, contract length, solution time, and response time [32]. The system agent is responsible for verifying the quality factors of all the incoming requests and helps the framework to better process the requests. After representing the requests in terms of quality factors, the service type requested by the user is identified. Based on the requested service type, the system agent either forward it to the negotiation or the coordinator agent.

C. *Negotiation Agent*

This agent is responsible for initiating the negotiation process between the user agent and the service provider. The negotiation process is established based on diverse technical factors including nature of service, reliability, response time, monitoring, reporting of service, and responsibilities. Based on the technical factors and the service type being requested by the user, the negotiation agent communicates with the available service providers. The available service providers on the other side, place bids in the given view of processing the requests based on the available resources, resource capabilities, market circumstances, and business objectives. The main significance of the proposed framework is that the negotiation agent broadcasts the request details to all the service providers to provide the best service to the requests. The negotiation process ensures maintaining a more feasible SLA in workload executions. Based on the requested details available, the negotiation agent evaluates the received bids from service providers. Then, the attributes of service providers are compared with the resource requirements of the user and the appropriate service provider has selected that best suit the request. The details regarding the selected service and the service provider are then forwarded to the coordinator for further processing. Further, the history agent shall maintain the state-of-art in records wherein the id of the user, the CSP being selected by the negotiator as optimal resource provisioning process using QoS with respect to different data centres.

D. *Coordinator Agent*

The coordinator agent receives the request and selected service details from the negotiation agent and evaluates the request. The agents evaluate the received request for first-time access or request for service upgradation. After analyzing the type of request received, appropriate actions are taken further. The agent also formally establishes an SLA between the respective user and service provider and the message is forwarded to both parties. Apart from sending the message, it is also preserved by the agent for enforcement. Finally, the SLA is sent to the monitoring agent for further effective actions.

E. *Monitoring Agent*

The main responsibility of the monitoring agent is to continuously monitor for SLA violations within the cloud environment. Based on the established SLA details received from the coordinating agent, the monitoring process is regulated by the agent. When a violation is encountered in the environment, the agent immediately sends an indication to the arbitrator to take appropriate actions or to provide a penalty to the respective party. If there is no violation in the workload execution, then the monitoring agent sends an indication about providing a profit message to the respective party for the successful execution of the task. It recommends the arbitrator provide rewards to the concerned service provider.

F. Arbitrator Agent

This agent is responsible to analyze the type of violation that has occurred and the reasons behind the occurrence of such violation. Then, based on the analysis, penalties are enforced on the service providers or the respective customers concerning the defined SLAs.

G. History Agent

The history agent is one of the significant agents in the proposed work that keeps track of service usage and workload executions. This helps the system to predict future workload requests and the type of services that could be predicted by those requests. The history agent maintains records where the id of the user, the type of service requested, the service being selected by the negotiator as optimal resource provisioning, and the service provider allocated to process the requests are stored as files. Based on these details, the agent predicts future workload requests and the type of service needed to process the request. By predicting these parameters, the proposed system decides on faster workload executions with minimized SLA violations. The history agent keeps a record wherein the service provider allocated to the request and other constraints are stored as files.

H. Poisson Distribution for Random Number Generation

The Poisson distribution function (PDF) is followed in this work for random number generation and these numbers are then given to the model for evaluation purposes. This distribution has very minimum parameters and is very simple to implement. Therefore, this distribution is chosen in our work to reduce the complexities. Consider a discrete random variable χ and it is assumed to follow a Poisson distribution with parameter $\lambda > 0$ if and only if it follows the following probability mass function:

$$f(k;\lambda) = \Pr(\chi = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (1)$$

where, k specifies the count of occurrences, e is the Euler number, λ specifies the positive real number which is equal to the expected value and variance of the random variable χ.

$$\lambda = \mathrm{E}(\chi) = \mathrm{var}(\chi) \quad (2)$$

This distribution can be generally followed in systems with a large number of rare but possible events and the count of such events within a fixed time interval can be specified as a random number with the Poisson distribution. Instead of knowing its value of λ it the system provides the value of the average rate δ, then λ is substituted by δt an Eq. (1) that can be adopted as follows:

$$\mathrm{P}(k \ events \ within \ \mathrm{int} \ t) = \frac{(\delta t)^k e^{-\delta t}}{k!} \quad (3)$$

IV. Results and Discussion

A detailed analysis of the results obtained through evaluations of the proposed framework is presented in this section. The entire simulations of the proposed work have been carried out using the CloudSim tool with the Java Agent Development Environment (JADE). The proposed system includes multiple active agents such as a user agent, system agent, negotiation agent, history agent, coordinator agent, monitoring agent, and arbitrator agent. The user agents receive the requests and provide a registration form if it is a new request or evaluates and forwards the details to the system agent. The system agent evaluates the requests and represents them in technical form and then forwards it to the negotiation agent where the negotiation process is initiated. The history agent is responsible to keep track of certain important records and the coordinator agent chooses the appropriate service to process the request based on the SLAs. The monitoring agent monitors the entire cloud environment for SLA violations and if there is any violation, then the agent sends an indication to the arbitrator for penalty enforcement or directly forwards a message for rewards if there is no violation. The proposed framework is autonomic and automatically monitors and controls the environment without the need for the intervention of a cloud engineer.

In the JADE environment and evaluations, the overall framework is implemented as agents and the random numbers are generated via the Probability Distribution Function for realistic datasets that are exchanged between agents for collaboration. Apart from the seven agents of the framework, the environment also accommodated a resource manager, cloud broker, VM manager, physical machine manager, and cloud registry. The entire simulations are carried out with a total of 05 data centers and 20 service providers. In the simulations, the requests arising from the VMs are forwarded to the service broker. The workloads are simulated based on the business workload traces provided by GWA-T-12 Bitbrains. The simulated dataset includes the performance values for different VMs running in datacenters and the data are recorded in .CSV files. The data values are generated for 5 datacenters to provide extensive evaluations and analysis. The generated dataset included performance values such as CPU usage, memory usage, network throughput, disk throughput, CPU capacity provisioned and memory capacity provisioned. A total of 1500 workloads are generated to evaluate the proposed system and each workload included the above-mentioned performance values. Moreover, the dataset consisted of no missing or duplicate values and this reduced the need for preprocessing. For comparison, the proposed method selected PRA [36], PURS [37], and a multi-agent system [32]. All these results are respectfully taken from the multi-agent system [32] for comparison; the results are undertaken by varying the number of workloads, resources, and execution times for optimal resource provisioning using QoS in cloud computing.

A. *Performance Metrics*

The proposed framework has been evaluated in terms of SLA violation rate and energy consumption [32]. The mathematical representations and descriptions of the metrics chosen are as follows:

SLA violation rate: SLA violation rate indicates the rate of violations occurring in the environment for different workload executions. The mathematical representation for the SLA violation rate can be given as follows:

$$S_{VR} = f_R * SLA_W \tag{4}$$

where, f_R is the failure rate and SLA_W is the weight of SLA. The failure rate can be measured using the following formulation:

$$f_R = \frac{W_{fR}}{W_{total}} \tag{5}$$

wherein, W_{fR} is the workloads' failure rate and W_{total} indicates the total count of workloads involved. The SLA violation rate is taken by varying the number of workloads, number of resources, and the number of execution times which is given by the following formulation:

$$E_T = \frac{W_{ct} - W_{st}}{W_{total}} \tag{6}$$

where, W_{ct} is the completion time of workload, and W_{st} is the submission time of workload.

Energy consumption: Energy consumption indicates the consumption of energy by the VM to complete the execution of a workload. The mathematical formulation is as follows:

$$E_C = (l * \max) + (1 - l) * \max E_{VM} \tag{7}$$

wherein, E_{VM} indicates the energy consumed by VM and l indicates the constant set to 0.5 in simulations.

B. *Performance Analysis*

The overall performance of the proposed framework is analyzed in this section. The simulations are performed with user-defined QoS constraints like CPU, RAM etc. All the results obtained are compared with the methods such as PRA [36], PURS [37], and multi-agent system [32]. The existing methodologies also follow the same configurations and parameter settings. The analysis of the obtained results is presented:

The results of the SLA violation rate for the different workloads are recorded. The performance of the proposed method is more optimal than the other methods. The addition of a history agent helped the model to accurately predict future workloads so that the SLA violation rates are reduced. The results are taken by varying the number of workloads from 0 to 1200. For all the workload input, the proposed model maintained higher performance compared to the other models. When the number of workloads is low, the violation rate is also low and when the workload is increased, the violation rate is scanty also and gradually increased. The performance comparison of SLA violation rate with respect to number of workloads is presented in Table II. A graphical representation of the results is presented in Fig. 2. The figure shows that there is only a minimal increase in violation rate for the proposed method showing its efficacy. The graph also shows that there is a huge impact on the overall performance of the framework when the number of workloads are varied. The proposed approach depicts a result of 19.5% violation rate when the number of workloads is 200 and resulted in 55.71% violation rate when the number of workloads is increased to 1200. Among the compared methods, the multi-agent system resulted in better performance compared to PRA and PURS and other details of QoS [32].

TABLE II. PERFORMANCE VALUES OF SLA VIOLATION RATE VS. NUMBER OF WORKLOADS

Methods	**Workloads**					
	200	**400**	**600**	**800**	**1000**	**1200**
PRA	41.81	46.88	47.11	57.99	69.06	79.35
PURS	32.52	37.2	42.46	50.44	62.28	69.48
Multi-agent system	23.35	28.3	36.64	45.38	56.52	59.47
Proposed	**19.5**	**26.79**	**32.03**	**38.41**	**46.21**	**55.71**

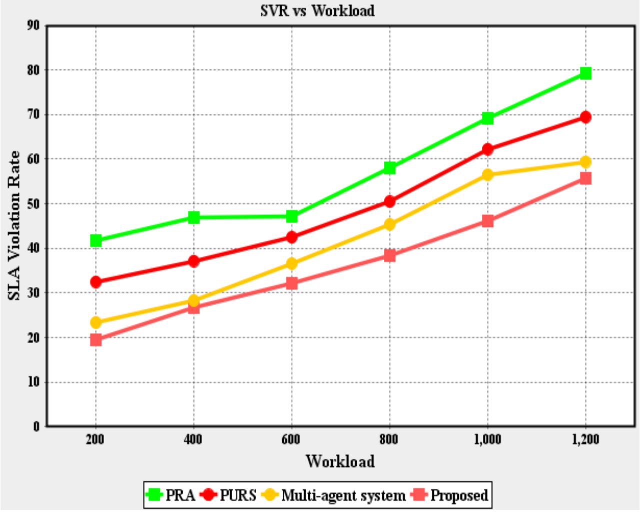


Fig. 2. Graphical representation of SLA violation rate vs number of workloads.

TABLE III. PERFORMANCE VALUES OF SLA VIOLATION RATE VS. NUMBER OF RESOURCES

Methods	**Resources**					
	50	**100**	**150**	**200**	**250**	**300**
PRA	90.91	166.51	195.22	235.41	310.05	354.07
PURS	81.34	145.45	157.89	220.1	282.3	309.09
Multi-agent system	64.72	113.08	140.76	197.22	269.85	290.35
Proposed	**48.29**	**91.27**	**110.91**	**134.88**	**164.11**	**199.77**

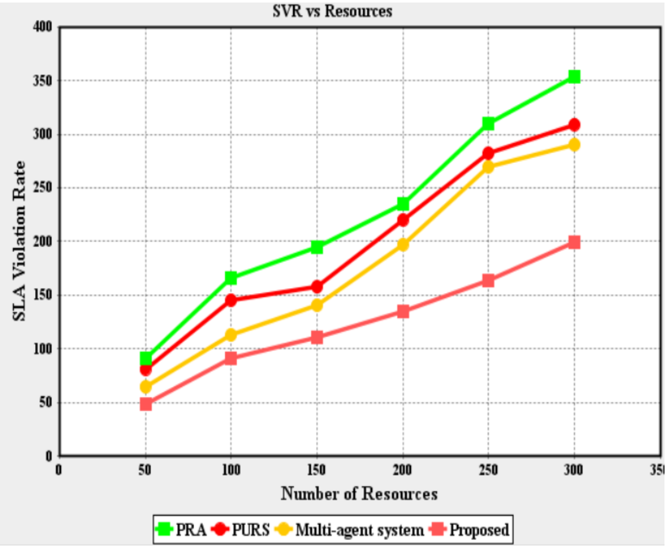


Fig. 3. Graphical representation of SLA violation rate vs number of resources.

The results of the SLA violation rate based on the number of resources are presented. The proposed technique is more optimal in resource provisioning using QoS than the existing methods in reducing SLA violations accordingly. The number of resources used for executing the workloads has a major impact on the variations in SLA violations. When the number of resources used in execution is less, the SLA violation rate is low significantly and when the resource is increased, the violation rate is also increased gradually in a scanty manner. This is because of the increase in the number of resources required to handle more workloads that results in increased SLA violations. This is also plotted in the graphical representation shown in Fig. 3. The values obtained on comparison of SLA violation rate with respect to number of resources are shown in Table III. The proposed method resulted in 48.29% of violations whereas for a total of 300 resources of violation rate could be affirmed accordingly. Among the compared techniques, the multi-agent system yielded better results.

The results of energy consumption for the different workloads are presented. The proposed method consumed less energy as compared to other methods in workload execution. When there is a minimum number of workloads, the energy consumption is less, and it increases gradually with the scanty increase in the number of workloads as per the provisioning of resources using QoS. This is also shown in the graphical representation presented in Fig. 4. The values obtained for energy consumption comparison are presented in Table IV. For 250 workloads, the energy consumed by the proposed method is 17.67kWh and for 1500 workloads, the energy consumed is 47.84kWh. Among the compared techniques, the multi-agent system consumed less energy to execute the workloads, and the other two methods consumed more energy for executions.

The results of the SLA violation rate for different execution times are recorded. The proposed method is more optimal than the existing methods. The values obtained on comparison of SLA violation rate with respect to execution time are shown in Table V. The graphical representation of the SLA violation rate based on execution times is shown in Fig. 5. The figure shows that the SLA violation rate is low for smaller execution times and gradually increases with the increase in execution times. Also, the proposed multi-agent system (AMAPH) produced better results as being compared to existing methods [32].

TABLE IV. PERFORMANCE VALUES OF ENERGY CONSUMPTION VS. NUMBER OF WORKLOADS

Methods	Workloads					
	250	500	750	1000	1250	1500
PRA	35.87	41.12	52.72	54.79	73.8	72.7
PURS	26.18	36.13	48.11	47.74	63.78	64.52
Multi-agent system	20.63	26.9	38.03	42.94	56.21	59.17
Proposed	**17.67**	**23.38**	**27.87**	**33.31**	**39.89**	**47.84**

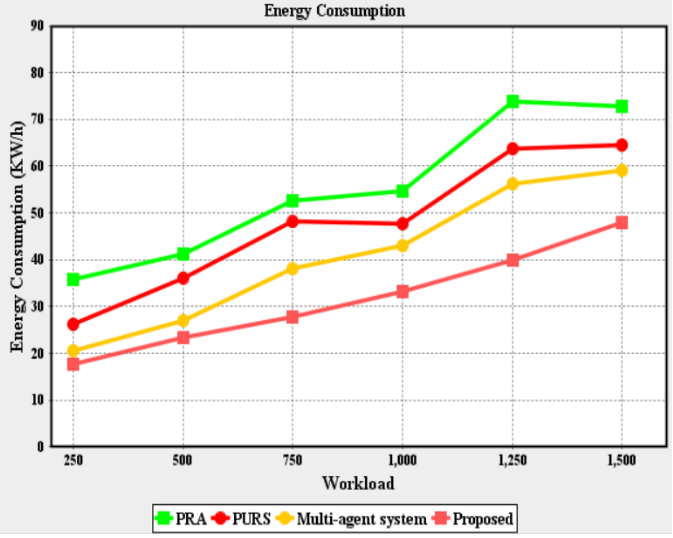


Fig. 4. Graphical representation of energy consumption vs number of workloads.

TABLE V. PERFORMANCE VALUES OF SLA VIOLATION RATE VS. EXECUTION TIME

Methods	Execution time						
	10	20	30	40	50	60	70
PRA	235.29	378.52	421.99	508.95	718.67	780.05	813.3
PURS	212.28	273.66	327.37	473.15	682.26	726.34	780.05
Multi-agent system	179.1	236.32	293.53	420.4	589.55	701.49	694.03
Proposed	**93**	**114.05**	**138.28**	**176.84**	**218.99**	**300.92**	**363.3**

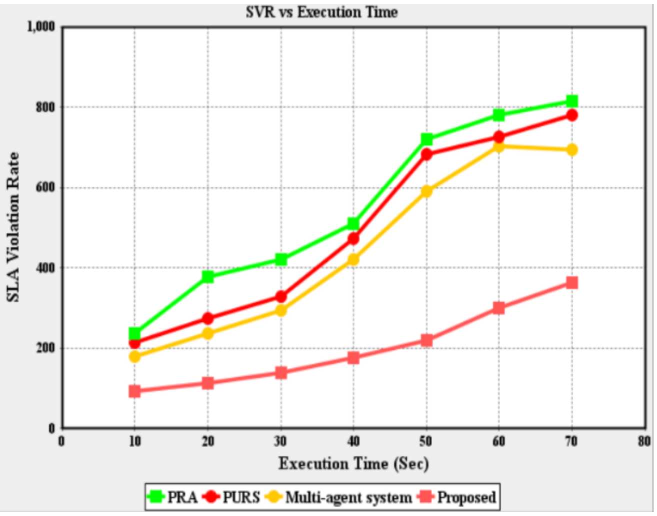


Fig. 5. Graphical representation of SLA violation rate vs execution time.

The overall simulations suggested that the proposed framework of resource provisioning using QoS is more optimal than the other compared techniques in reducing the SLA violations occurring in the cloud environment. As per the above results, the simulation build time has been considered accordingly but it may have a little change as in ground level of implications. The violation rate generally increases when the number of executions is increased. With the increase in the number of workload executions, the performance of the cloud network slows down due to higher energy consumption and increased violation rates as different case studies analyses for 05 different data centers. Therefore, the proposed framework is presented that is highly optimized for reducing the SLA violation rate and energy consumption in data centers. The performance of the framework is analyzed by varying the number of resources and workloads as these are the major factors influencing the performance. The results also proved that SLA violations increase when the number of executions needed is increased. While dealing with more workloads, the elasticity of the cloud and the resource availability is required to be regularly maintained to reduce SLA violations. The proposed framework of modelling and simulation measures the Quality of Service (QoS) and performance in Data-Center along with resource utilization policy. The analysis proved that the proposed model worked on reducing both the energy consumption in data centers and SLA violations in every dimension. The inclusion of a history agent within the architecture helped the model to forecast the arriving workloads and to predict the future requirement of resources. It kept track of the records of utilized resources and the available resources to maintain normal execution without any deviation in SLAs. The conjecture can be clarified for the response time as one of the major components of QoS factors based upon different workload's build time (Minimum is 21 sec and Maximum is 113 seconds). The method optimized the workload executions thereby reducing the overall violation rates and enhancing the overall cloud performance. Therefore, the proposed framework can be suggested as a promising tool to mitigate SLA violations and issues of higher energy consumption in cloud data centers and to achieve optimal performance for QoS as MOHFO and CGR analyses [15].

V. Conclusion

Cloud computing technology is one of the most popular computing technologies followed by most organizations throughout the world. This is because of its elastic and distributed nature that is capable of supporting faster network services with abundant provisioning of resources. In this work, a new and efficient framework is designed that supports the optimal execution of workloads with minimized SLA violations and energy consumption. The proposed framework includes multiple agents such as a user agent, system agent, negotiation agent, history agent, coordinator agent, monitoring agent, and arbitrator agent. The user agent obtains the request details from users and forwards them to the system agent where the technical terms of the requests are explored. The negotiation agent initiates the negotiation process between the service provider and the customer to avoid SLA violations. It selects the best service that can execute the current workload without SLA violation and with minimum consumption of energy. The history agent keeps track of workload executions to provide better forecasts of future executions. The coordinator agent receives the selected service details from the negotiator and establishes a formal SLA. The monitoring agent monitors the environment continuously for violations and sends an indication to the arbitrator if any violation is encountered. The arbitrator provides penalties or rewards to the service provider or customer and analyses the cause of the violation. The method is simulated and evaluated using a random number generated by Poisson distribution. The analysis proved that the method minimized the SLA violation rate and energy consumption in data centers much better compared to other existing techniques. Therefore, a resource provisioning framework using QoS attribute requirements to manage the resources of the Cloud while taking into account the Customer's Quality of Service as determined by the Service-Level Agreement (SLA) in the Cloud Computing environment has been incorporated successfully.

VI. Conflicts of Interest

The author declares no conflict of interest.

Author's Contribution

Conceptualization, Methodology, Software, Analysis, Resources & Investigations: Jaspal Singh. Supervision: Dr. Major Singh Goraya (my respected guide to PhD).

References

[1] K.S.S. Kumar, and N. Jaisankar, "An automated resource management framework for minimizing SLA violations and negotiation in collaborative cloud." International Journal of Cognitive Computing in Engineering vol. 1, pp. 27-35, 2020.

[2] S. Tuli, S.S. Gill, M. Xu, P. Garraghan, R. Bahsoon, S. Dustdar, R. Sakellariou et al., "HUNTER: AI based holistic resource management for sustainable cloud computing." Journal of Systems and Software vol. 184, pp. 111124, 2022.

[3] S.S. Gill, I. Chana, M. Singh, and R. Buyya, "RADAR: Self-configuring and self-healing in resource management for enhancing quality of cloud services." Concurrency and Computation: Practice and Experience vol. 31, no. 1, pp. e4834, 2019.

[4] M.A. Haghighi, M. Maeen, and M. Haghparast, "An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms." Wireless Personal Communications vol. 104, no. 4, pp. 1367-1391, 2019.

[5] Y. Jararweh, M.B. Issa, M. Daraghmeh, M. Al-Ayyoub, and M.A. Alsmirat, "Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation." Sustainable Computing: Informatics and Systems vol. 19, pp. 262-274, 2018.

[6] D. Saxena, A.K. Singh, and R. Buyya, "OP-MLB: An online VM prediction based multi-objective load balancing framework for resource management at cloud datacenter." IEEE Transactions on Cloud Computing 2021.

[7] S.S. Gill, S. Tuli, A.N. Toosi, F. Cuadrado, P. Garraghan, R. Bahsoon, H. Lutfiyya et al., "ThermoSim: Deep learning based framework for modeling and simulation of thermal-aware resource management for cloud computing environments." Journal of Systems and Software vol. 166, pp. 110596, 2020.

[8] N. Gholipour, E. Arianyan, and R. Buyya, "A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers." Simulation Modelling Practice and Theory vol. 104, pp. 102127, 2020.

[9] B.K. Dewangan, A. Agarwal, M. Venkatadri, and A. Pasricha, "Autonomic cloud resource management." In 2018 fifth international conference on parallel, distributed and grid computing (PDGC), IEEE, pp. 138-143, 2018.

[10] I. Odun-Ayo, B. Udemezue, and A. Kilanko, "Cloud service level agreements and resource management." Adv. Sci. Technol. Eng. Syst. vol. 4, no. 2, pp. 228-236, 2019.

[11] S. Mustafa, K. Sattar, J. Shuja, S. Sarwar, T. Maqsood, S.A. Madani, and S. Guizani, "Sla-aware best fit decreasing techniques for workload consolidation in clouds." IEEE Access vol. 7, pp. 135256-135267, 2019.

[12] S.S. Gill, I. Chana, M. Singh, and R. Buyya, "CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing." Cluster Computing vol. 21, no. 2, pp. 1203-1241, 2018.

[13] J.N. Witanto, H. Lim, and M. Atiquzzaman, "Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management." Future generation computer systems vol. 87, pp. 35-42, 2018.

[14] D. Saxena and A.K. Singh, "Workload forecasting and resource management models based on machine learning for cloud computing environments." arXiv preprint arXiv:2106.15112, 2021.

[15] J. Singh and M.S. Goraya, "Multi-objective hybrid optimization based dynamic resource management scheme for cloud computing environments." In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, pp. 386-391, 2019.

[16] M. Ghobaei-Arani, "A workload clustering based resource provisioning mechanism using Biogeography based optimization technique in the cloud based systems." Soft Computing vol. 25, no. 5, pp. 3813-3830, 2021.

[17] S.A. Ali, M. Ansari, and M. Alam, "Resource management techniques for cloud-based IoT environment." In Internet of Things (IoT), Springer, Cham, pp. 63-87, 2020.

[18] S. Goodarzy, M. Nazari, R. Han, E. Keller and E. Rozner, "Resource management in cloud computing using machine learning: A survey." In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp. 811-816, 2020.

[19] D.P. Sharma, B.K. Singh, A.T. Gure, and T. Choudhury, "Emerging Paradigms and Practices in Cloud Resource Management." In Autonomic Computing in Cloud Resource Management in Industry 4.0, Springer, Cham, pp. 17-39, 2021.

[20] M.R. Raza, and A. Varol, "QoS parameters for viable SLA in cloud." In 2020 8th International Symposium on Digital Forensics and Security (ISDFS), IEEE, pp. 1-5, 2020.

[21] M. Daraghmeh, S.B. Melhem, A. Agarwal, N. Goel, and M. Zaman, "Linear and logistic regression based monitoring for resource management in cloud networks." In 2018 IEEE 6th international conference on future internet of things and cloud (FiCloud), IEEE, pp. 259-266, 2018.

[22] S.R. Swain, A.K. Singh, and C.N. Lee, "Efficient Resource Management in Cloud Environment." arXiv preprint arXiv:2207.12085, 2022.

[23] M.H. Khalil, M. Azab, A. Elsayed, W. Sheta, M. Gabr, and A.S. Elmaghraby, "Auto resource management to enhance reliability and energy consumption in heterogeneous cloud computing." International Journal of Computer Networks & Communications vol. 12, no. 2, 2020.

[24] R. Yadav, W. Zhang, O. Kaiwartya, P.R. Singh, I.A. Elgendy, and Y-C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing." IEEE Access vol. 6, pp. 55923-55936, 2018.

[25] R. Mandal, M.K. Mondal, S. Banerjee, and U. Biswas, "An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing." The Journal of Supercomputing vol. 76, no. 9, pp. 7374-7393, 2020.

[26] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, "Machine learning (ML)–Centric resource management in cloud computing: A review and future directions." Journal of Network and Computer Applications pp. 103405, 2022.

[27] F. Zaker, M. Litoiu and M. Shtern, "Formally Verified Scalable Look Ahead Planning for Cloud Resource Management." ACM Transactions on Autonomous and Adaptive Systems (TAAS) 2022.

[28] M.A.N. Saif, S.K. Niranjan, and H.D.E. Al-Ariki, "Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis." Wireless Networks vol. 27, no. 4, pp. 2829-2866, 2021.

[29] M.O. Agbaje, O.B. Ohwo, T.G. Ayanwola, and O. Olufunmilola, "A Survey of Game-Theoretic Approach for Resource Management in Cloud Computing." Journal of Computer Networks and Communications vol. 2022, 2022.

[30] S. Mustafa, K. Bilal, S.U.R. Malik, and S.A. Madani, "SLA-aware energy efficient resource management for cloud environments." IEEE Access vol. 6, pp. 15004-15020, 2018.

[31] R. Rajavel and M. Thangarathanam, "Agent-based automated dynamic SLA negotiation framework in the cloud using the stochastic optimization approach." Applied Soft Computing vol. 101, pp. 107040, 2021.

[32] A. Kannaki, V. Azhagu and J.M. Gnanasekar, "A Novel Multi-Agent Approach to control Service level Agreement Violations in Cloud Computing." Turkish Journal of Computer and Mathematics Education vol. 12, no. 12, pp. 1431-1438, 2021.

[33] A. Heidari, and N.J. Navimipour, "A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm." PeerJ Computer Science vol. 7, pp. e539, 2021.

[34] J.K. Samriya, S.C. Patel, M. Khurana, P.K. Tiwari, and O. Cheikhrouhou, "Intelligent SLA-aware VM allocation and energy minimization approach with EPO algorithm for cloud computing environment." Mathematical Problems in Engineering vol. 2021, 2021.

[35] A. Belgacem, S. Mahmoudi, and M. Kihl, "Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing." Journal of King Saud University-Computer and Information Sciences 2022.

[36] S. Son, and S.C. Jun, "Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing." In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, IEEE, pp. 168-171, 2013.

[37] L. Wu, S.K. Garg, R. Buyya, C. Chen and S. Versteeg, "Automated SLA negotiation framework for cloud computing." In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, pp. 235-244, 2013.

An Efficient Deep Learning based Hybrid Model for Image Caption Generation

Mehzabeen Kaur[1], Harpreet Kaur[2]
Ph.D Research Scholar, Department of Computer Science and Engineering, Punjabi University, Patiala[1]
Faculty, Department of Computer Science & Engineering, Punjabi University, Patiala[2]

***Abstract*—In the recent yeas, with the increase in the use of different social media platforms, image captioning approach play a major role in automatically describe the whole image into natural language sentence. Image captioning plays a significant role in computer-based society. Image captioning is the process of automatically generating the natural language textual description of the image using artificial intelligence techniques. Computer vision and natural language processing are the key aspect of the image processing system. Convolutional Neural Network (CNN) is a part of computer vision and used object detection and feature extraction and on the other side Natural Language Processing (NLP) techniques help in generating the textual caption of the image. Generating suitable image description by machine is challenging task as it is based upon object detection, location and their semantic relationships in a human understandable language such as English. In this paper our aim to develop an encoder-decoder based hybrid image captioning approach using VGG16, ResNet50 and YOLO. VGG16 and ResNet50 are the pre-trained feature extraction model which are trained on millions of images. YOLO is used for real time object detection. It first extracts the image features using VGG16, ResNet50 and YOLO and concatenate the result in to single file. At last LSTM and BiGRU are used for textual description of the image. Proposed model is evaluated by using BLEU, METEOR and RUGE score.**

Keywords*—*CNN; RNN; LSTM; YOLO

I. Introduction

In this www world, every day in our life, all have experienced with the huge number of images in a real world which are self-interpret by the individual human being by using their wisdom. Human are naturally programmed to convert the natural scene in to text but it is the complex task for the machine as they are not much efficient like human. Still, human generated captions are considered better as machine need human intervention and programmed accordingly for the better result. Due to the recent development in deep learning-based techniques, computers are capable to handle the challenges of image captioning like detection of object, attribute and their relationship, image feature extraction and generating syntactic and semantic image caption [1].

With the advancement of AI, so many new ideas have revolutionized in the areas of image processing and it has transformed the world in a surprising way. The image captioning Approach (Fig. 1) has wider application in the real world as it provides the better platform for human computer interaction. Due to the emerging application in image processing, image captioning becomes the topic of interest for the academician and researchers.

By seeing the Fig. 2, picture someone guess that two dogs are playing with toy and someone might say two dogs hauling in floating toy from the ocean or two dogs run through the water with rope in their mouths, so all of these captions are appropriate to describe this picture. Our brain is so much trained and advanced that it can describe a picture almost accurate but same was not the case with machines.

Hence, the main aim of the image captioning is first identified the different objects and their relationship present in the image using deep learning-based technique, generating the textual description using the natural language processing and evaluate the performance of the natural language-based description using different performance matrices. Object detection and segmentation are the part of the computer vision and done with the help of popular CNN and DNN and generating image description (Fig. 3) are the part of natural language processing which is done by RNN and LSTM. CNN works for understanding the objects of the image or scene and provide the answers the various questions about the objects in image like what, where, how, etc.

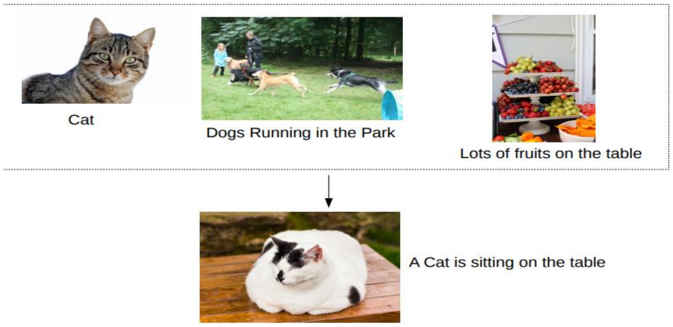


Fig. 1. Image captioning.

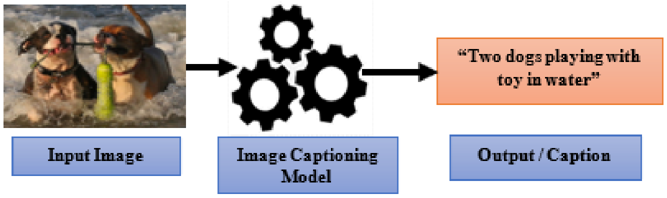


Fig. 2. Working of image captioning.

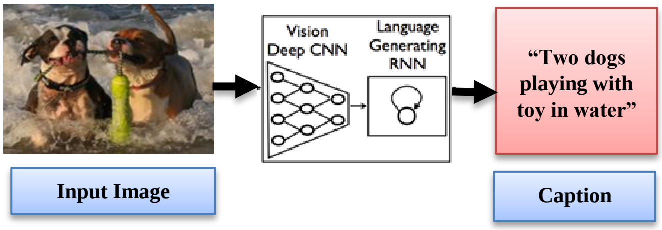


Fig. 3. Image captioning architecture.

For example, in Fig. 3, CNN identify the "dog", "toy", "water" and their relationship in the scene. Further RNN give the shape in textual form by using the keywords described by CNN by considering it in group of words. This one is also called the encoder-decoder architecture. Object detection is a part of computer vision which uses various algorithms, like YOLO, R-CNN, Mask R-CNN, MobileNet and SqueezeDet for detecting the different parts of the image efficiently.

II. Literature Survey

In this section, review of literature in image captioning is presented. Various state-of-the-art techniques and model have been published in previous years to generate the human like captions. Image captioning approaches [11], [14] and [17] broadly classified in to Template-based [18-21], Retrieval-based [22-26], and Encoder-decoder methods [27-30]. In paper [31] a content selection approach has been proposed for image description by using geometric, conceptual and visual features of image. All of these models work on CNN, first use encode the image and extract the feature and further use RNN or LSTM to make captions of the image. In paper [1] researchers presented an image captioning model with probabilistic distribution using successor and predecessor words and image captioning. Attention and visual based approach are very famous approach in image captioning. In [2,3] authors generate the captions using the attention mechanism. In most of the papers predefined models were used in bulk of papers like VGG16 papers [1], [3-7], YOLO [8], Inception V3 [9-10], AlexNet [5], [7], ResNet [4-5], [12] and Unet [13] are the famous encoder or CNN model used for image feature extraction. For image caption generation or decoding, LSTM [4], [8-10] and [15], BiLSTM [7], [13], RNN [16]. Image captions are also generated in various languages like Chinese, Japanese, Hindi, Punjabi and German, etc.

Template based approach uses predefined templates of objects, actions and attributes to identify the input image [18], the authors use visual elements like object, action and scene for predicting the caption of the image. In [19] author takes the advantages of Conditional Random Field (CRF) based technique extract the features of the image. The proposed model evaluated using BLUE and ROUGE score on PASCAL dataset. As it is based upon pre-defined template it is not able to generate the caption of image with variable lengths.

Retrieval based approach generate caption by capering the features of the image with the datasets. It tries to finds the caption for input image by discovering similar features in the dataset. In [22] authors proposed a model to extract feature of the query image by searching it through the dataset and in [32], the authors propose the caption by using the density estimation method. In [25], the authors used semantic and visual features for image caption generation.

In the original dataset we have five captions for each image and our goal is to train a particular model on this dataset. After the training phase model becomes efficient for extracting the features of the particular image, various predefined image classification models are available which uses state-of-the-art algorithms for classifying the thousands of different objects/images efficiently. These models come up with better accuracy with respect to image rate classification, like ResNet. These are very easy to implement.

Encoder-decoder based approach is a most widely used for machine translation and image caption generation which is based upon deep neural networks. A dual graph convolution network based is proposed in [33] and NIC (Neural Image Caption) model based on encoder-decoder architecture is in [27]. This one is a simple model where CNN is used as a encoder, and in the decoder end LSTM and RNN are used for image caption generation.

III. Research Methodology

Here, for extracting the visual feature of the image, CNN used as an encoder which have Convolution layer, Pooling layer, and fully connected layer. Earlier AlexNet was used for compute vision problems but nowadays, the transfer learning are in trends in where several pre-trained CNN based models are available like VGGNet, Inception V3, DenseNet, ResNet etc. which are available with different convolutional neural layers and used for saving the training time of the model. Further, decoder is used to generating the final captions which gets the input from the encoder. GRU, LSTM and RNN are the most commonly used decoder. RNN are suitable for short words sequence and LSTM is best for long sequence.

This section depicts the proposed hybrid research methodology. Our main objective of the proposed model is to achieve the higher Meteor value. Our model is based on an Encoder-Decoder approach where it used the concept of transfer learning. Here in the first phase, features of the image is extracted by using VGG16, ResNet50 and YOLO (You Only Look Once) separately. YOLO is an efficient object detection algorithm in real time with is developed in 2015 Joseph Redmon et al. whereas VGG16 (Visual Geometry Group) is an object detection and classification approach which is pretrained on ImageNet dataset. This is deep Convolutional Neural Network (CNN) architecture which uses 16 convolutional layers. ResNet50 is a deep CNN with 50 convolutional layers which is able to classify more than 1000 object category.

In second phase, concatenate of the features of image extracted by the VGG16, ResNet50 and YOLO and all the duplicate words are eliminated.

In third phase, captions are generated by using the BiGRU and LSTM. BiGRU (Bidirectional Gated Recurrent Units) is a Neural Network architecture used in NLP (Natural Language Processing). This architecture uses two GRUs for taking input in forward and backwards directions. LSTM (Long Short - Term Memory) is a type of recurrent neural network architecture which used feedback connections and capable of identifying the relation between objects. In the last phase, both the captions are compared with the Meteor performance evaluation metrices. Final caption has the higher meteor value.

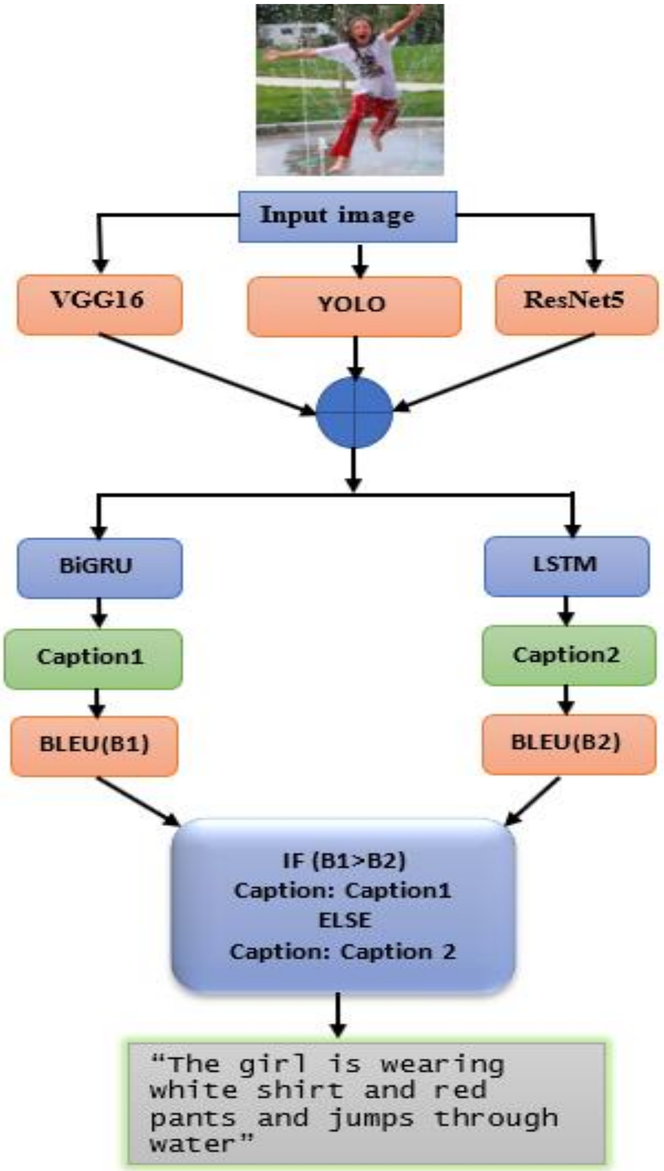


Fig. 4. Proposed image captioning architecture.

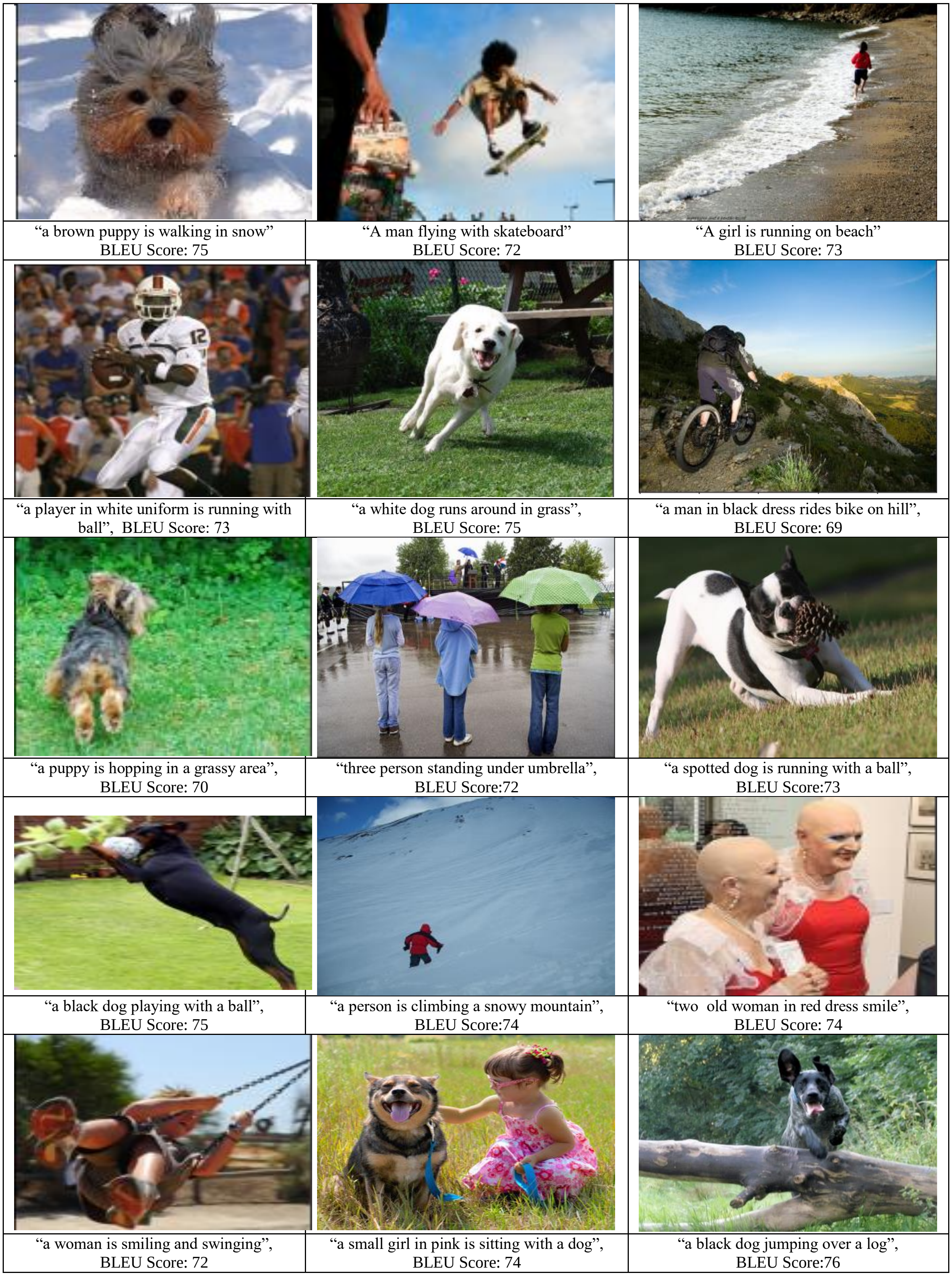
IV. Datasets

Data are the backbone of any AI based systems. Recently image captioning is blessed with rich datasets like MSCOCO, Flickr8k, Flickr30k, PASCAL etc. in the dataset, every image is described in related five reference sentences. Every description of the scene is described by using different algorithms and grammar. MSCOCO is a large dataset which was developed by Microsoft whose target to describe the image as a human being. It first understands the scene and complete the image recognition, segmentation and generating suitable caption of the image. It contains 82,783 images, with validation set 40,504 images, and the test set 40,775 images. Flickr30k dataset has 28000 training images, 1000 testing and 1000 validation images.

Here, in this paper a benchmark dataset Flickr8k for the training of the model. It contains 8000 images with 5 captions of each image which provides the clear descriptions of the silent objects. It has manually labelled captions for all the images in English language. The dataset is divided into two categories. First one is image directory which has 8k images with 5 captions. Out of 8000 images, 6000 are used for training and remaining 2k images are for training purpose. Images in Flickr8k dataset are in jpg format with resolution 256*500 to 500*500 and average length of sentence is 12 words.

V. Result and Analysis

Performance of the image captions are evaluated by using different evaluation BLEU, METEOR, ROUGE, CIDEr and SPICE metrics. When analyzing the proposed model and matching the predicted words to their original captions, the BLEU score is applied. Fig. 4 illustrates how the loss gradually decreased as the number of training epochs grew. it could train our datasets across more epochs to get better descriptions, and here it did so for 100 epochs to enable comparison study. The loss value is between 0.5 and 0.1 epochs. Maximum and minimum values are observed for 10 epochs with losses of 0.5+ and less than 0.1 epoch, respectively. In Fig. 5, the comparison of the predicted caption with five additional original captions using a graphic representation of the BLEU score is illustrated. From 5 to 10 epochs, a sharp increase is observed from 0.50 to 0.56 BLEU score, then the graph experiences slight ups and downs till 50 epochs. Another score called "match words" counts the words that match up with the produced text of a picture. As shown in the graphical representation, the match words undergo significant upswell with changes as time passes. Witnessed as 0.49 match words in the case of 50 epochs and 0.40 in the case of 5 epochs. When Match Word and BLEU Score were compared, it was found that both inclined before reaching the heights. In the instance of Match words, the score increased from 0.500 to 0.555 from 5 to 10 epochs. After that, this sample saw minor changes through 50 epoch, reaching a score of 0.575. When discussing the BLEU score, it had two distinct peaks at 0.450 and 0.470 score at the 15 and 30 epochs. At 35, the graph had a slight decline (0.460), and at 50, it finally hit the score (0.480).



"a brown puppy is walking in snow"
BLEU Score: 75

"A man flying with skateboard"
BLEU Score: 72

"A girl is running on beach"
BLEU Score: 73

"a player in white uniform is running with ball", BLEU Score: 73

"a white dog runs around in grass",
BLEU Score: 75

"a man in black dress rides bike on hill",
BLEU Score: 69

"a puppy is hopping in a grassy area",
BLEU Score: 70

"three person standing under umbrella",
BLEU Score:72

"a spotted dog is running with a ball",
BLEU Score:73

"a black dog playing with a ball",
BLEU Score: 75

"a person is climbing a snowy mountain",
BLEU Score:74

"two old woman in red dress smile",
BLEU Score: 74

"a woman is smiling and swinging",
BLEU Score: 72

"a small girl in pink is sitting with a dog",
BLEU Score: 74

"a black dog jumping over a log",
BLEU Score:76

Fig. 5. Image captions generated by proposed approach.

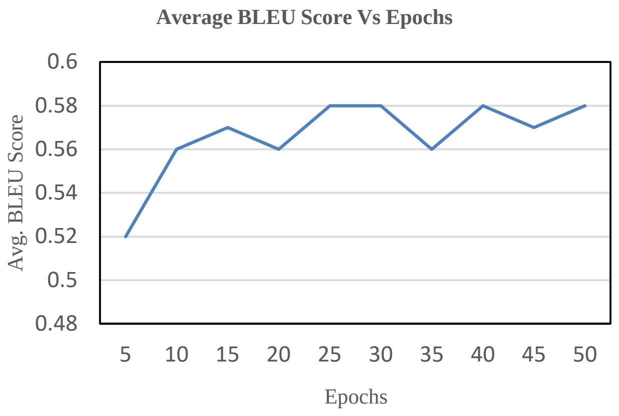


Fig. 6. Average BLEU Score vs Epochs.

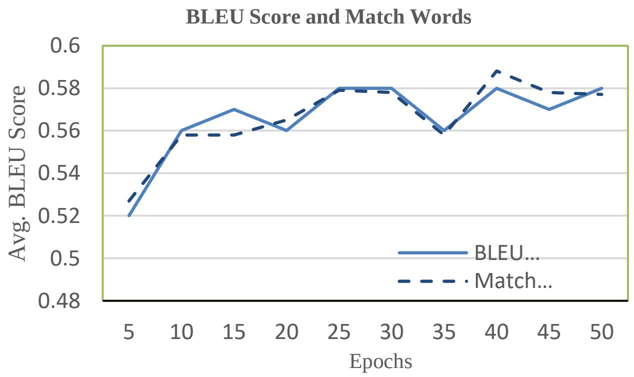


Fig. 7. BLEU Score vs Match words.

The graphical representation illustrates the model's recall changes with threshold values. Threshold values from 0.0 to 0.25 remained constant at 1. After then, a steady fall was observed from 0.25 to 0.75 and approached 0.0 value until a very little increase with around 0.1 recall value was noted too and final recalled value is accounted as 64.056. The graph that depicts the variation in accuracy with threshold values changes the shape of a sharp peak that is constant at 0.500 accuracy up until 0.0 to 0.25v threshold value, then a straight climb up to 0.675 accuracy, followed by a similar value fall up until 0.75 threshold value (Fig. 6). And resultant accuracy is 67.052. The graph shows model precision levels as well as variations in threshold settings. Although the precision value overall is 68.138, changes are seen from a 0.2 threshold value to a 0.75 with a simple increase in the precision values. Other starting and ending values were 1.0 from .075 to 0.25 and 0.5 from 0.0 to 0.25. Further in Fig. 7, BLEU score and Match score are compared which shows the compatible score. First average score of both are .52 on 5 Epochs. At 10 Epochs the values are increased to 0.56. it shows its best performance in 30 Epochs and decreases in 35 Epochs due the overfitting. In Fig. 8, 9 and 10 precision recall and accuracy are shown.

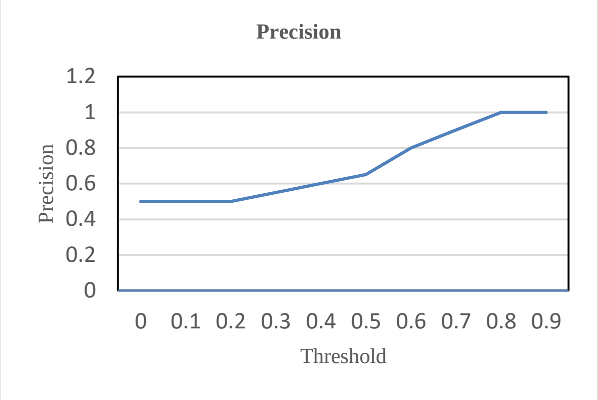


Fig. 8. Precision of proposed systems.

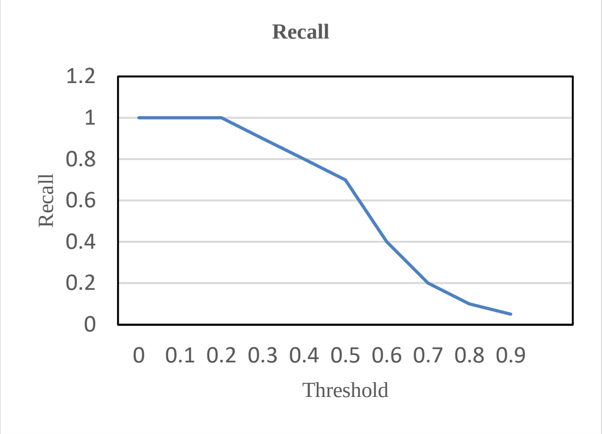


Fig. 9. Recall of proposed systems.

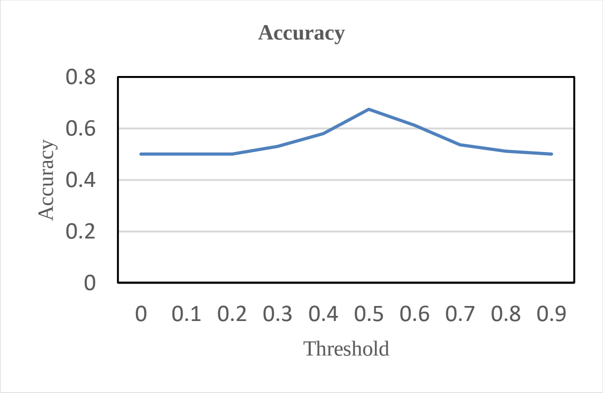


Fig. 10. Accuracy of proposed systems.

The represented graph illustrates the loss and the epochs. According to the provided scale, maximum values are attained by 1.0 on 0.0 epochs. The loss reached a value of 0.75 at 1.0 epochs. Moving further with a curved change value of loss and epochs graph, the loss stopped at 17.5 epochs when the value of loss was witnessed as 0.3.

TABLE I. COMPARATIVE ANALYSIS OF PROPOSED APPROACH WITH SINGLE MODEL

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
Inception V3	0.65	0.43	0.29	0.17	0.21	0.41
VGG16	0.66	0.38	0.30	0.16	0.23	0.22
Res Net50	0.56	0.31	0.18	0.12	0.27	0.51
VGG19	0.61	0.35	0.28	0.18	0.21	0.22
Proposed Hybrid Approach	0.67	0.46	0.35	0.26	0.31	0.54

TABLE II. COMPARATIVE ANALYSIS OF PROPOSED APPROACH WITH HYBRID MODEL

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
Densenet169 + LSTM [34]	63.73	45.00	30.87	21.13	46.41	19.95
Resnet101 + LSTM [35]	62.77	44.11	30.62	21.10	43.54	18.79
VGG-16 + LSTM [36]	60.56	41.98	28.66	19.51	44.82	19.04
Densenet121 + Attention + LSTM[34]	65.00	46.99	32.83	22.56	47.57	20.44
ResNet152 + Attention + LSTM [37]	65.26	47.55	33.72	23.67	47.54	20.94
VGG-16 + Attention + LSTM [36]	63.81	45.77	32.35	22.55	46.72	20.19
Proposed Hybrid Approach	0.67	0.46	0.35	0.26	0.31	0.54

The given Tables I and II are the results from an LSTM based decoder model using a signal encoder on the flickr8k dataset. There are five encoders (Inception V3, VGG16, Res Net50, VGG19, and Proposed Hybrid Approach) given each represents their own values of BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, and METEOR in the table chart. The maximum value in terms of BLEU-1 data is 0.67 for the proposed Hybrid Approach Encoder. However, in BLEU-2, the minimum value is held by Res Net50. Considering the data in BLEU-3 and BLEU-4, the minimum is send in the case of ResNet50 as 0.18 and 0.12, whereas the maximum is witnessed in the case of the proposed Hybrid Approach Encoder. In ROUGE-L, data is numbered as 0.21, 0.23, 0.27, 0.21, and 0.31 for Inception V3, VGG16, Res Net50, VGG19, and Proposed Hybrid approach, respectively. On the other hand, 0.22 was the value which was similar to VGG16 and VGG19 in the case of METEOR.

VI. CONCLUSION

In this paper, a hybrid encoder-decoder based model to generate the effective caption of the image by using the Flickr8k dataset. During the encoding phase, the proposed model used transfer learning-based model like VGG16 and ResNet5o and YOLO for extracting the image features. A concatenate function is used to combine the feature and removes the duplicate one. For the decoding, BiGRu and LSTM are used to get the complete caption of the image. Further BLEU value is evaluated of both the captions generated by BiGRU and LSTM. Final caption is considered whose METEOR value is high. The proposed model is also evaluated by METEOR and ROUGE. The proposed model achieved score BLUE-1: 0.67, METEOR: 0.54 and ROUGE: 0.31 on Flickr8k dataset. The experimental results show the better results through BLUE, METEOR and ROUGE when compared to another state-of-art models. The model is also helpful in generating the captions at real time.

REFERENCES

[1] J. Gu, G. Wang, J. Cai, and T. Chen, "An Empirical Study of Language CNN for Image Captioning," Proc. IEEE Int. Conf. Comput. Vis., vol. 2017-October, pp. 1231–1240, 2017, doi: 10.1109/ICCV.2017.138.

[2] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional Image Captioning," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 5561–5570, 2018, doi: 10.1109/CVPR.2018.00583.

[3] K. Xu et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." Available: http://proceedings.mlr.press/v37/xuc15.

[4] K. Xu, H. Wang, and P. Tang, "Image Captioning With Deep Lstm Based On Sequential Residual Department of Computer Science and Technology , Tongji University , Shanghai , P . R . China Key Laboratory of Embedded System and Service Computing , Ministry of Education ," no. July, pp. 361–366, 2017.

[5] S. Liu, L. Bai, Y. Hu, and H. Wang, "Image Captioning Based on Deep Neural Networks," MATEC Web Conf., vol. 232, pp. 1–7, 2018, doi: 10.1051/matecconf/201823201052.

[6] R. Subash, R. Jebakumar, Y. Kamdar, and N. Bhatt, "Automatic image captioning using convolution neural networks and LSTM," J. Phys. Conf. Ser., vol. 1362, no. 1, 2019, doi: 10.1088/1742-6596/1362/1/012096.

[7] C. Wang, H. Yang, and C. Meinel, "Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning," ACM Trans. Multimed. Comput. Commun. Appl., vol. 14, no. 2s, 2018, doi: 10.1145/3115432.

[8] M. Han, W. Chen, and A. D. Moges, "Fast image captioning using LSTM," Cluster Comput., vol. 22, pp. 6143–6155, May 2019, doi: 10.1007/s10586-018-1885-9.

[9] H. Dong, J. Zhang, D. Mcilwraith, and Y. Guo, "I2T2I: Learning Text To Image Synthesis With Textual Data Augmentation."

[10] Y. Xian and Y. Tian, "Self-Guiding Multimodal LSTM - When We Do Not Have a Perfect Training Dataset for Image Captioning," IEEE Trans. Image Process., vol. 28, no. 11, pp. 5241–5252, 2019, doi: 10.1109/TIP.2019.2917229.

[11] K. Xu, H. Wang, and P. Tang, "Image Captioning With Deep Lstm Based On Sequential Residual" Department of Computer Science and Technology , Tongji University , Shanghai , P . R China Key Laboratory of Embedded System and Service Computing , Ministry of Education ," no. July, pp. 361–366, 2017.

[12] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain Images with Multimodal Recurrent Neural Networks," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1410.1090.

[13] W. Cui et al., "Landslide image captioning method based on semantic gate and bi-temporal LSTM", ISPRS Int. J. Geo-Information, vol. 9, no. 4, 2020, doi: 10.3390/ijgi9040194.

[14] H. Dong, J. Zhang, D. Mcilwraith, and Y. Guo, "I2T2I: Learning Text To Image Synthesis With Textual Data Augmentation."

[15] C. Liu, F. Sun, and C. Wang, "MMT: A multimodal translator for image captioning," , Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10614 LNCS, p. 784, 2017.

[16] Q. You, H. Jin, Z. Wang, C. F.-P. of the I., and undefined 2016, "Image captioning with semantic attention," openaccess.thecvf.com Available: http://openaccess.thecvf.com/.

[17] X. Liu, Q. Xu, and N.Wang, "A survey on deep neural network-based image captioning" ,The Visual Computer, 35(3):445– 470, 2019.

[18] A. Farhadi, M. Hejrati, M. Amin Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images", In European conference on computer vision, pages 15–29. Springer, 2010.

[19] G. Kulkarni, V. Premraj, V. Ordonez, S, Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg, "Baby talk: Understanding and generating simple image descriptions", IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(12):2891–2903, 2013.

[20] Y. Yang, C. Lik Teo, H. Daum´e, and Y. Aloimonos, "Corpus-guided sentence generation of natural images", EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, (May 2014):444–454, 2011.

[21] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daum´e, " Midge: Generating image descriptions from computer vision detections", EACL 2012 - 13th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings, pages 747– 756, 2012.

[22] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg, "Im2Text: Describing images using 1 million captioned photographs", Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011, pages 1–9, 2011.

[23] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk", In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pages 139–147, 2010.

[24] N. Gupta and A. Singh Jalal, "Integration of textual cues for fine-grained image captioning using deep cnn and lstm", Neural Computing and Applications, 32(24):17899– 17908, 2020.

[25] C. Sun, C. Gan, and R. Nevatia, "Automatic concept discovery from parallel text and visual corpora", Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter:2596–2604, 2015.

[26] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics", IJCAI International Joint Conference on Artificial Intelligence, 2015-Janua(Ijcai):4188–4192, 2015.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan., "Show and tell: A neural image caption generator", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June:3156–3164, 2015.

[28] C. Amritkar and V. Jabade, "Image Caption Generation Using Deep Learning Technique", Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018, pages 1–4, 2018.

[29] A. Ghosh, D. Dutta, and T. Moitra, "A Neural Network Framework to Generate Caption from Images", Springer Nature Singapore Pte Ltd., pages 171–180, 2020.

[30] J. Donahue, L. Anne Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description", IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):677–691, 2017.

[31] G. Barlas, C. Veinidis, and A. Arampatzis, "What we see in a photograph: content selection for image captioning", The Visual Computer, 37(6):1309–1326, 2021.

[32] R. Mason and E. Charniak, "Nonparametric Method for Data-driven Image Captioning", pages 592–598, 2014.

[33] X. Dong, C. Long, W. Xu, and C. Xiao, "Dual graph convolutional networks with transformer and curriculum learning for image captioning", arXiv preprint arXiv:2108.02366, 2021.

[34] H., Gao, Z. Liu, L. Van Der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. 2017.

[35] He, Kaiming, X. Zhang, S. Ren, and J. Sun," Deep residual learning for image recognition", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[36] Simonyan, Karen and Zisserman, Andrew, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556, 2014.

[37] He, Kaiming, X. Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

Collaborative based Vehicular Ad-hoc Network Intrusion Detection System using Optimized Support Vector Machine

Azath M, Vaishali Singh
Department of Computer Science & Engineering-School of Engineering & Technology,
Maharishi University of Information Technology, Lucknow, Uttar Pradesh 226013, India

***Abstract*—The Vehicular Ad hoc Network (VANET) can be used to provide secured information to the user vehicles. However, these days the immunity of safeguarding the information from vulnerabilities and threats are of great challenge. Therefore, it is necessary to provide a secured solution for the improvement of security with the deployment of advanced technology. In context with this, a blockchain based VANET structure for secured communication incorporated with the enhancement of confidentiality, scalability, and privacy is planned. The k-means clustering model forms cluster formation. The cluster head selection is carried out with Tabu Search-based Particle Swarm Optimization (TS-PSO) algorithm. The proposed approach aims to mitigate the delay with the enhancement of throughput and energy efficiency. Meanwhile, the deployed blockchain will enhance reliability and security. Moreover, the novel War Strategy Optimization (WSO) based Support Vector Machine (SVM) model (Optimized SVM) can be used for the trust-based collaborative intrusion detection in the VANET. Our work targets to detect the intrusion and non-intrusion classes. Meanwhile, our proposed work can be used for the prevention of repetitive detection processes and therein it enhances the security by rewarding the vehicles. An experimental analysis is carried out to ensure its usage in detecting the malicious node from the resource constraint vehicles and also used to achieve better security, energy utilization and end-to-end delay.**

Keywords*—*Vehicular Ad hoc network; intrusion detection; tabu search based particle swarm optimization; war strategy optimization; support vector machine

I. Introduction

The emerging trend of vehicular technology leads to the addition of smart equipment in vehicles and advancement of the intelligent transportation system and self-driving vehicles. The communication among vehicles and the roadside unit (RSU) can be accomplished with the VANET which alerts the drivers about the traffic, emergency alerts, and safety messages [1, 2]. The increase in mobility of vehicles, intricate topology, and diverse communication are the major possibilities of the VANET system to have been attacked by intruders. The main purpose of the intruders is to break the smart communication between the vehicles and RSUs. This might lead to damage in the VANET system and overcoming this intrusion detection system is the best choice [3].

The performance of the IDS relies on its deployed location in the cluster head, vehicles, and RSU and the conventional IDS contains three various architectures including distributed centralized and hybrid IDS [4]. Most probably the detection of various attacks by the traditional approaches is limited and it is necessary to design a precise approach to detect all types of abnormalities in the system. The centralized approach of software-defined networking (SDN) provides better flexibility and security throughout the network; however, a system bottleneck occurs due to failure in the single point. The issues in the SDN such as reliability and scalability are visibly moved by the distributed networks [5, 6]. Moreover, the communication and computation overhead of the SDN controllers are also tackled by the distributed SDN via VANET. It also possesses some demerits and it is necessary to overcome them [7].

The securities of the vehicles are the most challenging ones since most of the vehicles are designed without considering the security system. However, they can perform communication without any delay but still increases the attacks [8]. This can be dealt with the conventional approaches such as encryption and neglecting the irrelevant nodes. Recently the vehicles are linked in the VANET and thus detect the attacks. So to secure the communication between the vehicles and RSUs a Blockchain-based collaborative intrusion detection approach is introduced for the VANETs. Different kinds of swarm-based optimization models such as particle swarm optimization, fish swarm optimization, Cuckoo Search algorithm, Glowworm Swarm Optimization (GSO), Genetic Algorithms (GA) etc. contains more searchability, less cost, less time execution and it is easy to solve the optimization problems. Tabu search creates a metaheuristics search technique that could really evaluate the optimum solution in addition to local optimization problems. The critical feature of Tabu Search is the use of adaptable storage to produce a much more adaptable behavior. PSO is the motion, which is controlled by two factors: information from particle to particle and iteration to iteration. The intrusion detection can be achieved with the SVM-based WSO approach. Prior to this the cluster head and cluster formulation are done with the Tabu Search-based Particle Swarm Optimization (TS-PSO) algorithm. This approach collaboratively trains the samples and achieves better detection.

The major contribution of this study is summarized as follows:

- The cluster formation in VANET is performed via the k-means clustering model.

- The Tabu Search-based Particle Swarm Optimization (TS-PSO) algorithm selects the optimal cluster heads.
- The blockchain-based security performs VANET security. Both intrusion and non-intrusion are detected via the novel War Strategy Optimization (WSO) based Support Vector Machine (SVM) model.

A. Problem Statement

Most vehicles are created without taking into account the security system, it is one of the most difficult areas to secure. They can communicate instantly, which makes attacks more common. Traditional methods, like encryption and ignoring unnecessary nodes, can be used to handle this. Recently, vehicles have been connected to the VANET, allowing them to detect threats.

Therefore, the existing studies created a Blockchain based collaborative intrusion detection approach for the VANETs to secure the connection between the cars and RSUs. The searchability, cost, and execution time of various swarm-based optimization models, such as fish swarm optimization, particle swarm optimization, glowworm swarm optimization, Cuckoo Search algorithm, genetic algorithms, and others, are higher. Tabu search develops a metaheuristics search method that can effectively access both the ideal solution and local optimization issues. The use of adaptable storage to create a significantly more adaptive behavior is a key component of Tabu Search. The SVM-based WSO technique can be used to detect intrusions.

II. Literature Review

A hybrid machine learning model was suggested by Bangui et al. [9] for the detection of intrusion in VANET. The Random Forest (RF) models were used to improve the accuracy of IDS. Compared to other methods, this RF model considerably improves the accuracy of detection based on the CICIDS2017 dataset. The detection accuracy is increased and the computational time is decreased but it failed to describe any security concept.

A hybrid data-driven model was introduced by Bangui et al. [10] for VANET intrusion detection. The explosive development in computing power deals to improve the IDS performance. The possible novel intruders were detected with the help of the post-detection stage. The corsets-based clustering and data classification were combined via a hybrid data-driven model. The computational overhead was less but the execution time was higher and had more complicated process.

In VANET, Zaidi et al. [11] suggested Host-based intrusion detection (H-ID). The statistical and graphical techniques represent the collection of extensive data. The rogue nodes were easily detected with the usage of different traffic conditions. According to the cooperative information, the application layer IDS were observed and evaluated with respect to the state-of-art results. Due to increased vehicular data, the computational complexity was also higher.

Based on ToN-IoT dataset, the machine learning (ML) techniques were suggested by Gad et al. [12] to detect the VANET intrusion. The ML techniques are the combination of Naive Bayes (NB), Support Vector Machine (SVM), XGBoost, Logistic Regression (LR), and k-Nearest Neighbor (kNN) and Decision Tree (DT). Both multi-class and binary classification issues were easily solved by using this ToN-IoT dataset. The class balancing was performed with the usage of the Synthetic minority oversampling technique. The higher feature dimensionality increases the overall complexity.

The deep learning (DL) model was suggested by Aboelfottoh et al. [13] for intrusion detection in ACVs and VANETs. More accurate and smarter IDS were made via deep learning thereby providing an efficient intrusion detection model but it failed to satisfy the security process. Based on the time series classification approach, the LSTM model was introduced by Yu et al. [14] for the VANETs intrusion detection system. The classification model of LSTM was used to enhance the false emergency message detection accuracy. For both collusion attack and normal scenarios, the time series feature vectors train and the traffic incident classifier were designed to identify traffic parameter patterns with higher computational difficulties.

The context of an intrusion detection system in VANET was designed by Gonçalves et al. [15]. Based on the geographic region, the publicly available VANET datasets were evaluated. The far more popular security policy was used to employ conventional instruments which can aid in the prevention of threats. In vehicular ad hoc networks, Alsarhan et al. [16] suggested an SVM-based intrusion detection system. This model outperformed classification accuracy with a better intrusion detection model and higher execution time than the existing methods such as PSO, ACO and GA. The distributed ensemble learning model was introduced by Ghaleb et al. [17] for misbehavior-aware on-demand collaborative IDS. The remote and locally trained classifiers were encompassed with the weighted random forest-based classifiers. Compared to the previous CIDS model, this distributed ensemble learning approach delineated 97% F1 score performances but the computational cost was higher and it needed more security during intrusion detection.

III. Proposed Methodology

The VANET are critical enablers of eventual collaborative transportation systems. Vehicles on VANETs exchange actual information regarding its location, congestion, and traffic conditions. Nevertheless, VANETs are vulnerable to threats that can result in existing circumstances. IDS depending on automotive collaboration to detect attackers in VANET were the most frequently proposed privacy model. The schematic diagram of the proposed intrusion detection model is shown in Fig. 1. The proposed methodology includes four stages namely cluster formation, cluster head selection, VANET security and Collaborative intrusion detection. Each of these stages is delineated in the following section.

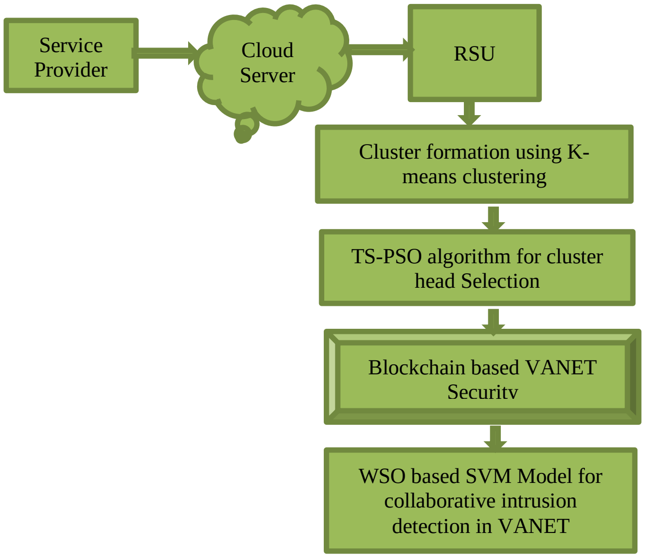

Service Provider
Cloud Server
RSU
Cluster formation using K-means clustering
TS-PSO algorithm for cluster head Selection
Blockchain based VANET Securitv
WSO based SVM Model for collaborative intrusion detection in VANET

Fig. 1. Proposed intrusion detection framework.

A. Cluster Formation

The stable vehicle cluster formations are the major step of this study. In this study, the K-means clustering algorithm was used to perform cluster formation. The edges were represented by using the distances between vehicles and the graph vertex act as each vehicle [18]. Based on roadside units (RSU) transmission region, the cluster formation of each vehicle was calculated. In the d-dimensional real vector, the observation set is $\left(A_1, A_2,.., A_m\right)$ [14]. Within cluster square sum (WCSS) is minimized via the m observation partition into K-sets $(K \le m) s = \left\{s_1, s_2, ..., s_K\right\}$.

$$WCSS = \arg\min_{s} \sum_{j=1}^{K} \sum_{A_i \in s_j} \left\| A_i - \chi_j \right\|^2 \qquad (1)$$

During cluster formation, the points mean S_j in is Xj.

B. Cluster Head Selection

The space solution apart from local optimality was analyzed via Tabu Search (TS) to represent the search process. The component of TS is an adaptive memory, which is more efficient. In a similar period, the respective optima were not attained with multiple objectives. In a rapid manner, the TS had hard combinatorial optimization problems and which led to select the solution design [19]. The particle swarm optimization (PSO) model contains several advantages in case of a direction to initial converging towards local optima, less population diversity, resolving optimization and greater convergence [20]. Tabu Search is a powerful stochastic efficient algorithm that, in theory, might aggregate monotonically to a global optimum, but it required a lot of time to reach the close-to-global minimum. The algorithm can maintain population variety by incorporating TS into PSO as a local development phase, which prevented it from leading to an incorrect local optimal solution.

In this study, Tabu Search based Particle Swarm Optimization (TS-PSO) algorithm was used for the selection of cluster heads in VANET. While compared to the PSO algorithm, the TS takes less computational time with the average energy consumption during CH selection [21]. The tradeoff between both PSO and TS can be neglected by combining the TS-PSO algorithm.

The following steps explain the TS-PSO model for cluster head selection.

- Initialize the base station location and energy nodes.
- The TS-PSO algorithm with a maximum number of iterations is initialized.
- The base station corresponding to the node's distances is calculated via the cluster formation.
- Determine the local best position of the PSO algorithm.
- Tabu memory entries to zero. The initialization of the PSO solution with the Tabu list calculates the global best solution.
- In the Taby memory, create the entry and routes are swapped.
- Evaluate the fitness function of the next position and note the fitness value in the Tabu list.
- The most effective solution from the Tabu list is eliminated.
- Tabu Search based Particle Swarm Optimization (TS-PSO) algorithm is used to optimally select the cluster heads in VANET.

C. VANET Security based on Blockchain Model

The blockchain is made up of a collection of blocks that are linked together. A block is a decentralized network ledger that is connected simultaneously [22]. The actions inside the block are irreversible and unchangeable. Each block inside the blockchain is connected by the subsequent block's hash. Any changes to a single block will have an impact on the rest network. Furthermore, the information loaded into blocks are entirely open. The data transfer from one vehicle to another is the major security issue in VANET. No necessary data is detected by the vehicles before sharing in the VANET task manager. Consequently, there was a probability for updated data to be uploaded into the Vehicular network [23]. This blockchain security model that successfully maintains the security across the system, to overcome these issues. Blockchain is the distributed immutable ledger that can be used to record transactions and asset tracking.

If the vehicle user transfers through one RSU zone to the next, that following RSU requires the vehicle users to be really authorized. This one will substantially add significant cost and reduce the VANET system's performance. The assets might include tangible and intangible. The values in the blockchain can be tracked virtually by the authenticated user thus mitigating the risk and cost. With the distributed data sharing and storage capacity, it can also avoid the hazards of an attack and outage at a single location [24]. Furthermore, the integrity and validity of the initial global strategy can be guaranteed using the open ledger in the blockchain. Based on the reward and punishment system, the cars in the VANET system submit exact information. This stops false information from being uploaded. It is simple to trace the features of each transaction.

D. Intrusion Detection

This section describes the war strategy optimization (WSO) based support vector machine (SVM) for collaborative intrusion detection in VANET.

1) Support Vector Machine (SVM): The machine learning approach SVM [25] was used for intrusion detection in our proposed approach. This proposed approach is based on the WSO-SVM-based collaborative to detect the attacks and normal traffics. The data were represented in n-dimensional space and detection of intrusion was conducted by detecting the hyper-plane first and classifying the malicious node as intrusion and others as normal. The workflow of the proposed SVM is shown in Fig. 2.

To spate, the various points from the input the SVM utilizes the free parameters which rely on the separation margin as depicted in Fig. 3 [26]. However, there occur over fitting issues which can be rectified by the introduction of the WSO algorithm. The main reasons for utilizing the SVM are speed and scalability and also reduced complexity.

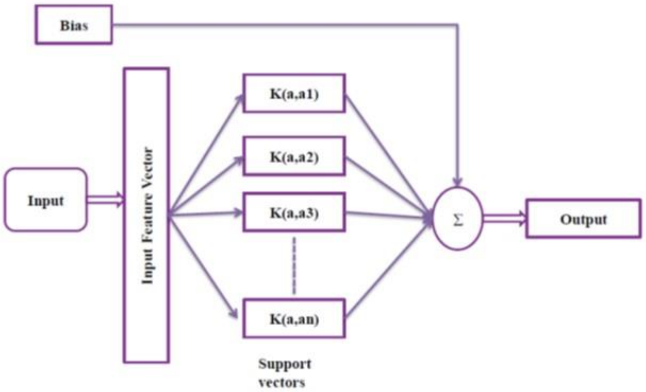


Fig. 2. Flowchart representation of the SVM model.

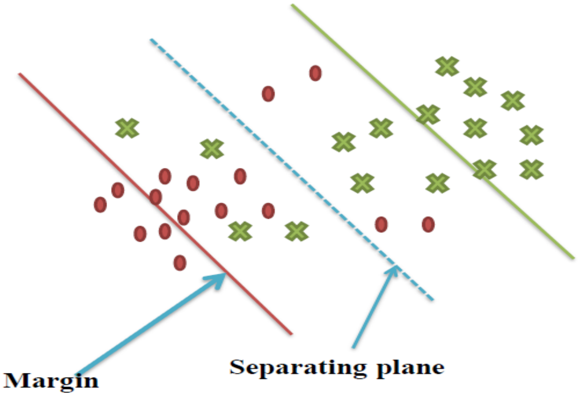


Fig. 3. Separating plane model of SVM.

2) War Strategy Optimization (WSO)

a) This WSO is based on the war strategy followed by the ancient kings based on the mission objectives, struggles, threats, and prospects. War is a continuous process in which the armed soldiers simply get together and fight the enemies.

The numerical expression followed by the strategy that the troops follow the king or commander on the war field. To avert falling prey to local troops the soldiers follow the combined movement tactics along with the king and the commanders.

- Attack Strategy

The war strategies have two models in which the former follows the updated locations of the soldiers with respect to the locations of the king. The soldier with high attack and fitness abilities are considered king. At first, all the soldiers possess the same rank and position and update the ranking location based on the finishing of war strategy. At end of the war the soldier, commander, and king all pretend to be close to each other and can be formulated as,

$$A_i(t+1) = A_i(t) + 2\times\tau\times(D-L) + ran\times(V_i \times L - A_i(t)) \quad (2)$$

The current location of the soldiers is Ai and the updated locations are depicted as A$_i$ (t+1) the location of the commander is D and L is the location of the king with the weight of V*i*.

- Up-gradation of rank and weight

The location of the soldiers can be updated with the location of the attack force. If the new location of the fitness (Fn) is lower than the current location (Fc) then the soldier will remain in the previous location [27].

$$A_i(t+1) = (A_i(t+1))\times(F_n \geq F_c) + (A_i(t))\times(F_n < F_c) \quad (3)$$

Then the rank of the soldier is updated as shown below,

$$S_i = (S_i + 1)\times(F_n \geq F_c) + (S_i)\times(F_n < F_c) \quad (4)$$

The estimation of weights based on the rank can be performed as,

$$V_i = V_i \times \left(1 - \frac{S_i}{Max\_iter}\right)\beta \quad (5)$$

- Defense strategy

The latter strategy was based on the up-gradation of locations of the king, commander, and a random soldier. The ranking and weight-up gradation follow the same strategy.

$$A_i(t+1) = A_i(t) + 2\times\tau\times(L - A_{ran}(t)) + ran\times V_i \times (D - A_i(t)) \quad (6)$$

- Weak Soldiers replacement/relocation

The detection of the worst soldiers is performed for each iteration and replaced with a random soldier as shown below,

$$A_v(t+1) = LC + ran\times(UC - LC) \quad (7)$$

The next approach is to reposition the weak soldier to the median of the whole army as shown below,

$$A_v(t+1) = -(1-rann)\times\left(A_v(t)-median(A)\right)+L \quad (8)$$

3) Enhancement of WSO-based SVM for intrusion detection.

- This proposed approach maintained the balance between exploitation and exploration.
- Each soldier (solution) maintained a unique weight with respect to the rank.
- After finishing the fitness step the weight of the soldiers got updated.
- At the starting stage of iteration the weights changed in large amount and decreases towards the end and this will result in a global optimum value.
- The stated approach simply increases the convergence speed and leads to less computation complexity.

4) Optimized SVM-based collaborative intrusion detection: The machine learning approach SVM provides efficient intrusion detection, and swift implementation, and is simple in nature. However, it possessed some shortcomings like over fitting issues and higher complexities [28]. These were tackled by the WSO algorithm which increased the convergence speed and searchability and increased the detection speed and accuracy. Thus the merging of WSO and SVM is performed in this proposed collaborative intrusion detection approach. The position parameters of SVM are updated by the fitness function of the proposed WSO algorithm. The schematic structure of the proposed WSO-SVM-based collaborative intrusion detection in VANET is illustrated in Fig. 4.

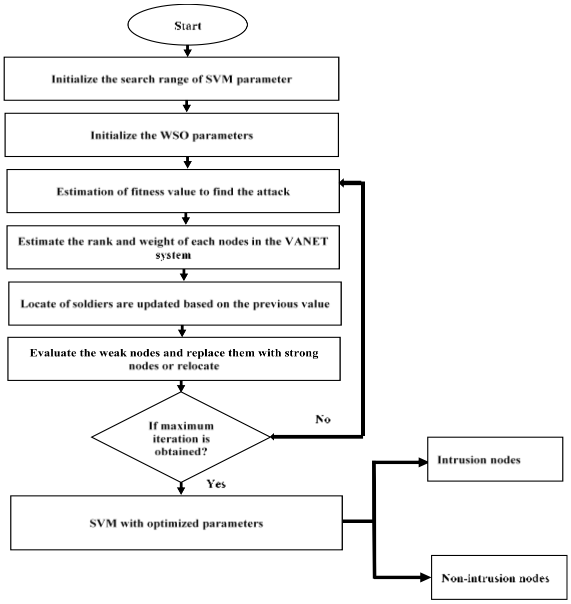

Start
Initialize the search range of SVM parameter
Initialize the WSO parameters
Estimation of fitness value to find the attack
Estimate the rank and weight of each nodes in the VANET system
Locate of soldiers are updated based on the previous value
Evaluate the weak nodes and replace them with strong nodes or relocate
If maximum iteration is obtained?
No
Yes
SVM with optimized parameters
Intrusion nodes
Non-intrusion nodes

Fig. 4. Schematic overflow of proposed optimized SVM-based collaborative intrusion detection approach in VANET.

IV. Result and Discussion

This section discusses the experimental investigation of a proposed framework based on collaborative intrusion detection in VANET. The GTX1050 GPU at 16GB RAM and Intel Core i5-8300H CPU with Tensorflow 1.15 on a GPU-based computer based on NS-2 implement the experimental results [29]. Table I explains the parametric description based on the proposed framework.

In this study, the KDD99 dataset was used for experimental investigation [30]. There were five million records involved in the KDD99 dataset and 41 features described these records.

TABLE I. Parameter Description

Parameters	Ranges
Number of population	50
Maximum number of iteration	100
Number of nodes	50
Simulation range	1.5 kms * 1.5 kms
Kernel function	Gaussian
Regularization	0.01

A. Performance Measures

The performance metrics such as accuracy (A), precision (P) and recall (R) are used to validate the effectiveness of proposed framework [30], [31]. The following equations explain all these performance measures for intrusion detection performance efficiency.

$$A = \frac{T_N + T_P}{F_N + T_N + F_P + T_P} \quad (9)$$

$$P = \frac{T_P}{F_P + T_P} \quad (10)$$

$$R = \frac{T_P}{F_N + T_P} \quad (11)$$

Where, the number of correctly predicted intrusion and the number of correctly predicted non-intrusion classes are true positive (T_P) and true negative (T_N). Moreover, a number of incorrectly predicted intrusion and the number of incorrectly predicted non-intrusion classes are true positive (F_P) and true negative (F_N).

B. Evaluation based on Performance

The analysis of accuracy is plotted in Fig. 5. A different number of nodes such as 10, 20, 30, 40 and 50 are used for the analysis of accuracy. The methods like RF, HDDM, H-ID, and ML are proposed. However, these proposed method offers superior accuracy than previous methods like RF, HDDM, H-ID, and ML in terms of all nodes.

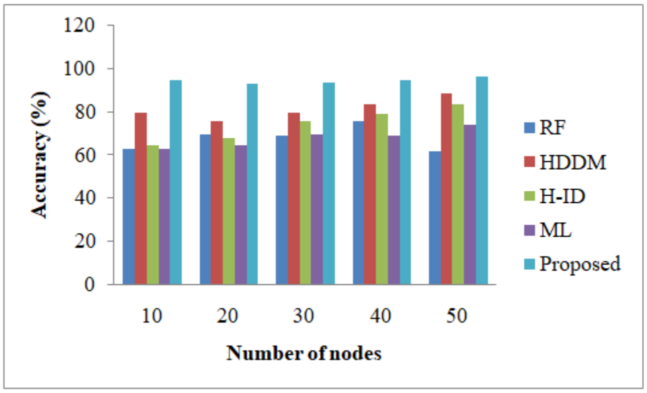


Fig. 5. Graphical representation of accuracy results.

The graphical representation of energy consumption results is plotted in Fig. 6. The methods including RF, HDDM, H-ID, and ML with proposed methods are taken as the state-of-art methods. There are 50 nodes which were taken for this investigation. The proposed method consumed less energy while comparing to the existing methods such as RF, HDDM, H-ID, and ML.

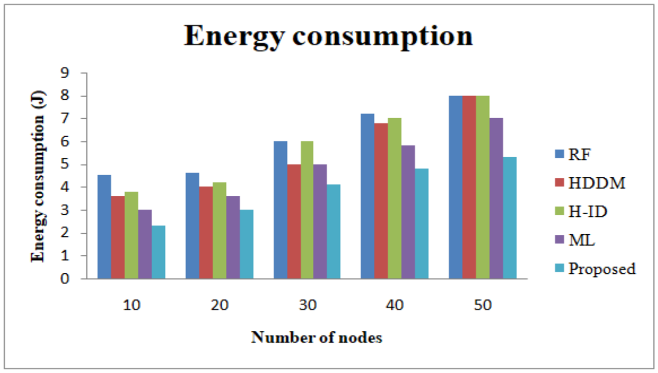


Fig. 6. Graphical representation of energy consumption results.

Fig. 7 represents the graphical representation of end-to-end delay results. The state-of-the-art methodologies include RF, HDDM, H-ID, and ML with proposed methodologies. For this examination, 50 nodes were taken and the delay is represented in time seconds. When compared to existing technologies like RF, HDDM, H-ID, and ML, the proposed method demonstrated minimum delay.

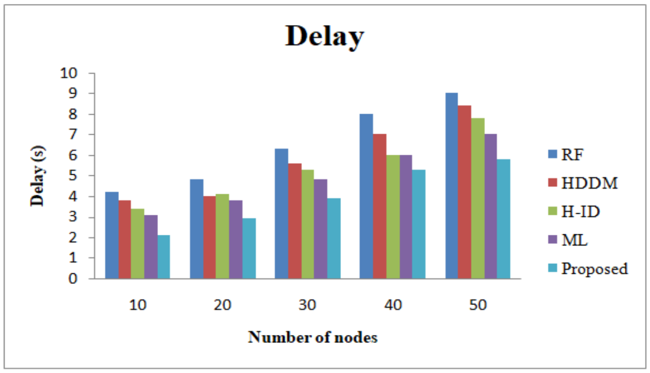


Fig. 7. Graphical representation of end-to-end delay results.

The performance evaluation of this proposed technique based on security was analyzed and compared with state-of-art work such as RF, HDDM, H-ID, and ML and reported in Table II. Since this proposed work utilized a collaborative blockchain-based WSO-SVM technique, it effectively secured the VANET system and avoided the vehicles from attacking. The security of the proposed approach is 95.89% and the other approaches RF, HDDM, H-ID, and ML provide security of 91.9%, 89.06%, 90.56%, and 93.23% respectively.

TABLE II. EVALUATION BASED ON SECURITY

Methods	Security (%)
RF	91.9
HDDM	89.06
H-ID	90.56
ML	93.23
Proposed	95.89

The evaluation based on the performance metrics was estimated and compared with state-of-art works such as RF, HDDM, H-ID, and ML, as stated in Table III. From the table III, it is clear that due to the inclusion of WSO along with the SVM classifier our approach provided better accuracy, precision, and recall of about 96.48%, 95.89%, and 96.56% respectively. Meanwhile, the other existing approaches show lacking strategies for above-stated metrics than this proposed approach.

TABLE III. EVALUATION BASED ON RECALL, PRECISION, AND ACCURACY

Methods	Recall (%)	Precision (%)	Accuracy (%)
RF	87.78	89.37	90.38
HDDM	91.09	90.56	92.77
H-ID	89.09	91.45	89.55
ML	93.45	90.76	93.00
Proposed	96.56	95.89	96.48

V. CONCLUSION

This article introduced a collaborative-based vehicular ad hoc network intrusion detection system using an optimized support vector machine. The cluster formation has been performed via k-means clustering. The Tabu Search-based Particle Swarm Optimization (TS-PSO) algorithm has been used to select the cluster heads for the formulated cluster. The suggested strategy helped to reduce the delay in the transmission of data by improving throughput and energy efficiency. The security of the proposed VANET system has been accomplished with the utilization of blockchain and enhanced its reliability. Additionally, the trust-based collaboration intrusion detection on the VANET can be performed using the optimized SVM model. The KDD99 dataset has been utilized and simulated with NS-2 software to analyze the performance of the proposed work. The proposed method offers minimum energy consumption as well as an end-to-end delay compared to the existing methods such as RF, HDDM, H-ID, and ML with the consideration of 50 nodes. The proposed strategy has a security level of 95.89%, while the other approaches—RF, HDDM, H-ID, and ML—offer security levels of 91.9%, 89.06%, 90.56%, and 93.23% correspondingly.

This study has few challenges in which the efficiency and accuracy of the intrusion detection framework can be enhanced via granularity and in-depth monitoring. The usage of different protocols and data diversity of the modern VANET networks induces a high level of complexity when identifying the intrusions in future.

References

[1] Tonguz, Ozan, Nawapom Wisitpongphan, Fan Bai, Priyantha Mudalige, and Varsha Sadekar, "Broadcasting in VANET," Mobile networking for vehicular environments, pp. 7-12. IEEE, 2011.

[2] Lee, M. and Atkison, T., "Vanet applications: Past, present, and future," Vehicular Communications, 28, p.100310, 2021.

[3] Hasrouny, Hamssa, Abed Ellatif Samhat, Carole Bassil, and Anis Laouiti, "VANet security challenges and solutions: A survey," Vehicular Communications, 7, 7-20, 2017.

[4] Calandriello, G., Papadimitratos, P., Hubaux, J.P. and Lioy, A., "Efficient and robust pseudonymous authentication in VANET," In Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks (pp. 19-28), 2007.

[5] Shu, Jiangang, Lei Zhou, Weizhe Zhang, Xiaojiang Du, and Mohsen Guizani, "Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach," IEEE Transactions on Intelligent Transportation Systems 22, no. 7, 4519-4530, 2020.

[6] Raja, G., Anbalagan, S., Vijayaraghavan, G., Theerthagiri, S., Suryanarayan, S.V. and Wu, X.W., "SP-CIDS: Secure and private collaborative IDS for VANETs," IEEE Transactions on Intelligent Transportation Systems, 22(7), pp.4385-4393, 2020.

[7] Maglaras, Leandros A, "A novel distributed intrusion detection system for vehicular ad hoc networks," International Journal of Advanced Computer Science and Applications 6, no. 4, 101-106, 2015.

[8] Shu, J., Zhou, L., Zhang, W., Du, X. and Guizani, M., "Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach," IEEE Transactions on Intelligent Transportation Systems, 22(7), pp.4519-4530, 2020.

[9] Bangui, H., Ge, M. and Buhnova, B., "A hybrid machine learning model for intrusion detection in VANET," Computing, 104(3), pp.503-531, 2022.

[10] Bangui, H., Ge, M. and Buhnova, B., "A hybrid data-driven model for intrusion detection in VANET," Procedia Computer Science, 184, pp.516-523, 2021.

[11] Zaidi, K., Milojevic, M.B., Rakocevic, V., Nallanathan, A. and Rajarajan, M., "Host-based intrusion detection for VANETs: A statistical approach to rogue node detection," IEEE transactions on vehicular technology, 65(8), pp.6703-6714, 2015.

[12] Gad, A.R., Nashat, A.A. and Barkat, T.M., "Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset," IEEE Access, 9, pp.142206-142217, 2021.

[13] Aboelfottoh, A.A. and Azer, M.A., "Intrusion Detection in VANETs and ACVs using Deep Learning," In 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 241-245). IEEE, 2022.

[14] Yu, Y., Zeng, X., Xue, X. and Ma, J., "LSTM-Based Intrusion Detection System for VANETs: A Time Series Classification Approach to False Message Detection," IEEE Transactions on Intelligent Transportation Systems , Vol. 23, Issue 12, 2022.

[15] Gonçalves, F., Macedo, J. and Santos, A., "Evaluation of VANET Datasets in context of an Intrusion Detection System," International Conference on Software, Telecommunications and Computer Networks (SoftCOM) (pp. 1-6). IEEE, 2021.

[16] Alsarhan, A., Alauthman, M., Alshdaifat, E., Al-Ghuwairi, A.R. and Al-Dubai, A., "Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks," Journal of Ambient Intelligence and Humanized Computing, pp.1-10, 2021.

[17] A. Ghaleb, F., Saeed, F., Al-Sarem, M., Ali Saleh Al-rimy, B., Boulila, W., Eljialy, A.E.M., Aloufi, K. and Alazab, M., "Misbehavior-aware on-demand collaborative intrusion detection system using distributed ensemble learning for VANET," Electronics, 9(9), p.1411, 2020.

[18] Hussain, I. and Chen, B., "Cluster formation and cluster head selection approach for vehicle ad-hoc network (VANETs) using K-means and floyd-Warshall technique," International Journal of Advanced Computer Science and Applications, 8(12), 2017.

[19] Kandali, K., Bennis, L. and Bennis, H., "A new hybrid routing protocol using a modified K-means clustering algorithm and continuous hopfield network for VANET," IEEE Access, 9, pp.47169-47183, 2021.

[20] Alinaghian, M., Tirkolaee, E.B., Dezaki, Z.K., Hejazi, S.R. and Ding, W., "An augmented Tabu search algorithm for the green inventory-routing problem with time windows," Swarm and Evolutionary Computation, 60, p.100802, 2021.

[21] Pervaiz, S., Ul-Qayyum, Z., Bangyal, W.H., Gao, L. and Ahmad, J., "A systematic literature review on particle swarm optimization techniques for medical diseases detection," Computational and Mathematical Methods in Medicine, Vol. 2021, Article ID 5990999, 2021.

[22] Vijayalakshmi, K. and Anandan, P., "A multi objective Tabu particle swarm optimization for effective cluster head selection in WSN," Cluster computing, 22(5), pp.12275-12282, 2019.

[23] Maria, A., Rajasekaran, A.S., Al-Turjman, F., Altrjman, C. and Mostarda, L., "Baiv: An efficient blockchain-based anonymous authentication and Integrity Preservation Scheme for secure communication in VANETs," Electronics, 11(3), p.488, 2022.

[24] Alkadi, O., Moustafa, N. and Turnbull, B., "A review of intrusion detection and blockchain applications in the cloud: Approaches, challenges and solutions," IEEE Access, 8, pp.104893-104917, 2020.

[25] Liang, J. and Ma, M., "Co-maintained database based on blockchain for idss: A lifetime learning framework," IEEE Transactions on Network and Service Management, 18(2), pp.1629-1645, 2021.

[26] Safaldin, Mukaram, Mohammed Otair, and Laith Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks," Journal of ambient intelligence and humanized computing Vol. 12, no. 2 (2021): 1559-1576, 2021.

[27] Ayyarao, Tummala SLV, N. S. S. RamaKrishna, Rajvikram Madurai Elavarasan, Nishanth Polumahanthi, M. Rambabu, Gaurav Saini, Baseem Khan, and Bilal Alatas, "War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization," IEEE Access 10 (2022): 25073-25105, 2022.

[28] Ayyarao, Tummala SLV, and Polamarasetty P. Kumar, "Parameter estimation of solar PV models with a new proposed war strategy optimization algorithm," International Journal of Energy Research 46, no. 6 (2022): 7215-7238, 2022.

[29] Zhang, T. and Zhu, Q., "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," IEEE Transactions on Signal and Information Processing over Networks, 4(1), pp.148-161, 2018.

[30] Belenko, V., Krundyshev, V. and Kalinin, M., "September. Synthetic datasets generation for intrusion detection in VANET," International conference on security of information and networks (pp. 1-6), 2018.

[31] Shu, J., Zhou, L., Zhang, W., Du, X. and Guizani, M., "Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach," IEEE Transactions on Intelligent Transportation Systems, 22(7), pp.4519-4530, 2020.

Content-based Image Retrieval using Encoder based RGB and Texture Feature Fusion

Charulata Palai[1], Pradeep Kumar Jena[2*], Satya Ranjan Pattanaik[3], Trilochan Panigrahi[4], Tapas Kumar Mishra[5]
Department of Computer Science and Engineering, NIST Institute of Science and Technology, Berhampur-761008, India[1, 2]
School of Computing, Gandhi Institute for Technology, Bhubaneswar-752054, India[3]
Department of ECE, National Institute of Technology, Goa-403401, India[4]
Department of Computer Science and Engineering, SRM University-AP, Amaravati-522240, India[5]

Abstract—**Recent development of digital photography and the use of social media using smartphones has boosted the demand for image query by its visual semantics. Content-Based Image Retrieval (CBIR) is a well-identified research area in the domain of image and video data analysis. The major challenges of a CBIR system are (a) to derive the visual semantics of the query image and (b) to find all the similar images from the repository. The objective of this paper is to precisely define the visual semantics using hybrid feature vectors. In this paper, a CBIR system using encoded-based feature fusion is proposed. The CNN encoding features of the RGB channel are fused with the encoded texture features of LBP, CSLBP, and LDP separately. The retrieval performance of the different fused features is tested using three public datasets i.e. Corel-lK, Caltech, and 102flower. The result shows the class properties are better retained using the LDP with RGB encoded features, this helps to enhance the classification and retrieval performance for all three datasets. The average precision of Corel-lK is 94.5% and it is 89.7% for Caltech, and 88.7% for the 102flower. The average f1-score is 89.5% for Caltech, and 88.5% for the 102flower. The improvement in the f1-score value implies the proposed fused feature is more stable to deal the class imbalance problem.**

***Keywords*—CBIR; CNN Encoded Feature; LBP; CSLBP; LDP; feature fusion**

I. INTRODUCTION

Content-based image retrieval (CBIR) is the technique to retrieve similar images from a large image database using visual characteristics such as color, shape, structure, Zernike values, and histogram of the images[1, 2]. Nowadays it has an inevitable requirement in various application areas such as video surveillance, medical image retrieval, crime detection, military surveillance, remote sensing applications, the textile industry etc. [3-6]. The efficiency of the CBIR system greatly depends upon the visual feature selection. The high-level semantic features [3,7] of an image are its color, shape, structure, Zernike values, and histogram are used for manual image annotation and are less biased with noise [8,9]. Features represented using the spatial layout of the pixels within an image patch are referred as low-level features or local descriptors [10-12]. Some of the popular low-level image descriptors are Local Binary Patterns (LBP) [13-15], Orthogonal-Combination of Local Binary Patterns (OC-LBP), Center-Symmetric Local Binary Patterns (CS-LBP), Local Ternary Patterns (LTP), Local Directional Patterns (LDP)[15], Scale-Invariant Feature Transform (SIFT) [16] are used for image retrieval. The performance of a unique texture feature varies with different datasets. The major limitation of the texture feature is directly mapping the texture image to its histogram [1, 9, 17-19], which is represented on a scale of 0 to 255, so that all the information learned from the patches of images are not well preserved. With the implementation of Deep learning features a new breakthrough is achieved in the field of computer vision and its applications. It uses the Convolutional neural networks (CNNs) features [14, 20-23] as the image descriptor. The Deep learning technique requires adequate images for its training. The several layers of the CNN encoder represent the image features at different levels [11]. The lower layers contain the detailed image features, whereas the higher layers present the semantic information of the image [10, 11]. The fully connected layer extracts discriminative image features using an order-less quantization approach. Finally, these features are mapped to the class label using the dimension reduction technique and soft-max pooling [10, 24].

An effectual feature extraction technique precisely describes the image contents. It also helps to maintain a distinctive signature for the images of different classes. In recent years image retrieval using feature fusion has been emphasized by many researchers[3, 8] to build a more powerful image descriptor using the feature fusion technique [7,10,23,25-27]. These are more sensitive to noise and image resolution. Moreover mapping the low-level image features to the high-level visual semantics is challenging [7, 8, 28, 29]. Thus, there is a need to design an enhanced CBIR system.

In this work, a deep-learning feature fusion framework is proposed, where the auto-encoding features of the RGB channels are fused with the auto-encoding feature of the texture image. Here two different CNN models are trained independently. The first model usages the RGB channels data, which learns the spatial image information using automatic encoding. The second model usages the texture image data for the training to learn the auto-encoder-based texture features. The spatial and texture features extracted by CNN encoders are fused together to provide more precise feature descriptors for the image. The texture feature of an image i.e. the histogram of the texture image is biased by the background image textures, which impedes the learning ability of the classifier [9, 17]. Textures of similar images are expected to be alike. More effective learning can be possible from the texture image set, as the CNN uses the batch mode for the training. For the extensive analysis of the proposed fusion framework, a CBIR system is developed. Here the encoding features of three different textures such as LBP, CSLBP, and LDP are fused with the RGB channel encoding feature individually. The classification and retrieval performance of the different fused features are presented. These are also compared with encoding features of only RGB channels. The model is tested for three different datasets such as Corel-lK, Caltech, and 102flower. It is observed that the classification result of the LDP_RGB fusion outperforms the results of LBP_RGB, CSLBP_RGB, and RGB. Moreover, the proposed fusion features preserve more class-oriented properties, so that the retrieval rate is enhanced. The performance analysis for the top 80 images retrieval using the proposed auto-encoder-based feature fusion and the auto-encoder-based RGB channel feature are shown in the result section. The retrieval rate using LDP_RGB fusion also surpasses all the other methods discussed.

The major contributions of this work are mentioned below:

- A new enhanced feature fusion technique is used, where the texture image and RGB channel image features are fused.
- The auto-encoding features of the texture and RGB channels are extracted by two different CNN models to save the low variance pixel information of the texture image.
- The CBIR model is tested for three different textures i.e. LBP, CSLBP, and LDP textures with RGB channel encoding feature.
- The model is tested with three different datasets such as Corel-lK, Caltech, and 102flower.
- Improvement in the f1-score implies the proposed feature descriptor handles the class imbalance issue more precisely.
- The retrieval result is enhanced with the fusion of LDP and RGB encoded features.

The rest part of the paper is arranged in the following order: Section II presents a review of feature fusion and CBIR system. Section III shows the proposed feature fusion model, CNN encoding architecture, and performance evaluation metrics. The detailed results are shown in Section IV i.e. the results and discussions. The conclusion of the work is presented in Section V.

II. Related Work

Kayhan, N., et al. [1] build a weighted feature-based CBIR system using modified local binary patterns (MLBP), local neighbourhood differences patterns (LNDP), filtered gray level co-occurrence matrix (GLCM), and the quantization color histogram features. Khan, U. A., et al. [2] used hybrid classification model using three color moments, Haar Wavelet, Daubechies Wavelet and Bi-Orthogonal wavelets features. They have used genetic algorithm (GA) and SVM classification and L2 Norm is used for the similarity measure. Kashif, M., et al. [3] proposed a hybrid image descriptor using local ternary pattern, local phase quantization, and discrete wavelet transform. They used joint mutual information (JMI) based feature selection to derive the optimal feature for effective image retrieval. Carvalho, E. D., et al. [4] proposed a histopathological breast image classification model using phylogenetic diversity indexes. They have also used the phylogenetic diversity indexes to rank the images. Authors claim, it outperforms XGBoost, random forest, and support vector machine. Choe, J., et al. [5] proposed a medical image retrieval model for interstitial lung disease diagnosis using the deep learning features of CT images. Pradhan, J., et al. [7] proposed a regions-of-attention-based feature fusion technique for image retrieval, here authors used multi-directional texture features with spatial correlation-based color features to derive the image semantics. Pathak, D., et al. [9] proposed a retrieval system by concatenating the deep learning GoogleNet features with the hue, saturation, and intensity features of the HIS image, and Histogram of orientated gradient (HOG) feature of the RGB image. Here the authors claim this technique is used to reduce information loss due to image resizing. A. Latif, et al. [10] presented a comprehensive review of the recent development and the state-of-the-art CBIR systems. The study explored the major concepts of CBIR like image representation, image retrieval, low-level feature extraction, and recently used semantic deep-learning approaches, it also includes future research directions in CBIR. M. Sotoodeh, et al. [17] presented a local texture descriptor referred as Color Radial Mean Local Binary Pattern (CRMLBP). The CRMLBP is computed for the sign-difference, magnitude-difference, and central gray value patterns in the RGB color space and their histograms are concatenated. The feature weights are optimized using Particle Swam Optimization (PSO) technique. The performance of this feature vector is tested with various datasets such as Wang, Holidays, Corel data. Sampathila, N., et al. [18] presented an image retrieval method using Grey-level co-occurrence-based Haralik's features and histogram-based cumulative distribution function (CDF) for the brain MRI image retrieval. Here the KNN approach is used to find the distance between the query image and other images. Khan, M. A., et al. [19] proposed an intelligent human action recognition system using Hand-crafted and deep convolutional neural network features fusion. Here the histogram of oriented gradients (HoG) and deep features are fused. A multi-class support vector machine (M-SVM) is used for the classification. Ma, W., et al. [22] suggested a cloud-based privacy-preserving image retrieval service using deep convolutional features with from the encrypted image. For image encryption, a hybrid encryption method is adopted. Wang, S. H., et al. [23] suggested deep feature fusion technique using graph convolutional network and convolutional neural network features for Covid-19 classification. Here they used the CT images to test their model performance. L. T. Alemu, et al. [25] proposed a multi-feature fusion-based CBIR system, where various hand-crafted features with deep NN features and membership score is applied based on their probabilistic distribution. Then an incremental nearest neighbour (NN) selection is used to implement k-NN for dynamic query selection. Wang, W., et al. [26] presented a two-stage CBIR model using the fusion of global and local feature. Authors use a sparse coding for the sparse representation of the local features followed by feature pooling and the Euclidean distance measure is used to find the similarity between the sparse feature vectors. Bella, M. I. T. et al. [28] proposed the image retrieval system using information fusion technique, where the GLCM and HSV color moment features are fused the model is tested with Corel-1K, Corel-5K, and Corel-10K datasets.

Table I presents a survey on the different feature fusion techniques used for image classification and retrieval. However, there is a scope to define a better image descriptor using the strength of the texture feature with the deep CNN feature. In this work, intend to define a more precise feature vector by combining the CNN-encoded texture feature with the encoded RGB channel feature.

TABLE I. SURVEY TABLE FEATURE FUSION AND CBIR SYSTEM

Sl. No.	Research Study / Year	Feature Fusion Method	Classification / Retrieval	Database
1	H. Wang / 2020 [6]	Visual saliency based multi-feature fusion	Retrieval	Corel-1K
2	Pradhan, J. / 2021 [7]	Texture, and spatial correlation-based color features fusion	Retrieval	Corel, and GHIM
3	K. T. Ahmed /2021 [16]	Spatial color with shaped features fusion	Retrieval	Caltech, Corel, COIL, and ALOT
4	Khan, M. A. /2020 [19]	Hand-crafted, HoG, and deep features fusion	Classification	Weizmann, UCF11, IXMAS
5	Wang, S. H. /2021 [23]	Deep feature fusion	Classification	Chest CT images
6	Wang, W. /2022 [26]	Global and Local features fusion	Retrieval	Coil20, Caltech
7	Wang, Z. / 2019 [27]	Deep features, morphological features, texture, and density feature fusion	Classification	400 mammograms pathologica dataset
8	Bella, M. I. T./2019 [28]	Fused Feature of GLCM and HSV Color Moments features.	Retrieval	Corel-1K, Corel-5K, and Corel-10K

III. PROPOSED MODEL

In the case of Deep learning, the image features are fetched automatically using a CNN encoder. The features extracted from the RGB channel carry more information than the Gray-scale image, as it learns from three channels R, G, and B coherently. At the same time, computational complexity increases. Moreover, information stored in all three channels is highly correlated, which impedes the learning rate. In this work, a feature fusion technique is proposed where the CNN-encoded feature of the texture image is fused with the encoded feature of RGB channels. Two different CNN encoders are used to derive the texture and RGB features from an image. The motivation behind two different CNN encoders instead of adding the texture image in the 4th channel in addition to the R, G, and B is that the range of the pixel values of the texture image is comparatively smaller than the pixel values of the R, G, and B channels. So the texture information will not be suppressed during the recursive MAX pooling and ReLU operations.

A. LBP Texture Image

The LBP texture of a 3 x 3 pixel block is achieved by thresholding the pixel values of the neighbours with its center pixel into binary values, where the value is 1 if the value of the neighbour pixels is greater or equal to the value of the center pixel, otherwise 0. The values of all the 8 neighbours are stored in an unsigned-byte form, here the range varies from 0 to 255. Eq. (1) shows the calculation of the LBP texture image, where R is the radius of the circle [14].

$$LBP_{P,R} = \sum_{p=0}^{P-1} S(g_p - g_c)\, 2^P \qquad (1)$$

$$where \;\; S(z) = \begin{cases} 1, & if\ z \geq 0 \\ 0, & otherwise \end{cases}$$

B. CSLBP Texture Image

Center-Symmetric Local Binary Patterns are produced by computing the thresholding difference of pixel values with their symmetrically opposite pixels with respect to the canter of a pixel block. Here the thresholding difference is a smaller integer value T. The CSLBP labels generate shorter histograms, which is a more stable feature for the flat image regions. Eq. (2) represents the calculation of the CSLBP texture image [15].

$$CSLBP_{P,R,T(x,y)} = \sum_{p=0}^{\left(\frac{P}{2}\right)-1} S\left(g_p - g_{p+\left(\frac{P}{2}\right)} - T\right) 2^P \qquad (2)$$

$$where \;\; S(z) = \begin{cases} 1, & if\ z \geq 0 \\ 0, & otherwise \end{cases}$$

C. LDP Texture Image

The LDP pattern illustrates the response values of all eight directional edges of a center pixel. It is calculated using the Kirsch masks in the eight different orientations i.e. (M0 ~ M7) with respect to a 3x3 pixel block, and Eq. (3) and Eq. (4) show the LDP value calculation at a point (x, y).

$$m_i = \sum_{l=-1}^{1} \sum_{k=-1}^{1} I(x+l, y+k) \times M_i(l,k) \qquad (3)$$

$$LDP_{x,y}(m_{0,}\dots, m_7) = \sum_{i=0}^{7} S(m_i - m_k) \times 2^P \qquad (4)$$

Fig. 1 shows the Original images of Corel-lK, Caltech, and 102Flower datasets with their equivalent LBP, CSLBP, and LDP texture images. Here four images are shown from each dataset and the *Image Ref* is a combination of the folder name and the image name.

D. Auto-encoder-based CNN Feature

The deep CNN feature of an image is generated using an automatic encoding technique. The image feature is learned through batch mode training, hence it is expected that the feature preserves the class information. As the layers of CNN architecture are densely connected, the learning becomes faster with automatic weight adjustment for a particular class using supervised learning.

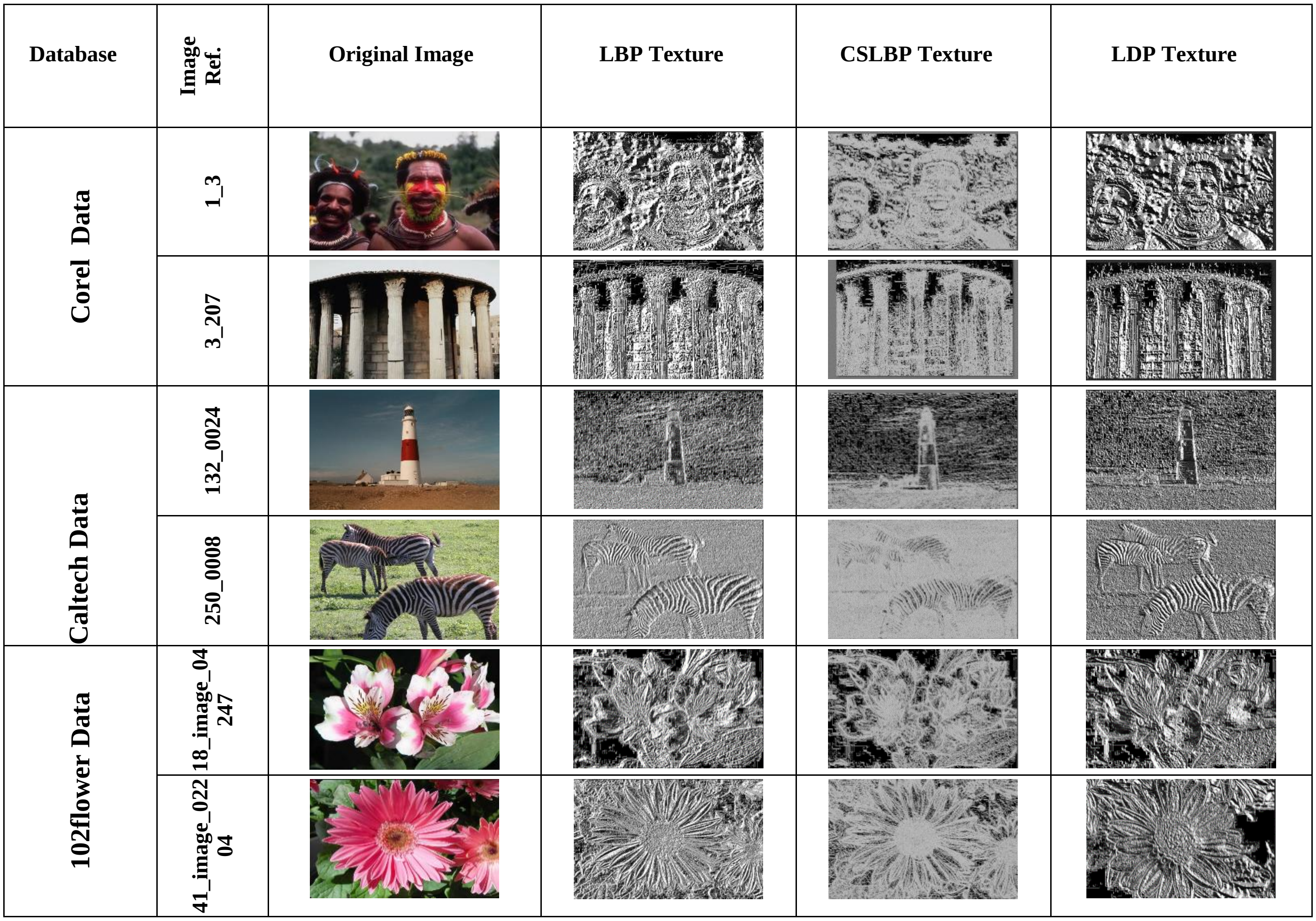


Fig. 1. The Original RGB image with their equivalent LBP, CSLBP and LDP texture image.

E. Feature Fusion Model

Fig. 2 shows the proposed model, here the size of the input image is 512 x 512 for both the CNNs i.e. the RBG and the texture input. The CNN architecture consists of seven layers and each layer contains a convolution operation followed by the ReLU and MAX pooling operations. Non-linearity property is introduced to the convolution output with the ReLU activation function. Whereas the image size reduction is done by the MAX pooling with each convolution operation. The flattened layer is used to reduce the image to a single-dimension feature vector of size 1 x 1024. Further dimension reduction is done with four fully connected layers. The soft-max operation is used to calculate the class label from the feature map using the energy function.

$$(I * f)_{x,y} = \sum_{s=1}^{H} \sum_{t=1}^{W} f_{s,t} . I_{x+s-1,y+t-1} + b \quad (5)$$

$$ReLU(x) = Max(0, x) \quad (6)$$

$$E(w, b) = \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(x_i)\big) + \alpha R(w) \quad (7)$$

Where:

L = model loss parameter.

R = regularization factor used to deal with the model complexity.

α = regularization strength control parameter.

The cross-entropy loss is determined as the penalty value in each iteration using that energy function. Eq. (5) represents the convolution operation at point a (x, y) of an image I used the filter f, where the H and W represent the height and width of the image. The ReLU operation is defined using Eq. (6). Eq. (7) represents the regularized training error of an instance. Eq. (8) represents the sigmoid function Si used to map the output value within (0, 1). The cross-entropy loss for each iteration is defined by Eq. (9).

$$S_i(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

$$LogLoss = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} log P_{ij} \quad (9)$$

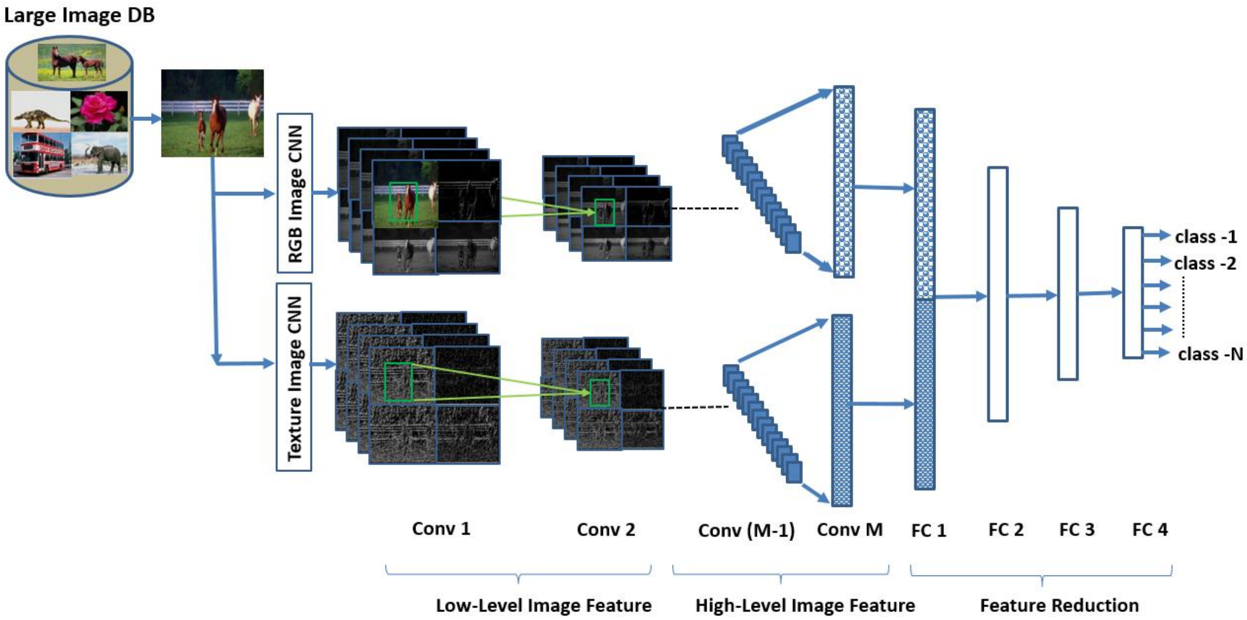


Fig. 2. Block diagram of the proposed CNN encoder-based RGB and texture features fusion.

Where N shows the number of samples and M is the number of labels, the y_{ij} represents if the label j is correctly classified as, for the instance, i. Here P_{ij} is the probability value of the model that assigns label j to the instance i.

The proposed feature fusion model using the CNN feature of the RGB image and the CNN feature of the texture image uses the standard learning rate with an early stop parameter value of 0.99. The model training is done using 80:20 holdout validation. Here a GTX 1650 graphics system with 16 GB RAM is used for the training and testing of the proposed model.

F. Performance Measures

The performance of the proposed feature fusion is evaluated using parametric quantifiers such as precision, recall, and f1-score [13], which are defined below using Eq. (10), Eq. (11), and Eq. (12) respectively.

$$Precision = \frac{True^{+}}{True^{+} + False^{+}} \tag{10}$$

$$Recall = \frac{True^{+}}{True^{+} + False^{-}} \tag{11}$$

$$f1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

Here the true positive (True+) value shows the number of images correctly identified into their belonging class by the system. The false positive (False+) shows the number of images falsely recognized by the system, and the false negative (False-) shows the number of images falsely rejected by the system. The precision shows the number of images correctly identified into their belonging class with respect to the total number of images identified by the system. Whereas recall represents the number of images correctly identified for a class with respect to all the images belonging to that class. Hence the average recall value is a significant performance measure for a retrieval system. The Caltech and 102flower datasets have a different number of total images in different classes. The harmonic mean of these classes i.e. f1-Score is also presented in addition to the precision [13, 29]. The receiver operating characteristics (ROC) curve, which is plotted using the true-positive rate vs. false-positive rate, illustrates graphically the classifier's performance. The City-block distance measure shown in the Equation (13) is used to measure the similarity between the images.

City-block distance measure:

$$D_{CT} = \sum_{i=0}^{L-1} \left| F_i^q - F_i^t \right| \tag{13}$$

Where:

F_i^q = feature vector of the query image

F_i^t = feature vector of the database image

IV. Results and Discussion

The results of the proposed CBIR model using encoded texture feature fusion are discussed in this section. Here the CNN-based auto-encoding features of the RGB channels are fused with the auto-encoding features of three different texture features i.e. LBP, CSLBP, and LDP. The image retrieval model is tested with three different datasets such as Corel-lK, Caltech, and 102Flower. To avoid the extensive processing time, selective 15 classes of the 102Flower dataset have been considered. The precision, recall, and f1-score of each class are presented for all three datasets. The ROC curve shows the overall classification performance using the CNN encoding features of RGB, RGB_LBP, RGB_CSLBP, and RGB_LDP. The average retrieval performance is shown separately for all three datasets using all the above-discussed four encoding features for top 80 image retrieval. The class-wise retrieval performances are illustrated with the bar graph for all four encoding features. The detailed analysis results of individual classes are discussed in the sub-sections below.

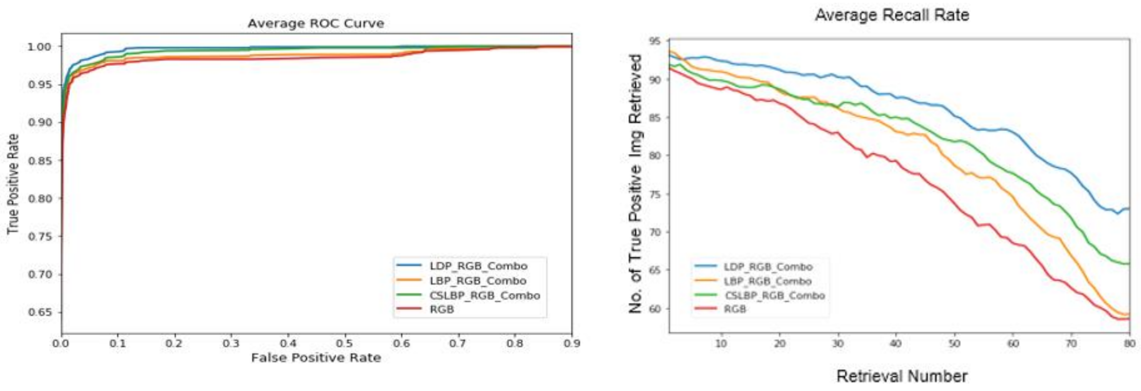
A. Results Analysis of Corel-lK

Table II shows the performance analysis of the Corel-1K dataset maximum average precision value is 94.5% using the LDP_RGB encoder feature. It is 94.2% using LBP with RGB, 94.3% for CSLBP with RGB, and 94.2% using the RGB encoder feature. The classification performance is presented using the ROC curve in Fig. 3(a).

The Recall and f1-score are 94.4%, and 94.5% respectively using the LDP_RGB feature, which is maximum in comparison to the other features. Though there is a small difference in the classification rate, the average retrieval rate is significantly enhanced using the LDP_RGB feature in comparison to the other features for retrieving the top 80 images shown in Fig. 3(b), and the class-wise retrieval analysis is shown in Fig. 4 for the top 10 images. In this dataset, each class consists of 100 images, so there is no major difference in the precision and f1-score values.

TABLE II. CLASS-WISE PERFORMANCE OF THE COREL-1K DATASET

Class Name	**LDP_RGB_Combo**			**LBP_RGB_Combo**			**CSLBP_RGB_Combo**			**RGB**			**No. of Images**
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
African Tribes	0.94	0.9	0.92	0.92	0.91	0.91	0.88	0.96	0.92	0.87	0.95	0.91	100
Beaches	0.93	0.91	0.92	0.94	0.89	0.91	0.92	0.89	0.9	0.84	0.95	0.89	100
Buildings	0.88	0.87	0.87	0.97	0.87	0.92	0.93	0.89	0.91	0.96	0.85	0.9	100
Buses	0.99	0.97	0.98	0.97	0.96	0.96	0.99	0.97	0.98	0.97	0.97	0.97	100
Dinosaurs	0.99	1	1	1	0.97	0.98	1	1	1	1	1	1	100
Elephants	0.98	0.97	0.97	0.95	0.95	0.95	0.94	0.9	0.92	0.93	0.92	0.92	100
Flowers	0.99	1	1	0.96	1	0.98	1	1	1	1	1	1	100
Horses	0.99	0.94	0.96	0.95	0.99	0.97	0.97	1	0.99	0.98	1	0.99	100
Mountains	0.85	0.94	0.9	0.86	0.95	0.9	0.83	0.91	0.87	0.92	0.87	0.89	100
Foods	0.91	0.94	0.93	0.9	0.92	0.91	0.97	0.89	0.93	0.95	0.88	0.91	100
Avg. Accuracy %	**94.5**	**94.4**	**94.5**	94.2	94.1	93.9	94.3	94.1	94.2	94.2	93.9	93.8	



(a) ROC Curve

(b) Average Recall Rate of the top 80 Image

Fig. 3. Analysis of the Corel 1K images using RGB and Texture with RGB CNN feature.

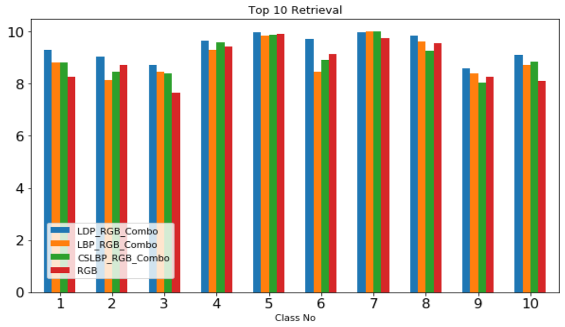
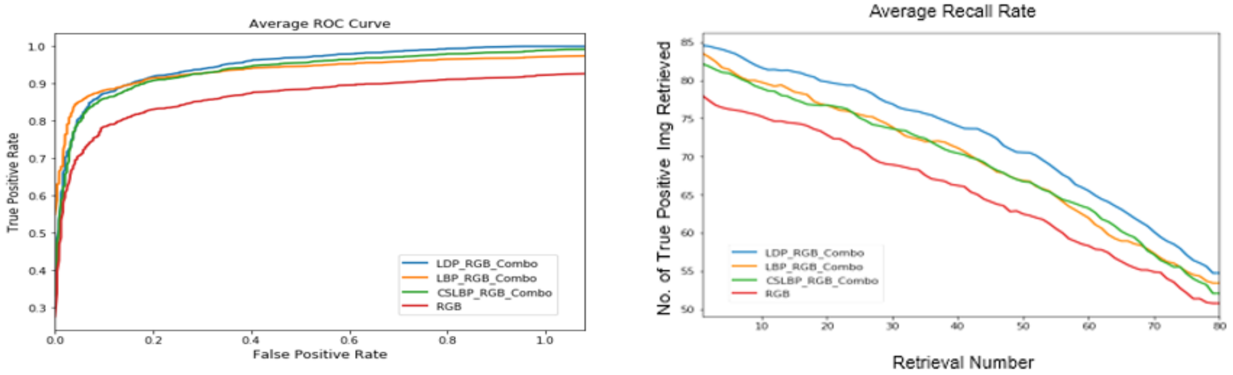


Fig. 4. Class-wise retrieval of the top 10 images of the Corel 1K dataset.

TABLE III. CLASS-WISE PERFORMANCE OF THE CALTECH DATASET

Class Name	LDP_RGB_Combo			LBP_RGB_Combo			CSLBP_RGB_Combo			RGB			No. of Images
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
Backpack	0.88	0.93	0.9	0.88	0.91	0.9	0.84	0.94	0.88	0.98	0.86	0.92	151
Billiards	0.91	0.88	0.9	0.92	0.93	0.92	0.84	0.91	0.88	0.91	0.81	0.86	278
Bonsai-101	0.9	0.87	0.88	0.9	0.89	0.89	0.96	0.85	0.9	0.98	0.82	0.89	122
Boxing-glove	0.86	0.97	0.91	0.94	0.89	0.91	0.81	0.94	0.87	0.87	0.9	0.89	124
Eiffel-tower	0.88	0.9	0.89	0.91	0.89	0.9	0.84	0.92	0.87	0.9	0.8	0.85	83
Fern	0.87	0.94	0.9	0.9	0.89	0.89	0.91	0.95	0.93	0.89	0.85	0.87	110
Fighter-jet	0.83	0.84	0.83	0.85	0.88	0.87	0.89	0.81	0.85	0.79	0.78	0.79	99
Fire-truck	0.94	0.87	0.91	0.87	0.85	0.86	0.97	0.6	0.74	0.98	0.89	0.93	118
Gorilla	0.93	0.89	0.91	0.85	0.9	0.87	0.89	0.9	0.89	0.89	0.9	0.9	212
Iris	0.81	0.85	0.83	0.79	0.85	0.82	0.91	0.86	0.89	0.56	0.91	0.69	108
Light-house	0.86	0.9	0.88	0.89	0.88	0.88	0.93	0.81	0.87	0.83	0.92	0.87	190
Sunflower-101	0.99	0.95	0.97	0.97	0.95	0.96	0.97	0.95	0.96	0.99	0.93	0.95	80
Watch-101	0.95	0.93	0.94	0.95	0.93	0.94	0.91	0.9	0.9	0.81	0.95	0.87	201
Waterfall	0.91	0.87	0.89	0.81	0.83	0.82	0.87	0.81	0.84	0.95	0.79	0.86	95
Zebra	0.93	0.85	0.89	0.91	0.81	0.86	0.71	0.94	0.81	1	0.84	0.92	96
Avg. Accuracy%	**89.7**	**89.6**	**89.5**	88.9	88.5	88.6	88.3	87.3	87.2	88.9	86.3	87.1	



(a) ROC Curve

(b) Average Recall Rate of the top 80 Image

Fig. 5. Analysis of the Caltech images using RGB and texture with RGB CNN feature.

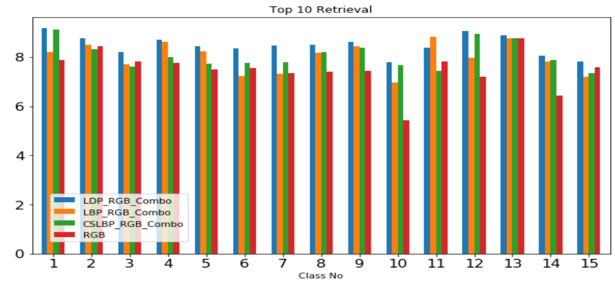


Fig. 6. Class-wise retrieval of the top 10 images Caltech dataset.

B. Results Analysis of Caltech

The result analysis of the Caltech dataset is shown in Table III, in this case, value of maximum average precision is 89.7% using the LDP_RGB encoder feature. Whereas it is 88.9% using LBP with RGB, 88.3% for CSLBP with RGB, and 88.9% using RGB encoder feature. The classification performance is presented using the ROC curve in Fig. 5(a). The Recall and f1-score are 89.6%, and 89.5% respectively with the LDP_RGB feature, which is more in comparison to the other encoding features.

Though there is a visible difference in the classification rate, the average retrieval rate is also enhanced using the LDP_RGB feature in comparison to the other features for retrieving the top 80 images shown in Fig. 5(b), and Fig. 6 show the class-wise retrieval analysis for the top 10 images. As in this dataset, the number of images in the different classes varies the precision, and f1-score values are also different, moreover these values are more stable using the LDP_RGB encoder feature.

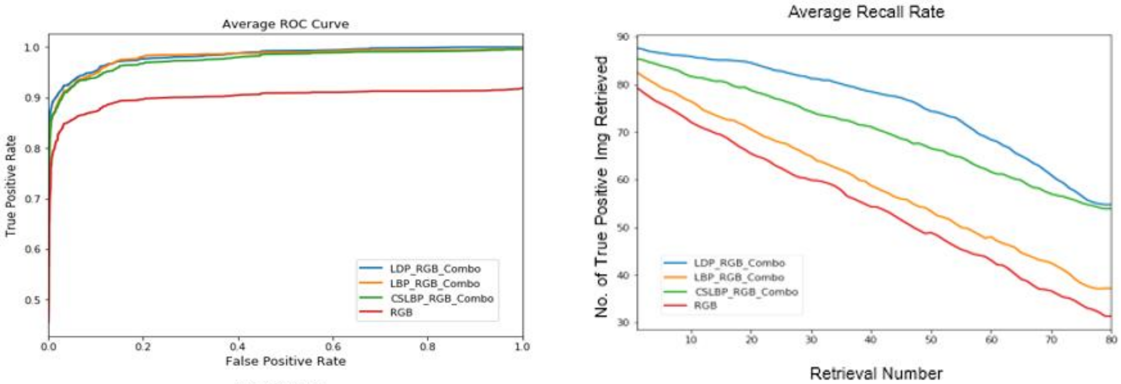
C. Results Analysis of 102Flower

The classification performance of the 102Flower dataset is shown in Table IV. Here 15 classes are selected, and the name of the flower and the number of images available for each class is mentioned in the table in the first and last columns respectively. Here the value of maximum average precision is 88.7% using the LDP_RGB encoder feature. Whereas it is 87.9% using the LBP with RGB, 85.6% for CSLBP with RGB, and 84.5% using the RGB encoder feature. The classification performance of the 102Flower dataset is presented using the ROC curve in Fig. 7(a).

There is a significant difference in the classification rate. The result shown in Fig. 7(b) claims that average retrieval rate using LDP_RGB feature is better than other fusion techniques for retrieving the top 80 images. Fig. 8 shows the class-wise retrieval of the top 10 images. The value of the f1-score is also enhanced using the LDP_RGB feature. Table V presents a state-of-art, where the performance of five other works available in the literature, using the same dataset are compared with the proposed feature fusion model. It shows the accuracy of Corel-1k is 94.5% and Caltech256 is 89.7%. A significant improvement is achieved for both datasets using the proposed feature fusion model.

TABLE IV. CLASS-WISE PERFORMANCE OF THE CALTECH DATASET

Class Name	**LDP_RGB_Combo**			**LBP_RGB_Combo**			**CSLBP_RGB_Combo**			**RGB**			**No. of Images**
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
Purple Coneflower	0.99	0.88	0.93	0.95	0.9	0.93	0.9	0.83	0.86	0.92	0.8	0.85	84
Peruvian Lily	0.94	0.83	0.88	0.76	0.81	0.79	0.86	0.83	0.84	0.9	0.7	0.79	81
Cape Flower	0.85	0.88	0.86	0.86	0.85	0.85	0.87	0.82	0.84	0.9	0.83	0.86	106
Barbeton Daisy	0.88	0.84	0.86	0.86	0.85	0.85	0.78	0.86	0.82	0.94	0.85	0.89	126
Sword Lily	0.79	0.84	0.82	0.84	0.84	0.84	0.92	0.84	0.87	0.8	0.8	0.8	129
Pink-Yellow Dahlia	0.92	0.9	0.91	0.85	0.98	0.91	0.8	0.83	0.81	0.74	0.81	0.77	108
Californian Poppy	0.94	1	0.97	0.91	0.99	0.95	0.87	0.87	0.87	0.76	0.96	0.85	101
Azalea	0.85	0.89	0.87	0.84	0.85	0.84	0.96	0.85	0.91	0.79	0.85	0.82	95
Rose	0.93	0.87	0.9	0.99	0.84	0.9	0.75	0.89	0.81	0.75	0.84	0.79	170
Lotus	0.87	0.93	0.9	0.95	0.9	0.92	0.88	0.88	0.88	0.79	0.9	0.84	136
Anthurium	0.8	0.83	0.82	0.77	0.84	0.8	0.71	0.84	0.77	0.84	0.86	0.85	104
Frangipani	0.96	0.95	0.95	0.94	0.97	0.95	0.88	0.84	0.86	0.82	0.88	0.85	165
Hibiscus	0.8	0.87	0.83	0.86	0.85	0.85	0.9	0.82	0.85	0.85	0.8	0.83	130
Cyclamen	0.9	0.86	0.88	0.92	0.88	0.9	0.93	0.84	0.88	0.95	0.82	0.88	152
Foxglove	0.89	0.89	0.89	0.88	0.88	0.88	0.83	0.84	0.84	0.93	0.82	0.87	161
Avg. Accuracy%	**88.7**	**88.4**	**88.5**	87.9	88.2	87.7	85.6	84.5	84.7	84.5	83.5	83.6	



(a) ROC Curve

(b) Average Recall Rate of the top 80 Image

Fig. 7. Analysis of the 102Flowerset images using RGB and Texture with RGB CNN feature.

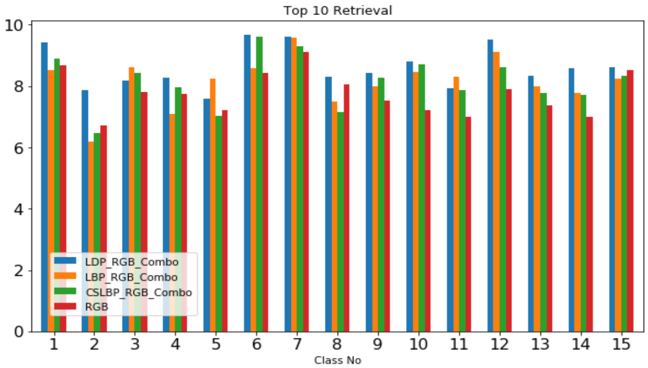


Fig. 8. Class-wise retrieval of the top 10 images of the 102 flowerset dataset.

TABLE V. COMPARISON OF CLASSIFICATION RESULTS OF EXISTING METHODS AND THE PROPOSED METHOD

References	Dataset	Methods Used	Accuracy%
Kayhan, N.[1]	Corel-lK	Weighted Color & Texture feature fusion	82.52%
Khan, U. A.[2]	Corel-lK	Wavelet features with GA and SVM	90.5%
K. T. Ahmed [16]	Corel-lK	Color and Object feature fusion	92.3%
K. T. Ahmed [16]	Caltech	Color and Object feature fusion	71.3.%
M. I. Thusnavis Bella [28]	Corel-lK	HSV Color and GLCM feature	83.3%
ElAlami, M. E. [30]	Wang-lK	GLCM feature using ANN	76.1%
Proposed Model	Corel-lK	LDP and RGB encoded feature	94.5%
Proposed Model	Caltech	LDP and RGB encoded feature	89.7%

V. CONCLUSION

This paper proposed a CBIR model using the feature fusion technique. Here the CNN-encoded features of the image are fused with the encoded features of the RGB image. As the range of pixel values in the texture image is comparatively smaller than that of the RGB image, two different encoders are employed to extract the CNN features separately. These two features are fused to define a more significant image descriptor.

The proposed model is tested with three public datasets i.e. Corel-lK, Caltech, and 102flower. It is observed that the classification performance is improved by the proposed feature fusion model as compared to the RGB channel encoding feature. The result shows the performance of the LDP with RGB feature fusion is better with respect to the LBP with RGB and CSLBP with RGB features. There is a significant improvement in the retrieval system for the top 10 as well as top 80 image retrieval. Moreover, the enhancement of the f1-score using the proposed feature fusion technique illustrates the class property is better retained using the fused features. The f1-score value improved significantly using the encoder-based LDP with RGB feature fusion for the dataset having class imbalance issues such as Caltech, and 102flower. In future, the model can be tested using the fusion of other textures like LTP, GLCM.

REFERENCES

[1] Kayhan, N., & Fekri-Ershad, S. (2021). Content based image retrieval based on weighted fusion of texture and color features derived from modified local binary patterns and local neighborhood difference patterns. *Multimedia Tools and Applications*, *80*(21), 32763-32790.

[2] Khan, U. A., Javed, A., & Ashraf, R. (2021). An effective hybrid framework for content based image retrieval (CBIR). *Multimedia Tools and Applications,* 80(17), 26911-26937.

[3] Kashif, M., Raja, G., & Shaukat, F. (2020). An efficient content-based image retrieval system for the diagnosis of lung diseases. *Journal of digital imaging,* 33(4), 971-987.

[4] Carvalho, E. D., Antonio Filho, O. C., Silva, R. R., Araujo, F. H., Diniz, J. O., Silva, A. C., ... & Gattass, M. (2020). Breast cancer diagnosis from histopathological images using textural features and CBIR. *Artificial intelligence in medicine,* 105, 101845.

[5] Choe, J., Hwang, H. J., Seo, J. B., Lee, S. M., Yun, J., Kim, M. J., ... & Kim, B. (2022). Content-based image retrieval by using deep learning for interstitial lung disease diagnosis with chest CT. *Radiology,* 302(1), 187-197.

[6] H. Wang, Z. Li, Y. Li, B. B. Gupta, and C. Choi, “Visual saliency guided complex image retrieval,” *Pattern Recognition Lett.*, vol. 130, pp. 64–72, 2020, doi: 10.1016/j.patrec.2018.08.010.

[7] Pradhan, J., Pal, A. K., Banka, H., & Dansena, P. (2021). Fusion of region based extracted features for instance-and class-based CBIR applications. *Applied Soft Computing,* 102, 107063.

[8] Salih, S. F., & Abdulla, A. A. (2022). An effective bi-layer content-based image retrieval technique. *The Journal of Supercomputing,* 1-24.

[9] Pathak, D., & Raju, U. S. N. (2022). Content-based image retrieval for super resolutioned images using feature fusion: Deep learning and hand crafted. *Concurrency and Computation: Practice and Experience,* e6851.

[10] A. Latif et al., “Content-based image retrieval and feature extraction: A comprehensive review,” *Math. Probl. Eng.,* vol. 2019, 2019, doi: 10.1155/2019/9658350.

[11] W. Yu, K. Yang, H. Yao, X. Sun, and P. Xu, “Exploiting the complementary strengths of multi-layer CNN features for image,” *Neurocomputing,* vol. 237, pp. 235–241, 2017, doi: 10.1016/j.neucom.2016.12.002.

[12] G. M. Galshetwar, L. M. Waghmare, A. B. Gonde, and S. Murala, “Local energy oriented pattern for image indexing and retrieval,” *J. Vis.*

Commun. Image Represent., vol. 64, p. 102615, 2019, doi: 10.1016/j.jvcir.2019.102615.

[13] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, "Medical image retrieval using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 8–20, 2017, doi: 10.1016/j.neucom.2017.05.025.

[14] A. Khatami, M. Babaie, H. R. Tizhoosh, A. Khosravi, T. Nguyen, and S. Nahavandi, "A sequential search-space shrinking using CNN transfer learning and a Radon projection pool for medical image retrieval," *Expert Syst. Appl.*, vol. 100, pp. 224–233, 2018, doi: 10.1016/j.eswa.2018.01.056.

[15] Chakraborty, S., Singh, S. K., & Chakraborty, P. (2019). R-theta local neighborhood pattern for unconstrained facial image recognition and retrieval. *Multimedia Tools and Applications*, 78(11), 14799-14822.

[16] K. T. Ahmed, S. Ummesafi, and A. Iqbal, "Content based image retrieval using image features information fusion," *Inf. Fusion*, vol. 51, pp. 76–99, 2019, doi: 10.1016/j.inffus.2018.11.004.

[17] M. Sotoodeh, M. R. Moosavi, and R. Boostani, "A novel adaptive LBP-based descriptor for color image retrieval," *Expert Syst. Appl.*, vol. 127, pp. 342–352, 2019, doi: 10.1016/j.eswa.2019.03.020.

[18] Sampathila, N., & Martis, R. J. (2022). Computational approach for content-based image retrieval of K-similar images from brain MR image database. *Expert Systems*, 39(7), e12652.

[19] Khan, M. A., Sharif, M., Akram, T., Raza, M., Saba, T., & Rehman, A. (2020). Hand-crafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition. *Applied Soft Computing*, 87, 105986.

[20] A. Alzu'bi, A. Amira, and N. Ramzan, "Content-based image retrieval with compact deep convolutional features," *Neurocomputing*, vol. 249, pp. 95–105, 2017, doi: 10.1016/j.neucom.2017.03.072.

[21] S. Pang, M. A. Orgun, and Z. Yu, "A novel biomedical image indexing and retrieval system via deep preference learning," *Comput. Methods Programs Biomed.*, vol. 158, pp. 53–69, 2018, doi: 10.1016/j.cmpb.2018.02.003.

[22] Ma, W., Zhou, T., Qin, J., Xiang, X., Tan, Y., & Cai, Z. (2022). A privacy-preserving content-based image retrieval method based on deep learning in cloud computing. *Expert Systems with Applications*, 117508

[23] Wang, S. H., Govindaraj, V. V., Górriz, J. M., Zhang, X., & Zhang, Y. D. (2021). Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. *Information Fusion*, 67, 208-229.

[24] K. R. Kruthika, Rajeswari, and H. D. Maheshappa, "CBIR system using Capsule Networks and 3D CNN for Alzheimer's disease diagnosis," *Informatics Med. Unlocked*, vol. 14, pp. 59–68, 2019, doi: 10.1016/j.imu.2018.12.001.

[25] L. T. Alemu and M. Pelillo, "Multi-feature fusion for image retrieval using constrained dominant sets," *Image Vis. Comput.*, vol. 94, p. 103862, 2020, doi: 10.1016/j.imavis.2019.103862.

[26] Wang, W., Jiao, P., Liu, H., Ma, X., & Shang, Z. (2022). Two-stage content based image retrieval using sparse representation and feature fusion. *Multimedia Tools and Applications*, 81(12), 16621-16644.

[27] Wang, Z., Li, M., Wang, H., Jiang, H., Yao, Y., Zhang, H., & Xin, J. (2019). Breast cancer detection using extreme learning machine based on feature fusion with CNN deep features. *IEEE Access*, 7, 105146-105158

[28] Bella, M. I. T., & Vasuki, A. (2019). An efficient image retrieval framework using fused information feature. *Computers & Electrical Engineering*, *75*, 46-60.

[29] Jena, P. K., Khuntia, B., Palai, C., Nayak, M., Mishra, T. K., & Mohanty, S. N. (2023). A Novel Approach for Diabetic Retinopathy Screening Using Asymmetric Deep Learning Features. *Big Data and Cognitive Computing*, *7*(1), 25.

[30] ElAlami, M. E. (2014). A new matching strategy for content based image retrieval system. *Applied Soft Computing*, *14*, 407-418.

Knowledge Graph based Representation to Extract Value from Open Government Data

An Application to the Public Procurement Domain

Kawtar YOUNSI DAHBI[1], Dalila CHIADMI[2], Hind LAMHARHAR[3]
Mohammed V University of Rabat
Rabat, Morocco[1, 2, 3]

***Abstract*—Open government data refers to data that is made available by government entities to be freely reused by anyone and for any purpose. The potential benefits of open government data are numerous and include increasing transparency and accountability, enhancing citizens' quality of life, and boosting innovation. However, realizing these benefits is not always straightforward, as the usage of this raw data often faces challenges related to its format, structure, and heterogeneity which hinder its processability and integration. In response to these challenges, we propose an approach to maximize the usage of open government data and achieve its potential benefits. This approach leverages knowledge graphs to extract value from open government data and drive the construction of a knowledge graph from structured, semi-structured, and non-structured formats. It involves the extraction, transformation, semantic enrichment, and integration of heterogeneous open government data sources into an integrated and semantically enhanced knowledge graph. Learning mechanisms and ontologies are used to efficiently construct the knowledge graph. We evaluate the effectiveness of the approach using real-world public procurement data and show that it can detect potential fraud such as favoritism**.

Keywords—Knowledge graph; open government data; knowledge graph construction; public procurement; fraud detection

I. Introduction

Open Government Data (OGD) is a concept that continues to prosper and evolve. It includes all data collected or produced by public administrations that are made available for the public to be freely used [1]. The publication of government data has a significant impact that could be identified through multiple aspects: fostering innovation, improving transparency, public accountability, and collaboration, and improving citizens' quality of life [2] [3]. However, despite these efforts, technical barriers limit the effective use of OGD. For example, users may have difficulty finding relevant data due to the large volume of information and the diversity of portals [4][5]. Additionally, the data may be presented in unstructured formats, requiring manual transcription and significant time to process [2], [6], [1]. Furthermore, a single dataset may not be enough to fulfill user requests, so combining data from multiple sources is often necessary. However, syntactic, structural, and semantic heterogeneity can make it challenging to integrate this data effectively [7]. Therefore, using OGD in its raw form requires technical expertise and significant time and effort to discover, process, integrate, and analyze the data. Given the potential benefits of OGD and the technical barriers to its effective exploitation, exploring new approaches and technologies that can facilitate its use is crucial. One promising avenue for such exploration is the use of knowledge graphs, which have already been successfully implemented by tech giants such as Google and widely used in several contexts [8] – [12]. A knowledge graph is a powerful tool for organizing and analyzing complex information, such as that contained in OGD. It is a semantic graph that captures information about entities and their relationships in an easily machine-processable way [12]. The use of semantics in knowledge graphs allows for a more nuanced and contextually rich understanding of the data, leading to more accurate and insightful analyses. Knowledge graphs offer also a centralized solution for integrating different types of data from heterogeneous sources, providing end-users with a single point of access. Furthermore, knowledge graphs support multiple advanced applications, such as Q&A systems, semantic search, reasoning, and knowledge inference, enabling the development of smart services that extract value from OGD[13] [14].

In this paper, we propose a generic approach that explores the potential of knowledge graphs to transform open government data (OGD) into valuable knowledge. The approach involves constructing a knowledge graph from structured, semi-structured, and unstructured OGD data sources, and offers the following contributions:

- Representing OGD in a semantically rich and machine-processable format to enhance their usefulness.
- Integrating heterogeneous data sources into a centralized solution represented by a knowledge graph, which enables users to have a unified and comprehensive view of the data.

By adopting this approach, we aim to address the challenges related to data processability and integration and to provide users with easily usable data that can be effectively and efficiently utilized. The proposed approach offers a powerful solution for maximizing the value of OGD and encapsulates the technical difficulties associated with processing open government data (OGD). The rest of this paper is organized as follows: Section II presents related works, Section III gives an overview of the proposed approach to construct a knowledge graph from OGD, and Section IV presents the use case related to the public procurement domain, which aims to construct a knowledge graph based on public procurement data and performs advanced analysis to detect anomalies such as favoritism and overpricing. The study concludes in Section V.

II. Related Works

In this section, we present related works that propose the construction of a knowledge graph to extract value from Open Government Data (OGD). We give a brief overview of each work and then draw their main limitations.

The authors in [15] propose the construction of a knowledge graph related to the cadaster in the Netherlands. The knowledge graph is exploited to offer improved data browsing, analysis for urban planning, and the development of location-aware chatbots. However, the approach considers only RDF-like datasets.

Authors in [16] propose the construction of a knowledge graph from Zaragoza's open data to provide a single point for the city's knowledge. Semantic enrichment and transformation are supported through the usage of scripts to transform datasets to RDF. However, the approach doesn't propose dataset integration.

In [17], the authors propose to construct a knowledge graph for the description of public services, the objective of the knowledge graph is to provide users with personalized information about different public services based on their profile and circumstances. The implementation of the knowledge graph was achieved by creating the schema defining entities, attributes, and relationships and populating the graph using GRAQL queries.

The TBFY project [18] [19] proposes the construction of a public procurement knowledge graph to support transparency. For semantic enrichment, the approach uses two ontologies. The transformation to RDF is supported by the RML Mapper tool which automates the generation of RDF triples based on the RML mapping. However, it requires the manual definition of the mapping between the data sources and the ontologies.

Authors in [20] propose the construction of a knowledge graph to support the supervision and analysis of fiscal projects funded by the European Union. The proposed approach collects data from the Open Data API provided by the Greek Ministry of Economy and Finance. Semantic enrichment is based on two ontologies. The approach proposes the usage of the ETL unified views for Data transformation to RDF and semantic enrichment and publication of data in an RDF triple store. Data can be retrieved via SPARQL queries. Performance indicators were defined to assess the state of the project and Density-Based Spatial Clustering of Applications with Noise, (DBSCAN) was used to identify Red Flags.

Authors in [21] propose a solution to support budget transparency. The proposed approach is based on the creation of a knowledge graph from data related to the public budget. Data is published by the budget office in the form of an XML file: an annual file for budget distribution and monthly files for monitoring budget execution. The data is semantically enriched with the National Budget ontology and transformed into RDF through the use of an ETL tool. For data exploitation, the approach proposes the publication of data through a SPARQL access point as well as a set of solutions for data visualization.

The related works presented to provide a background for extracting value from open government data (OGD) through their transformation into knowledge graphs. However, they have limitations that we present below. Firstly, most approaches are either domain-specific or related to specific data sources and cannot be adapted to diverse contexts and domains. Secondly, data transformation to RDF is carried out using scripts, ETL, or mapping languages. This process is tedious, time-consuming, and difficult to implement. It requires the intervention of the user to specify the mapping between data sets and semantic models, which requires a good understanding of the dataset content and structure, domain knowledge, and technical expertise. Thirdly, the majority of approaches focus on dataset transformation without proposing solutions for dataset integration, such as entity linking. When it is proposed, it is not generic and is done for specific types of entities based on linking rules specified by the users. Lastly, most approaches consider only structured data sources. This excludes a large proportion of OGD that is published in unstructured formats.

Therefore, in this work, we aim to address these challenges by proposing a generic and domain-independent approach to constructing a knowledge graph. The proposed approach considers structured, non-structured, and unstructured data sources, and provides an efficient solution to semantically transform OGD with a focus on data integration.

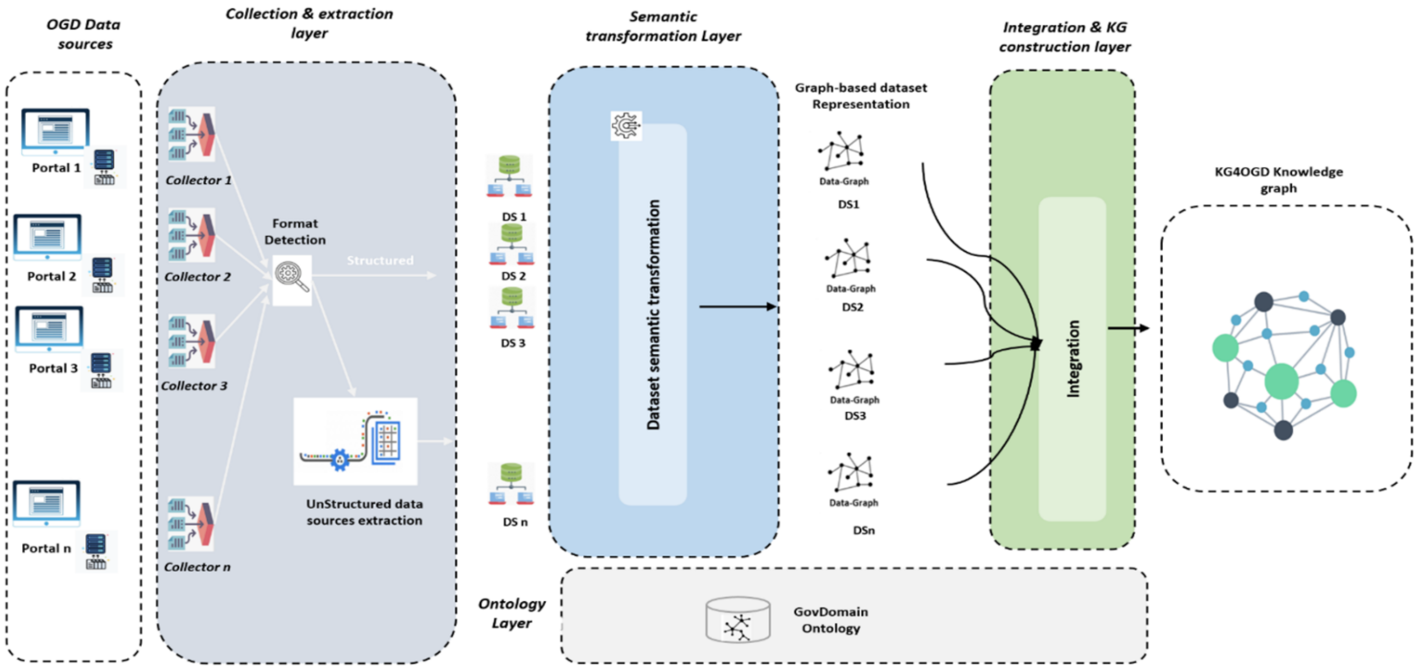


Fig. 1. Overview of the proposed architecture.

III. The Proposed Approach

This section presents a detailed description of the proposed approach for constructing a knowledge graph from Open Government Data sources. The approach is based on a layered architecture that can collect and extract data from structured, semi-structured, and unstructured heterogeneous government data sources. The collected data undergoes semantic enhancement and transformation, after which it is represented as a knowledge graph that offers an interconnected, integrated, and unified representation of data. The constructed knowledge graph can indeed be used through a plethora of applications, such as Q&A, semantic search, knowledge reasoning, and advanced analytics capabilities. These features enable to provide open government data users with useful insights and valuable information.

The proposed architecture depicted in Fig. 1 comprises three main layers: *The Collection and Extraction* layer, the *Semantic Transformation layer*, and the *Integration and Knowledge Graph Construction* layer. These layers are further enhanced by a domain ontology for the representation of domain knowledge. We present below a description of these layers.

A. Collection & Extraction Layer

The Collection and Extraction layer is responsible for discovering and collecting data sources published on various government portals (Pi). Algorithm 1 presents the algorithm that outlines the process performed by this layer.

To discover relevant data sources, the layer employs a set of collectors that use web scraping techniques and can be configured to filter data sources based on keywords, data format, or other specific criteria (CFi) (Line 6 Algorithm 1). This layer is designed to retrieve both structured (datasets) and unstructured data sources, including those published in non-dataset formats such as reports, PDFs, and databases (Line 7 Algorithm 1).

The layer performs format detection (Line 8 Algorithm 1), and structured sources are directly transferred to the Semantic Transformation Layer (Line 10 Algorithm 1).

For unstructured data sources, the layer performs additional processing to extract structured data. For instance, it uses optical character recognition (OCR) and natural language processing (NLP) techniques to identify and extract tables from unstructured data sources, such as PDFs (Line 11 Algorithm 1). Each extracted table is considered a dataset and is transferred to the Semantic Transformation Layer for further processing (Line 14 Algorithm 1).

Algorithm 1: Collect & extract datasets

```
Input: -  $P_i$ government data portal
       -  $CF_i$ Filter criteria
Output: D = {datasets}
S← {}
D← {}
DiscoverDataSource($P_i$ , $CF_i$)
S←RetrieveDataSources()
For each $s_i$ in S
    $fs_i$←DetectFormat($s_i$)
    If $fs_i$ is structured then D.add($fs_i$)
    Else IdentifyTable($s_i$)
        For j← 1 to n
            $t_{ij}$← ExtractTable ($s_i$)
            D.add($t_{ij}$)
        EndFor
    EndIf
EndFor
Return D
```

B. Semantic Transformation Layer

The Semantic Transformation Layer is responsible for adding semantics to the data and transforming it into a machine-processable and understandable format. This layer aims to enhance the usefulness of the collected and extracted datasets by enriching them with semantic information. For this purpose, the layer uses the GovDomain ontology, a domain ontology that formalizes and represents domain knowledge. The Gov domain ontology is intended to model a consensus among governments on the concepts and relationships that exist in published datasets. It is thus an essential tool to solve the problem of semantic heterogeneity.

The semantic transformation layer takes as input the datasets collected and extracted by the Collection & Extraction layer. For each dataset di, it generates an RDF graph-based representation of the dataset GD-di, which is semantically enriched with the GovDomain ontology. The process performed by this layer involves two main steps depicted in Fig. 2: Semantic model construction and RDF graph generation.

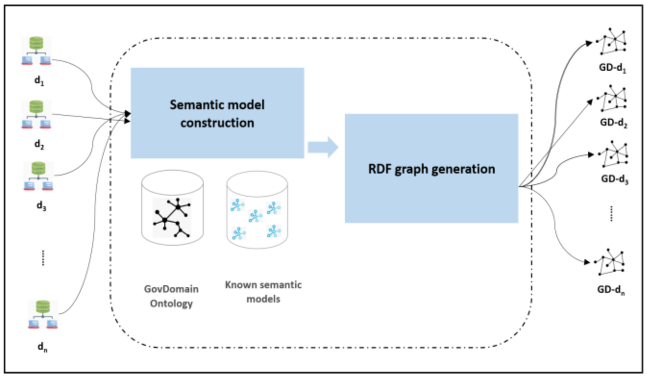


Fig. 2. Semantic transformation process.

The first step consists of constructing a semantic model for the dataset. This model is a requirement for semantic enrichment. It aims to define the structure of the dataset in terms of concepts and relations from the GovDomain ontology. Specifically, it establishes the mapping between the attributes of the dataset and the concepts or properties of the ontology, as well as the relationships between these attributes in terms of ontology relations.

For example, Fig. 3 depicts the semantic model of a public procurement dataset, which includes information on public contracts awarded by public entities, their details (title, reference, description, price, and duration, and the purchasing organization and selected supplier.

Constructing the semantic model for a dataset involves two sub-steps. The first sub-step involves mapping the dataset's attributes to their corresponding semantic types following the domain ontology. A semantic type can be a class URI or a combination of a class and a property from the GovDomain ontology. For instance, in the public procurement dataset, the attributes "Acheteur Public", "Fournisseur "," and "Titre" are mapped to their respective semantic types, which are (PCO: organization, orgName), (PCO: supplier, SupplierName), and (PCO: contract, title)). The second sub-step aims to establish the relations between the attributes in terms of ontology relations. For instance, the relations (PCO: Hasbuyer) and (PCO: HasSupplier) link the class (PCO: Contract) respectively with the classes (PCO: Organization) and (PCO: Supplier).

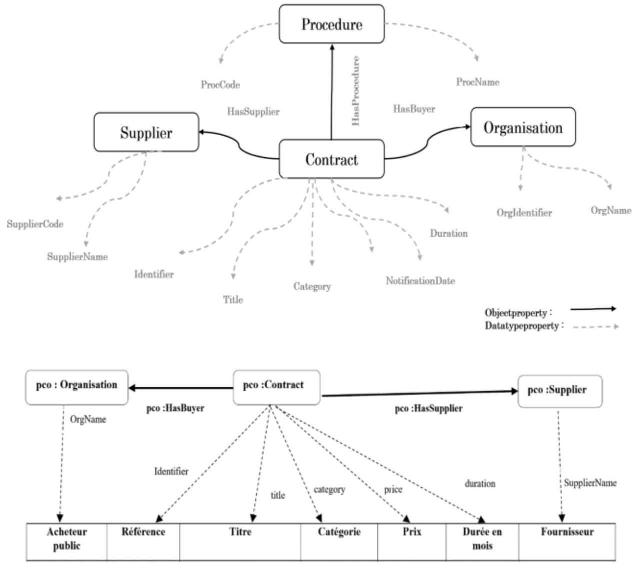


Fig. 3. Example of a semantic model for a dataset related to public procurement.

The manual creation of a semantic model for a dataset is a challenging and time-consuming task, which demands an extensive understanding of the dataset's content and structure, as well as domain-specific knowledge and technical expertise. To address this challenge, we propose an automatic approach to construct the semantic model of a dataset, thereby removing the need for manual intervention in the process. The proposed approach utilizes machine learning to automatically learn the dataset's semantic model by constructing it based on a set of known semantic models that serve as the training data. We employ the algorithm proposed by [22], which offers a method to learn semantic models of structured data sources by mapping them to a domain ontology. The algorithm constructs a weighted graph that represents the space of potential semantic models for a dataset, and provided a ranked list of the top-k best semantic models.

Based on the constructed semantic model the second step (Fig. 3) involves the RDF graph generation. The resulting model is formalized using the RDF Mapping Language (RML). RML [23] allows for the definition and expression of mapping rules between data sources and the RDF model, describing how the existing data can be represented according to the RDF model. The RML mapping document is interpreted by an RML processing engine and used to automatically generate RDF triples from the dataset. The resulting RDF graph provides a graph-based representation of the dataset, thereby enabling machine-readable and machine-understandable access to the dataset's content.

C. Integration and Knowledge Graph Construction Layer

The Integration & Knowledge Graph Construction Layer is responsible for the integration of data into a knowledge graph. This layer takes as input the RDF graphs that result from the semantic transformation of datasets. The integration process involves two steps, which are outlined in Algorithm 2

Algorithm 2: KG construction & integration

```
Input: -GD-$d_1$, GD-$d_2$…. GD-$d_n$: RDF graphs related to datasets { $d_1$, $d_2$…. $d_n$ }
        -C: List of ontology classes described in datasets { $d_1$, $d_2$….$d_n$ }
Output: KG4OGD knowledge graph.
KG4OGD ← ∅
For each GD-$d_i$
    KG4OGD ← KG4OGD.Add(GD-$d_i$)
EndFor
For each cp from C
    Bp ← ConstructBloc(cp)
        For each ($e_i$,$e_j$) ∈ Bp
            If Comp ($v_i$,$v_j$)= true then
            t←ConstructIdentityTriple($e_i$,$e_j$)
            KG4OGD ← KG4OGD.Add(t)
            EndIf
        Endfor
EndFor
Return KG4OGD
```

The first step involves publishing and consolidating all datasets into the knowledge graph (Line 8 Algorithm 2). The second step is crucial for completing data integration and involves entity alignment. Entity alignment refers to the process of identifying and mapping coreferent entities, which are entities that refer to the same real-world entity but with different URIs [24]. This step involves a pairwise comparison of all entities that have been published in the graph. To improve efficiency, we use a blocking technique [25] that reduces the number of comparisons and identifies potential candidates for entity alignment. We apply a class-based partitioning approach to construct blocks, which groups potential co-referenced entities into blocks based on their class (Line 11 Algorithm 2). This method reduces the quadratic complexity of the process by limiting the number of comparisons required. After constructing the blocks, we compare the entities belonging to the same block by examining their discriminative properties, which are unique properties that identify entities of a particular class (Line 13, Algorithm 2). These discriminative properties are described in the domain ontology by adding owl:hasKey axioms. If two entities have equal values for these properties, we consider the entities to be co-referent, and we add identity links between them to the knowledge graph using the owl:sameAs property (Line 14 &15, Algorithm 2).

D. Advantages of the Proposed Approach

The proposed approach makes significant contributions to the field of Open Government Data (OGD). Firstly, it enables the representation of OGD in a structured and semantically rich format that can be easily processed and interpreted by machines. This overcomes the limitations of existing data formats and promotes greater interoperability and reuse of OGD. Secondly, the approach provides an integrated representation of OGD that allows users to view all published data in a unified and complete manner. By utilizing knowledge graphs to integrate data from multiple OGD sources, the proposed approach facilitates data discovery and analysis and enables the identification of previously unseen patterns and relationships across diverse datasets.

When compared to related works, the proposed approach offers significant contributions. Firstly, it addresses the challenge of integrating diverse data sources with structured, semi-structured, or unstructured formats. The approach automates the process of semantically transforming datasets, which can be time-consuming and error-prone when done manually. This is achieved by utilizing learning mechanisms and mapping languages. Furthermore, the proposed approach is domain-independent and presents a generic solution to integrate heterogeneous datasets. It accomplishes this through a schema alignment and entity alignment approach, which is designed to handle all types of entities without the need for human intervention to specify linking rules.

IV. Use Case Study

In this section, we present a use case that implements the proposed architecture and demonstrates its operationalization. The use case focuses on the public procurement domain, which is a critical government sector for ensuring transparency and accountability in government spending.

A. Context

Public procurement is a significant part of government spending, but it is also a domain prone to corruption and fraud [26] [27] [29]. Public Procurement (PP) is a distinct area within Open Government Data (OGD) that plays a critical role in ensuring transparency and accountability in government spending. By releasing public procurement data as open and accessible, governments and civil society organizations can promote fair competition, identify patterns of corruption and fraud, and hold public officials accountable.

The use case aims to implement the proposed architecture in the context of public procurement in France. By integrating public procurement data from several open government data sources into a knowledge graph, we can apply anomaly detection algorithms to identify instances of fraud such as favoritism and overpricing. Favoritism in public procurement [28] refers to the practice of giving preferential treatment to a particular supplier, contractor, or bidder. Overpricing [29], on the other hand, refers to the practice of charging prices that are higher than what would be considered reasonable or fair.

B. Data Sources

In the French context, there exists a multitude of public procurement data sources, but for our specific purposes, we have opted to primarily utilize three sources: the BOAMP, Essential Data on Public Procurement (DECP), and the SIRET database (Cf. Table I).

[1]BOAMP is a French government platform that publishes public procurement notices and announcements, as well as detailed information about awarded contracts. The platform also provides an API that allows users to retrieve data in both JSON and XML formats

[1] https://www.boamp.fr/

Essential Data on Public Procurement (DECP) is a database that features structured and standardized information regarding all public procurement procedures. This expansive dataset includes details such as the contracting authority's name and location, the type and value of the contract, and the selected supplier. These data are consolidated in the French open government data portal[2].

The SIRENE[3] the database is a comprehensive registry of all business entities operating in France, providing information regarding each entity's legal status, activity sector, and location. By leveraging SIRENE data, we can identify the entities involved in public procurement procedures and establish links with other pertinent datasets.

TABLE I. PUBLIC PROCUREMENT DATA SOURCES

Data sources	Publisher	Format	Frequency of update
BOAMP	Direction of Legal and Administrative Information (DILA).	XML,JSON	Daily
Essential Data on Public Procurement (DECP)	The Ministry of the Economy, Finance, and Industrial and Digital Sovereignty	XML , JSON	Monthly
SIRENE Database	French National Institute of Statistics and Economic Studies (INSEE)	CSV	Quartly

By integrating and linking these various data sources in a knowledge graph, we can create a comprehensive and well-organized representation of public procurement data. To achieve this, we need to implement a domain ontology that captures and formalizes knowledge relevant to the field of public procurement

C. Ontology Development

Developing a domain ontology is a requirement for constructing the knowledge graph as it plays a role in the semantic enrichment of government data sources. To this end, we constructed a domain ontology for the public procurement domain by reusing the PCO ontology [30].

The PCO ontology, known as the Public Contracts Ontology, is a domain-specific ontology that models the essential concepts and relationships within the public procurement domain. Its primary objective is to support the integration and analysis of public procurement data from various sources. The ontology encompasses a broad range of concepts and relationships that apply to public procurement, such as procurement notices, contracts, suppliers, products and services, and tendering procedures. Additionally, it includes concepts linked to the legal and regulatory frameworks that oversee public procurement, such as procurement rules and regulations, procurement authorities, and procurement methods.

For the implementation, we have customized the PCO ontology to match the French context. This involved correlating the concepts and relationships in the ontology with the appropriate terms and structures in the French public procurement domain. This adaptation (Fig. 4) ensures that the constructed knowledge graph represents the relevant aspects of the French public procurement domain accurately and is tailored to our use case's particular needs.

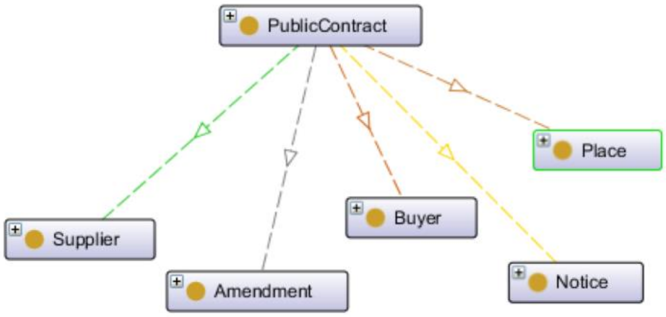


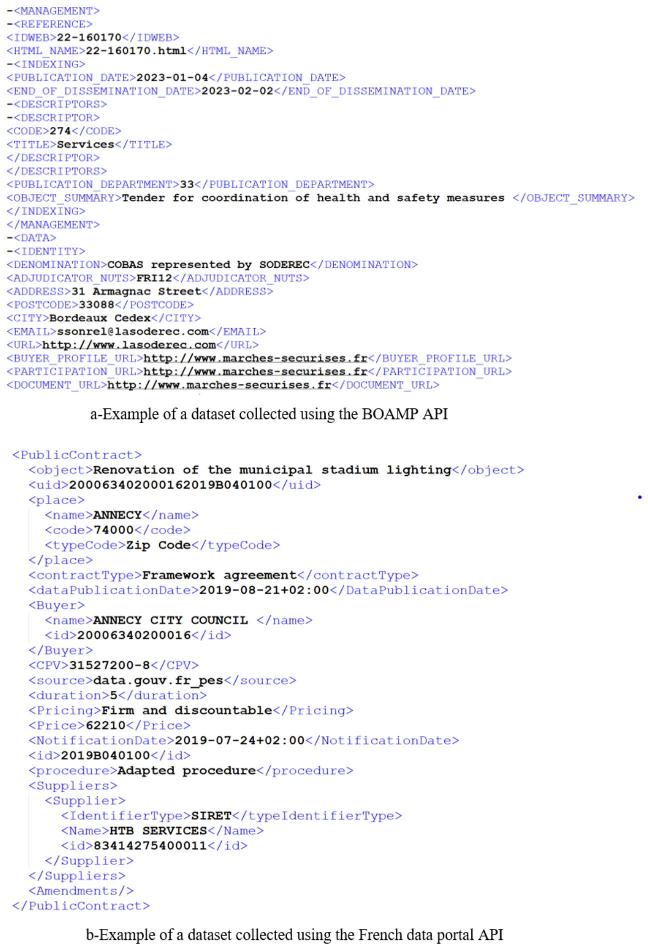
Fig. 4. Public procurement domain ontology implemented in protégé tool

D. Knowledge Graph Construction

The initial step in constructing the knowledge graph involved collecting data from identified data sources. We automated the data collection process by developing data collection scripts using Python and leveraging APIs. We utilized various APIs, such as the BOAMP API, and the API offered by the French government data portal, to collect structured data in different formats, including XML, JSON, and CSV. Fig. 5 shows an example of the collected data.

[2] https://www.data.gouv.fr/fr/datasets/donnees-essentielles-de-la-commande-publique-fichiers-consolides/

[3] https://www.data.gouv.fr/fr/datasets/base-sirene-des-entreprises-et-de-leurs-etablissements-siren-siret/



```
-<MANAGEMENT>
-<REFERENCE>
<IDWEB>22-160170</IDWEB>
<HTML_NAME>22-160170.html</HTML_NAME>
-<INDEXING>
<PUBLICATION_DATE>2023-01-04</PUBLICATION_DATE>
<END_OF_DISSEMINATION_DATE>2023-02-02</END_OF_DISSEMINATION_DATE>
-<DESCRIPTORS>
-<DESCRIPTOR>
<CODE>274</CODE>
<TITLE>Services</TITLE>
</DESCRIPTOR>
</DESCRIPTORS>
<PUBLICATION_DEPARTMENT>33</PUBLICATION_DEPARTMENT>
<OBJECT_SUMMARY>Tender for coordination of health and safety measures </OBJECT_SUMMARY>
</INDEXING>
</MANAGEMENT>
-<DATA>
-<IDENTITY>
<DENOMINATION>COBAS represented by SODEREC</DENOMINATION>
<ADJUDICATOR_NUTS>FRI12</ADJUDICATOR_NUTS>
<ADDRESS>31 Armagnac Street</ADDRESS>
<POSTCODE>33088</POSTCODE>
<CITY>Bordeaux Cedex</CITY>
<EMAIL>ssonrel@lasoderec.com</EMAIL>
<URL>http://www.lasoderec.com</URL>
<BUYER_PROFILE_URL>http://www.marches-securises.fr</BUYER_PROFILE_URL>
<PARTICIPATION_URL>http://www.marches-securises.fr</PARTICIPATION_URL>
<DOCUMENT_URL>http://www.marches-securises.fr</DOCUMENT_URL>
```

a-Example of a dataset collected using the BOAMP API

```
<PublicContract>
  <object>Renovation of the municipal stadium lighting</object>
  <uid>200063402000162019B040100</uid>
  <place>
    <name>ANNECY</name>
    <code>74000</code>
    <typeCode>Zip Code</typeCode>
  </place>
  <contractType>Framework agreement</contractType>
  <dataPublicationDate>2019-08-21+02:00</DataPublicationDate>
  <Buyer>
    <name>ANNECY CITY COUNCIL </name>
    <id>20006340200016</id>
  </Buyer>
  <CPV>31527200-8</CPV>
  <source>data.gouv.fr_pes</source>
  <duration>5</duration>
  <Pricing>Firm and discountable</Pricing>
  <Price>62210</Price>
  <NotificationDate>2019-07-24+02:00</NotificationDate>
  <id>2019B040100</id>
  <procedure>Adapted procedure</procedure>
  <Suppliers>
    <Supplier>
      <IdentifierType>SIRET</typeIdentifierType>
      <Name>HTB SERVICES</Name>
      <id>83414275400011</id>
    </Supplier>
  </Suppliers>
  <Amendments/>
</PublicContract>
```

b-Example of a dataset collected using the French data portal API

Fig. 5. Example of the collected data.

As the collected data was already structured, no further data extraction was required. Our data collection approach allowed us to retrieve data efficiently and accurately while reducing the resources needed for manual data collection. This enabled us to collect a significant amount of data related to public contracts from 2019 to 2022.

After collecting the data, we utilized Karma, an open-source tool, for semantic transformation. Karma[4] is a semi-automatic tool that implements the approach proposed by [22] for dataset transformation, enabling the definition of mappings from datasets to ontologies, building of semantic models, and publishing of data as RDF. With Karma's machine learning capability, it can learn to map datasets to an ontology by using the attribute values of mapped dataset attributes. When users define relationships between classes, Karma learns from them and can suggest properties and classes to model new sources automatically. Karma's validation feature allows users to verify the proposed model's quality, and it also supports the generation of RML mapping, which automates the RDF graph-based representation of the datasets. Incorporating the PCO ontology in the semantic transformation process enriched the RDF graph output for each collected dataset, representing relationships between data elements, entities, and concepts for constructing the knowledge graph. As a first step in the integration process, all datasets were aligned to the same schema, which was the PCO ontology.

The resulting RDF graphs were combined to form the knowledge graph. We further employed entity linking as a means of completing the integration process. We used the OWL: has key and OWL: Sameas axioms to identify unique discriminative properties for entities in the domain ontology and to link co-referent entities, respectively. For example, we used the SIRET identifier to link unique enterprises as suppliers and the SIREN identifier to link public organizations as public buyers.

Through the use of entity linking, we were able to more effectively integrate data from multiple sources into the knowledge graph, providing a unified and integrated view of data.

The data was stored in an RDF triple store using the Apache Jena framework[5] and made available through Apache Jena Fuseki. This allowed users to query the knowledge graph using the SPARQL query language and retrieve relevant information from the integrated datasets. In total, the knowledge graph consolidated data related to 96,845 public contracts.

E. Knowledge Graph Consumption

To demonstrate the value of our approach, we conducted an analysis using the isolation forest algorithm on integrated public procurement data in our knowledge graph. The isolation forest algorithm is a powerful machine learning tool well-suited to detecting anomalies in datasets. It works by creating a random forest of decision trees and isolating data points that are not consistent with the majority of the data. In the case of public procurement data, the isolation forest algorithm can be used to identify anomalies that may indicate corruption or fraud.

In our study, we used a SPARQL query to extract data from the knowledge graph and applied the isolation forest algorithm to our integrated public procurement data. We found a total of 56 anomalies in the dataset, which were categorized as either favoritism or overpricing. The algorithm identified 27 cases of favoritism and 29 cases of overpricing.

We qualitatively assessed the algorithm results and found that the contracts identified as anomalies were consistently awarded to the same suppliers, indicating a potential case of favoritism. Additionally, we observed that the contract prices for these anomalies were significantly higher than the market price or previous prices for similar contracts, indicating a potential case of overpricing.

The results of this case study demonstrate the value of the proposed approach, which proposes to consolidate data from multiple sources into a single knowledge graph and represent it in a format that promotes analysis and processing for value extraction. By utilizing a knowledge graph and the isolation forest algorithm, we were able to identify potential anomalies that may have gone undetected using a traditional approach.

[4] https://usc-isi-i2.github.io/karma/

[5] https://jena.apache.org/

This can help auditors better understand procurement data, detect potential fraud or corruption, and ultimately promote a more transparent and accountable government.

V. CONCLUSION

In this paper, we introduce an approach for extracting value from open government data and transforming it into valuable knowledge. Our approach utilizes a layered architecture to extract and collect structured and unstructured data sources, transform them into a machine-processable, understandable, and semantically rich format, and integrate them into a knowledge graph. The proposed approach offers an efficient and generic way to construct the graph, which can be further utilized for a plethora of applications. The approach was implemented in the context of public procurement and successfully identified anomalies such as overpricing and favoritism. Although the paper is focused on public procurement, the proposed approach applies to other governmental domains. As a future direction, we aim to propose a detailed description of how to exploit the knowledge graph, to further improve the effectiveness of the proposed approach.

REFERENCES

[1] Attard, J., Orlandi, F., Scerri, S., & Auer, S. (2015). A systematic review of open government data initiatives. Government information quarterly, 32(4), 399-418.

[2] Ubaldi, B. (2013). Open government data: Towards empirical analysis of open government data initiatives.

[3] Janssen, M., Charalabidis, Y., & Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. Information systems management, 29(4), 258-268

[4] K. Y. Dahbi, H. Lamharhar, D. Chiadmi Exploring dimensions influencing the usage of Open Government Data portals. In Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications (pp. 1-6).

[5] S. Neumaier, J. Umbrich, A. Polleres (2016). Automated quality assessment of metadata across open data portals. Journal of Data and Information Quality (JDIQ), 8(1), 1-29.

[6] K. Y. Dahbi, H. Lamharhar, D. Chiadmi .Toward a user-centered approach to enhance Data discoverability on Open Government Data portals. In 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS) (pp. 1-5). IEEE.

[7] M. Mountantonakis, Y. Tzitzikas, (2019). Large-scale semantic integration of linked data: A survey. ACM Computing Surveys (CSUR), 52(5), 1-40. 8. H.Purohit , R. Kanagasabai, N. Deshpande (2019, January).

[8] Towards Next Generation Knowledge Graphs for Disaster Management. In 2019 IEEE 13th International Conference on Semantic Computing (ICSC) (pp. 474-477). IEEE.

[9] Y. Jia, Y.Qi , H.Shang , R. Jiang, A. Li, (2018). A practical approach to constructing a knowledge graph for cybersecurity. Engineering, 4(1), 53-60. 10. M. Wang, Q. Zeng, W. Chen , J. Pan, H. Wu, C. Sudlow, D. Robertson, (2020).

[10] Building the Knowledge Graph for UK Health Data Science.

[11] J. M. Gomez-Perez, J. Z .Pan, G. Vetere, , H. Wu, (2017). Enterprise knowledge graph: An introduction. In Exploiting linked data and knowledge graphs in large organisations (pp. 1-14). Springer, Cham.

[12] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., ... & Zimmermann, A. (2021). Knowledge graphs. ACM Computing Surveys (CSUR), 54(4), 1-37.

[13] A. Hogan, E . Blomqvist, M .Cochez, C . d'Amato, G. de Melo, C. Gutierrez, , R. Navigli(2020). Knowledge graphs. arXiv preprint arXiv:2003.02320. 14. J. Yan, C. Wang, , W. Cheng, M. Gao, A. Zhou, (2018). A retrospective of knowledge graphs. Frontiers of Computer Science, 12(1), 55-74. https://doi.org/10.1007/s11704-016-5228-9

[14] J. Yan, C. Wang, , W. Cheng, M. Gao, A. Zhou, (2018). A retrospective of knowledge graphs. Frontiers of Computer Science, 12(1), 55-74. https://doi.org/10.1007/s11704-016-5228-9

[15] S. Ronzhin, E. Folmer, P. Maria, M. Brattinga, W. Beek, R. Lemmens, R., R. van't Veer(2019). Kadaster knowledge graph: Beyond the fifth star of open data. Information, 10(10), 310.

[16] P. Espinoza-Arias, M. J. Fernández-Ruiz, V.Morlán-Plo, R.Notivol-Bezares, O. Corcho (2020). The Zaragoza's Knowledge Graph: Open Data to Harness the City Knowledge. Information, 11(3), 129.

[17] Rafail, P., & Efthimios, T. (2020, November). Knowledge Graphs for Public Service Description: The Case of Getting a Passport in Greece. In European, Mediterranean, and Middle Eastern Conference on Information Systems (pp. 270-286). Springer, Cham.

[18] Soylu, A., Corcho, O., Elvesæter, B., Badenes-Olmedo, C., Blount, T., Yedro Martínez, F., ... & Roman, D. (2022). TheyBuyForYou platform and knowledge graph: Expanding horizons in public procurement with open linked data. Semantic Web, (Preprint), 1-27.

[19] Soylu, A., Elvesæter, B., Turk, P., Roman, D., Corcho, O., Simperl, E., ... & Lech, T. C. (2019). An overview of the TBFY knowledge graph for public procurement. CEUR Workshop Proceedings

[20] Bratsas, C., Chondrokostas, E., Koupidis, K., & Antoniou, I. (2021). The use of national strategic reference framework data in knowledge graphs and data mining to identify red flags. Data, 6(1), 2.

[21] Cifuentes-Silva, F., Fernández-Álvarez, D., & Labra-Gayo, J. E. (2020). National budget as linked open data: New tools for supporting the sustainability of public finances. Sustainability, 12(11), 4551.

[22] Taheriyan, M., Knoblock, C. A., Szekely, P., & Ambite, J. L. (2016). Learning the semantics of structured data sources. Journal of Web Semantics, 37, 152-169.

[23] Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., & Van de Walle, R. (2014, January). RML: a generic language for integrated RDF mappings of heterogeneous data. In Ldow.

[24] Oliveira, I. L., Fileto, R., Speck, R., Garcia, L. P., Moussallem, D., & Lehmann, J. (2021). Towards holistic entity linking: Survey and directions. Information Systems, 95, 101624.

[25] Papadakis, G., Skoutas, D., Thanos, E., & Palpanas, T. (2019). A survey of blocking and filtering techniques for entity resolution. arXiv preprint arXiv:1905.06167.

[26] Kundu, O., James, A. D., & Rigby, J. (2020). Public procurement and innovation: a systematic literature review. Science and Public Policy, 47(4), 490-502.

[27] Rustiarini, N. W., Nurkholis, N., & Andayani, W. (2019). Why people commit public procurement fraud? The fraud diamond view. Journal of public procurement, 19(4), 345-362.

[28] Baranek, B., & Titl, V. (2020). The cost of favoritism in public procurement. FEB Research Report Department of Economics.

[29] Modrušan, N., Rabuzin, K., & Mršic, L. (2021). Review of Public Procurement Fraud Detection Techniques Powered by Emerging Technologies. International Journal of Advanced Computer Science and Applications, 12(2).

[30] Nečaský, M., Klímek, J., Mynarz, J., Knap, T., Svátek, V., & Stárka, J. (2014). Linked data support for filing public contracts. Computers in Industry, 65(5), 862-877.

Deep Learning CNN Model-based Anomaly Detection in 3D Brain MRI Images using Feature Distribution Similarity

Amarendra Reddy Panyala[1], M. Baskar[2*]
Research Scholar, Department of Computer Science and Engineering-School of Computing-College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu,
Tamilnadu, 603 203, India[1]
Assiatant Professor, Department of Information Technology, MLR Institute of Technology, Hyderabad,
Telangana, 500043, India[1]
Associate Professor, Department of Computing Technologies-School of Computing-College of Engineering and Technology,
SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamilnadu, 603 203, India[2]

***Abstract*—Towards detecting an anomaly in brain images, different approaches are discussed in the literature. Features like white mass values and shape features have identified the presence of brain tumors. Various deep learning models like the neural network has been adapted to the problem tumor detection and suffers to meet maximum accuracy in detecting brain tumor. An Adaptive Feature Centric Distribution Similarity Based Anomaly Detection Model with Convolution Neural Network (AFCD-CNN) is sketched towards disease prediction problem to handle the problem. The model considers black-and-white mass features with the distribution of features. First, the method applies the Multi-Hop Neighbor Analysis (MHNA) algorithm in normalizing the brain image. Further, the process uses the Adaptive Mass Determined Segmentation (AMDS) algorithm, which groups the pixels of MRI according to the white and black mass values. The method extracts the ROI with the segmented image and convolves the features with CNN at the training phase. The CNN is designed to convolve the features into one dimension. The output layer neurons are designed to estimate different Feature Distribution Similarity (FDS) values against various features to compute the Anomaly Class Weight (ACW). According to the ACW value, anomaly detection is performed with higher accuracy up to 97% where the time complexity is reduced up to 32 seconds.**

***Keywords*—Deep learning; brian tumor; disease prediction; anomaly detection; CNN; FDS; ACW**

I. Introduction

The entry of modern diseases challenges human society. There are several diseases identified every year which have a significant impact on human life. Some of the diseases produce temporary illnesses, and some of them introduce permanent damage to the human. More than that, some diseases are claiming the lives of humans. The brain tumor is among them, producing permanent illness and is identified as a more challenging one. Whatever the disease, diagnosing the disease at the earliest would support the person in increasing survival. Analyzing the disease at the earliest would help the medical practitioner effectively treat the person.

Brain tumor analysis has been identified as a complicated task in the medical sector where such a disease would claim a person's life. Various detection models are available, and MRI images become the source of finding the disease, which must be passed through several stages to extract the features. For example, the classification system would use texture, mass values, shape, and binary features in classifying the brain image. Also, several methods include Support Vector Machine (SVM), Decision Tree, Ensemble learning, Genetic algorithms, and Neural networks. The methods differ regarding features being considered and the similar way being used.

Deep learning is the modern development of machine learning models which helps automated decisive support systems in handling huge volume of data towards solid support. Convolution neural network has been identified as more effective than the other deep learning models. The CNN has been designed with layers like the convolution layer, which involves convolving the ROI features into a single dimension. Such dimensionality reduction and feature retention support handling massive volumes of MRI images to support the process of image classification. The brain image classification is done according to the texture features and other features obtained from the ROI. The accuracy can be significantly improved by having a massive volume of samples.

An Adaptive Feature Centric Distribution Similarity Based Anomaly Detection Model is detailed to improve the performance, focusing on including multiple brain image features and their distribution metrics in finding the image class. A Multi-Hop Neighbor Analysis (MHNA) algorithm is discussed to normalize the brain image. Also, Adaptive Mass Determined Segmentation (AMDS) algorithm is adapted toward segmenting the regions of the brain image. Similarly, a convolution neural network with two convolution and pooling layers is designed.

The article is structured to present the introduction of brain tumor detection and anomaly detection in detail with disease prediction at Section I. Section II, discusses the complete related work and methods available in literature in detail. Section III discusses the complete working of the proposed anomaly detection and disease prediction system. Section IV details the experimental results and presents a detailed discussion. Finally, conclusion about the entire work is presented in Section V while future area of research is presented in Section VI.

II. RELATED WORKS

The problem of brain image classification has been discussed in various articles, and a subset of methods is discussed in this section.

A CNN-based model is presented in [1], which preprocess the brain images and extracts the features to train CNN. The method uses a T1 weighted image, enhancing the contrast of MRI images to support the classification. Further, a Deep CNN model is sketched in [2] to predict the status of gliomas. The method generates various mutations of the features and based on that, the classification is performed. A modified version of DCNN is discussed in [3], which focuses on adjusting the feature weights at various layers by removing the fully connected layers. A metastases segmentation model with CNN is presented in [4] to support brain image classification. The model segments the images to obtain the metastases and trains the CNN for efficient classification.

In [5], a transfer learning model named GoogleNet is presented for classifying various classes of tumors. The model extracts the features using a pre-trained CNN model and performs classification using algorithms like SVM, KNN, and softmax. A detailed analysis of various CNN models is sketched in [6], which considers the models like S-CNN (CNN trained from scratch). The method uses two brain image data sets to analyze the performance of various approaches.

An ADAM optimizer model is sketched in [7] for brain image classification, which uses different pre-trained models like Xception, NasNet Large, DenseNet121, and InceptionResNetV2 to extract the features. The features extracted are used to train the CNN model with an ADAM optimizer to perform classification. A cumulative variance-based feature selection approach is presented in [8] to classify various grades of malignant brain tumors. The method extracts the features, selects optimal features using CVM (Cumulative Variance Method), and classifies using KNN, NN, and multiclass SVM.

To support automatic diagnosing and help the medical practitioner, an integrated model is presented in [9], which extracts the features using CNN from MRI images. Extracted features are classified using LSTM (Long Short Term Memory) model. A novel classification model is presented in [10], which uses multi-view DNN to perform segmentation, and segmented features are fused with the dynamic fusion method. Segmented results are used to analyze the performance of segmentation.

A transfer learning-based deep learning model is presented in [11], which classifies the tumor according to the features extracted and compares it with the performance of others. An efficient brain tumor segmentation model is presented in [12], which uses the correlation among the features. The correlation model transformed the features and was used to perform classification. A deep convolution neural network-based model is presented in [12], which fine-tunes the layers to perform classification with higher accuracy. A Gaussian convolution neural network-based model is sketched in [13] for brain image classification. The model is designed to classify pituitary, glioma, and meningioma tumors.

A time-distributed CNN LSTM (TD-CNN-LSTM) scheme is discussed in [14], which extracts the features from time-dependent images using CNN. The classification is performed using LSTM. A DCNN model with Feature SVMnamed ((DCNN-F-SVM) is sketched in [15], which trains features and fuses the network to perform classification with SVM. A dilated 3D CNN model is illustrated in [16], which uses feature maps as the key to classification. A U-Net-based CNN model is discussed in [17], which uses U-Net for segmentation and performs classification with the 3D-CNN model. In [18], the author presents a detailed review and identifies the several challenges in classifying brain images into various classes. A forgery detection model is sketched in [19], which uses a propagation-based scheme in organizing the fingerprint image. A PCA-based classification model is sketched in [20], which classifies the finger quickly. A face recognition model is sketched in [21], which considers contour features in classification.

The methods analyzed in this section are subject to producing poor accuracy in classifying the brain images against different classes considered.

III. ADAPTIVE FEATURE-CENTRIC DISTRIBUTION SIMILARITY-BASED ANOMALY DETECTION WITH CNN MODEL (AFCD-CNN)

The proposed model (see Fig. 1) reads the brain image MRI data set and considers black-and-white mass features with the distribution of elements. The Multi-Hop Neighbor Analysis (MHNA) algorithm is initially applied to normalize the brain image and remove the noise from the image. Further, the method uses Adaptive Mass Determined Segmentation (AMDS) algorithm to group similar pixels of brain images to support feature extraction. Once the segmentation is done, the process extracts the ROI and the features passed through CNN to train the network. The CNN design convolves the elements into one dimension. The output layer neurons are subject to measuring Feature Distribution Similarity (FDS) values against various features to compute the Anomaly Class Weight (ACW). According to the ACW value, the classification is done.

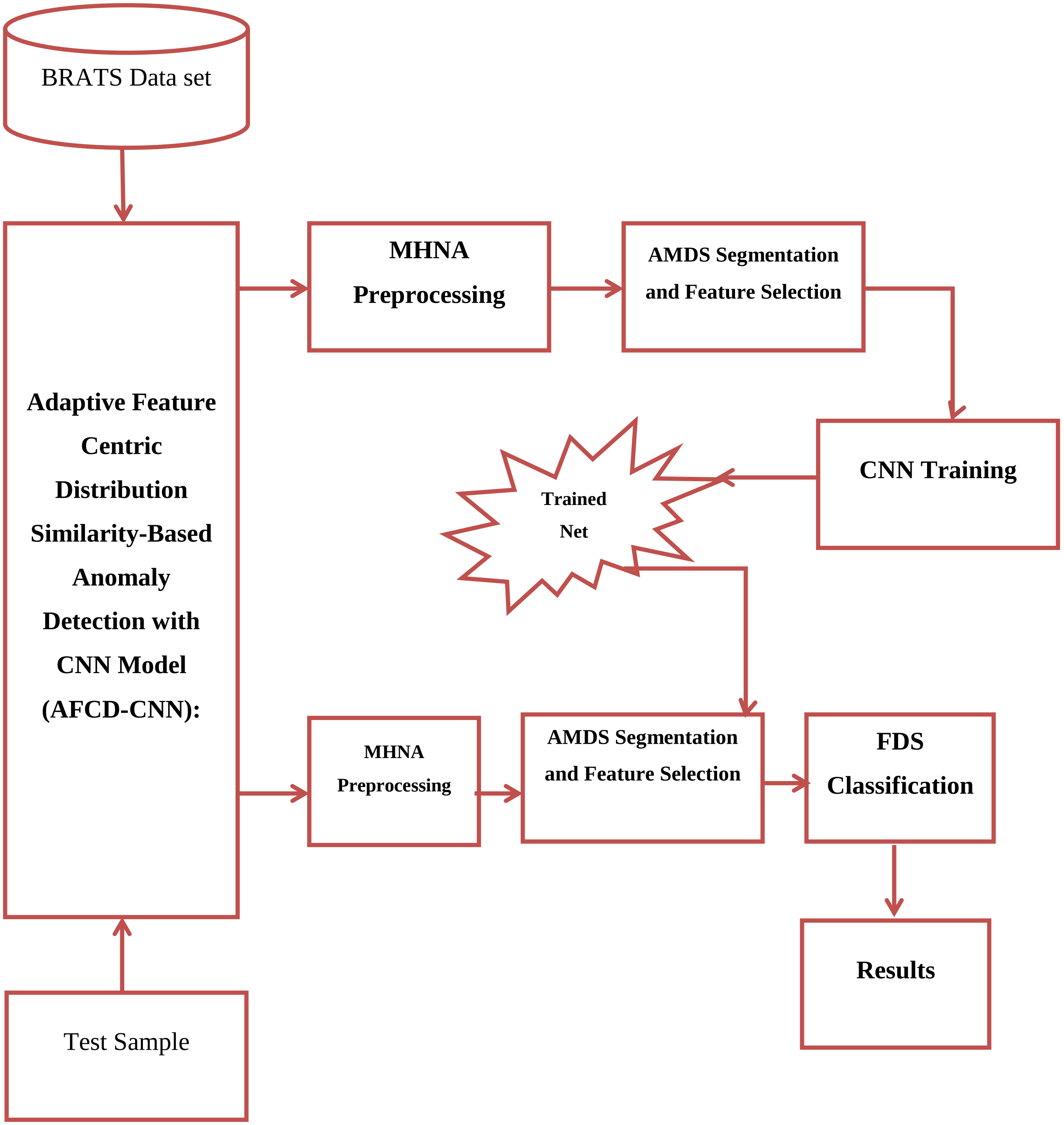

BRATS Data set
Adaptive Feature Centric Distribution Similarity-Based Anomaly Detection with CNN Model (AFCD-CNN):
MHNA Preprocessing
AMDS Segmentation and Feature Selection
CNN Training
Trained Net
MHNA Preprocessing
AMDS Segmentation and Feature Selection
FDS Classification
Results
Test Sample

Fig. 1. Architecture of proposed AFCD-CNN anomaly detection model.

The proposed brain tumor detection model reads the brat's image data set and applies MHNA preprocessing and AMDS segmentation to extract the features. Extracted are trained to a convolution neural network with two convolution layers and pooling layers to compute weight measures for anomaly detection.

A. *MHNA Preprocessing*

The brain image considered has been read towards improving the quality of the image. To perform normalization, the method initializes a window size x, which covers K hops for any pixel. The window is further divided into four regions, and the standard deviation value is measured at each region. Second, a region with the least standard deviation of pixels is identified and selected. Selected region pixels are used in estimating the mean standard value (MSV), and based on that, and the method adjusts the pixel value to normalize the pixel. This will be iterated for the k number of iterations, which denotes the hops. Such normalized image has been used to perform segmentation.

TABLE I. SAMPLE WINDOW

112	135	132	124	75
128	148	72	32	87
136	147	**58**	67	77
121	139	42	51	69
119	122	136	145	55

TABLE II. WINDOW WITH ONE NEIGHBOR

148	72	32
147	**58**	67
139	42	51

TABLE III. WINDOW WITH 2 NEIGHBORS

112	135	132	124	75
128	148	72	32	87
136	147	**58**	67	77
121	139	42	51	69
119	122	136	145	55

The example window considered for the problem is given in Table I and a sub section from the Table I with single neighbor is shown in Table II and a window with two neighbors is shown in Table III. According to Table I considered, the proposed MHNA preprocessing algorithm computes the standard deviation value with four different region pixels as follows:

Standard deviation {112,135,132,128,148,72,136,147,58 } among the region one values are computed. Similarly, the same has been measured for other region pixel values according to the center pixel marked. Now according to the standard deviation value, a single region with the least value is selected. For the selected region, the mean standard value is measured, and based on that, the method computes the new value for the concerned pixel.

```
Algorithm:

Given: Brain Image Bmg
Obtain: Normalized Image Nmg.
Start
    Read Bmg.
    Initialize window size w.
    Initialize neighbor k.
    For each pixel
        W= Construct K hop window.
        Crop image feature as Cm = Crop (Bmg,w,k)
        For each region R
            Compute standard deviation Rstd $= \overset{size(Rs)}{\underset{i=1}{}} Std(R)$
        End
        Region R = $\overset{size(Rs)}{\underset{i=1}{Max}}(Rs(i).Rstd)$
        Compute Mean standard Ms = $\frac{\sum_{i=1}^{Size(R)} Dist(p,R(i))}{size(R-1)}$
        Nmg(p) = p+($\frac{3}{8} \times Ms$)
    End
Stop
```

The MHNA preprocessing algorithm computes any pixel's mean standard value according to the selected region. Based on the value measured, the method normalizes the pixels to perform normalization. The normalized image has been used to perform segmentation and classification.

B. Adaptive Mass Determined Segmentation

The brain image obtained from preprocessing has been used for segmentation. The adaptive mass-determined segmentation algorithm uses two different mass values in grouping the tumor's pixels and other brain cells. To perform this, the method traverses through the entire image and finds the set of black mass values and white mass values. The method computes the boundary between the sets using these white and black mass values. Based on the identified boundary, the method groups the pixels under two classes to produce the segmented image.

```
Algorithm:

Given: Brain Image Pimg
Obtained: Segmented Image Simg
Start
    Read Pimg.
    Initialize black mass set Bms, white mass set Wms.
    For each pixel p
        If p.value<100 then
            If p.value ! ∈ $Bms,$ then
                Bms = Bms∪ $p.value$
            End
        Else
            If p.value ! ∈ $wms$ then
                wms = wms∪ $p.value$
            End
        End
    End
    Identify boundary set Bs = $\overset{size(wms)}{\underset{i=(size(Wms)-5)}{Bs \cup}} wms(size(wms)-i)$
    For each pixel p
        If p.value∈ $Bs$ then
            Simg(p) = 256
        Else
            Simg(p) = 0
        End
    End
Stop
```

The segmentation algorithm finds the black-and-white mass values in the image. According to the different sets identified, the method detects the most optimal sets for grouping the pixels under two classes and produces a segmentation image.

C. CNN Training

The method reads the Brats image data set and applies MHNA preprocessing on each of them. The method applies an adaptive mass determination segmentation algorithm with the preprocessed image. The result of segmentation is used to extract the tumor's texture and computes black-and-white mass distribution values. All these features extracted are used to train the neural network. The CNN has been designed with an input layer to take the features and have two different convolution layers where the features are convolved into single-dimensional features. The max pooling layers help to shape the features and pad the features. Finally, the network is designed with a fully connected layer to support classification.

The texture obtained from the image has been split into four regions. According to the areas identified, the method would compute the gray mean value, which produces the following result.

TABLE IV. FEATURE OBTAINED FROM THE TEXTURE

R1	R2	R3	R4
167	185	175	169

Table IV shows the features obtained from the texture extracted from the brain image. Similarly, the method extracts the white and black mass distribution values to form the feature vector as follows:

TABLE V. EXAMPLE FEATURE VECTOR

R1	R2	R3	R4	WMD	BMD
167	185	175	169	76	45

Table V shows the example feature vector produced by the proposed model. The value of WMD is measured as follows:

$$\text{Wmd} = \frac{\sum_{i=1}^{size(T)} T(i)>180}{size(T)}$$

Similarly, the value of BMD is measured as follows:

$$\text{Bmd} = \frac{\sum_{i=1}^{size(T)} T(i)<100}{size(T)}$$

Accordingly, generated feature vector has been used to train the CNN. At the first convolution, the method computes the gray mean values at each region to produce four values, whereas in the second convolution; the method computes the average among the four values to produce a single value, reducing the feature vector size to three. Such feature vector has been used to compute the similarity measure at the test phase.

D. FDS Classification

The proposed model classifies the brain image according to the feature distributional similarity measured against image features. The method applies MHNA preprocessing, eradicating the image noise to perform this. Further, AMDS segmentation is applied to group the pixels of various features. From the segmented image, ROI has been extracted and estimates white and black mass distribution values. Such features extracted are passed through the CNN trained, where the features are convolved in two-stage and apply max pooling. The convolved features are obtained at the output layer, and the method computes the FDS values towards various features like texture, white mass, and black mass values. Using these values, the method computes the similarity value to compute anomaly class weight (ACW). According to the ACW value, the method performs classification.

```
Algorithm:
Given: Brain Image Bmg, CNN Tcn
Obtain: Class C
Start
        Read Bmg, Tcn.
        Pmg = MHNA-Preprocessing (Bmg)
        Smg = AMDS-Segmentation (Pmg)
        Wmd = compute White mass distribution.
        Bmd = Compute Black Mass distribution.
        Texture T = Extract tumor feature.
        Feature vector fv = {T, Wmd, Bmd }.
        Pass through CNN.
        Convolve Fv at stage 1.
        Perform max pooling.
        Perform stage 2 convolution.
        Perform max pooling.
        For each class a

        Compute Texture Distribution similarity TDS.
                $TDS = \frac{\overset{size(C)}{\underset{i=1}{Dist(C(i).T.value,T.value)}}}{size(C)}$
        Compute white mass distribution similarity Wds.
                $WDs = \frac{\overset{size(C)}{\underset{i=1}{Dist(C(i).wmd,T.wmd)}}}{size(C)}$
        Compute black mass distribution similarity Bds.
                $BDs = \frac{\overset{size(C)}{\underset{i=1}{Dist(C(i).bmd,T.bmd)}}}{size(C)}$
                Compute ACW = $\frac{Wds}{Bds} \times Tds$
        End
        Class C = choose the class with maximum ACW.
Stop
```

The proposed approach performs classification by computing ACW value for the sample towards various classes of brain tumors. Finally, a single class is selected according to ACW.

IV. RESULTS AND DISCUSSION

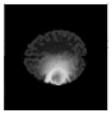
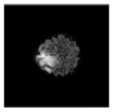
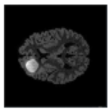
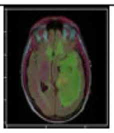
The proposed AFDS-CNN-based anomaly detection on brain images is enforced, and the performance of the model is measured against various classes with different constraints. The results achieved in each test case are discussed in this part. The method uses the Brats data set for performance evaluation.

TABLE VI. EVALUATION DETAILS

Factor	Value
Data Source	Brats 2019
Total Images	3000
No of Image Class	3
Platform	Python

Table VI denotes the constraints used in evaluating the performance of the models where the data set has three tumor classes, and the performance is measured on various factors.

TABLE VII. RESULTS OF CLASSIFICATION

Sl. No	Sample	Binarized Output	Label
1			Malignant
2			Benign
3			Benign
4			Malignant

The classification result produced for different brain images is plotted in Table VII, which has been used in measuring the accuracy of the method toward classification.

TABLE VIII. ANALYSIS OF CLASSIFICATION ACCURACY

Classification Accuracy in % vs. No of Samples			
	1000 samples	2000 samples	3000 samples
ADAM	74	78	82
TD-CNN-LSTM	77	81	86
3D-CNN	80	84	89
AFDS-CNN	85	91	97

The performance of methods in classification accuracy is analyzed in Table VIII, where the AFDS-CNN approach achieved higher classification accuracy in all the test cases.

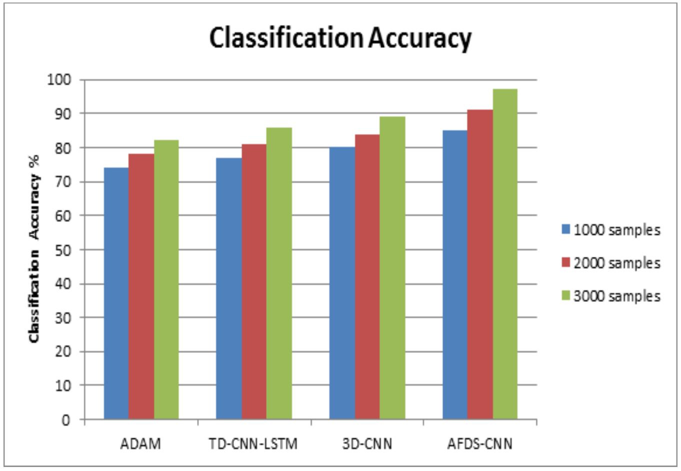


Fig. 2. Performance on classification accuracy.

The classification accuracy produced by different schemes is compared in Fig. 2, where AFDS-CNN achieved higher accuracy in all cases.

TABLE IX. ANALYSIS OF FALSE CLASSIFICATION RATIO

False Ratio in Classification % vs. No of Samples			
	1000 samples	2000 samples	3000 samples
ADAM	26	22	18
TD-CNN-LSTM	23	19	14
3D-CNN	20	16	11
AFDS-CNN	15	9	3

The false ratio introduced in brain image classification is measured and plotted in Table IX. The AFDS-CNN model achieves less false classification ratio compared to others.

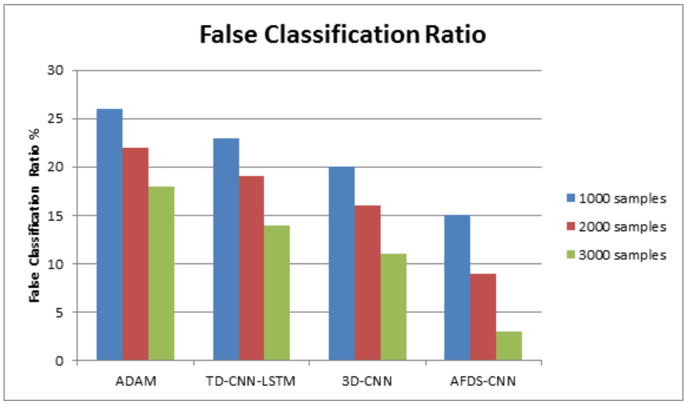


Fig. 3. Performance on false classification ratio.

The ratio of false classification is measured for different approaches according to the number of samples in the data set in Fig. 3. In each class, and the proposed AFDS-CNN has produced less false ratio than others.

TABLE X. ANALYSIS OF TIME COMPLEXITY

Time Complexity in Classification Seconds vs. No of Samples			
	1000	2000	3000
ADAM	54	61	86
TD-CNN-LSTM	43	52	77
3D-CNN	39	46	71
AFDS-CNN	21	27	32

The time complexity in classifying the images is measured and plotted in Table X. The RBP-CNN model produces little time complexity.

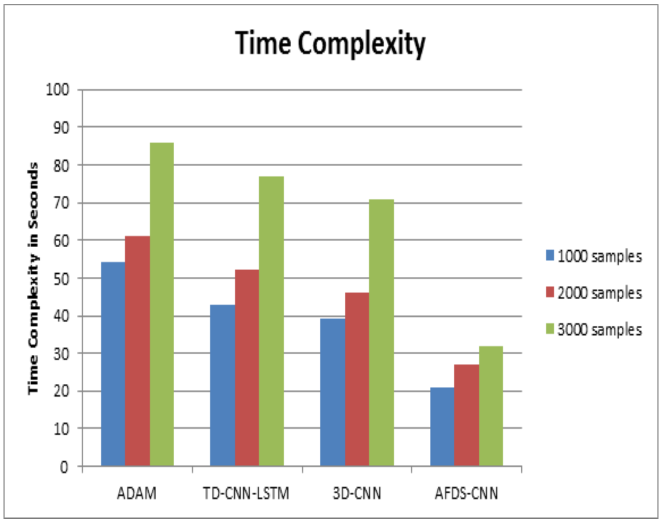


Fig. 4. Time complexity.

The time complexity in classifying the images is measured according to the total time taken for classifying the given image and plotted in Fig. 4. The RBP-CNN model produces negligible time complexity.

V. CONCLUSION

This paper presented a novel adaptive feature distribution similarity-based brain image classification with a convolution neural network (AFDS-CNN). The model applies MHNA preprocessing algorithm to the brain images to normalize the image features. Further, the method applies an adaptive mass discrimination segmentation algorithm to segment the images. With the segmented images, the method extracts texture, white mass distribution, and black mass distribution values. Extracted features are convolved with the CNN designed with different convolution and max pooling layers. At the test phase, the features extracted are used to compute texture distribution similarity (TDS), White mass Distribution Similarity (WDS), and Black mass distribution similarity (BDS) to compute the value of Anomaly class weight (ACW). Based on the value of ACW, the method performs classification and produces an accuracy of up to 98.6% with a reduced complexity of 21 seconds

VI. FUTURE WORK

The problem of brain tumor detection and disease prediction can be further improved by adapting time variant directional and distribution growth of tumor cells in classifying the brain image.

FUNDING STATEMENT

The authors received no specific funding for this study.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] J. Seetha, Selvakumar and Raja S, “Brain tumor classification using convolutional neural network,” Biomedical and Pharmacology Journal, vol. 11, no. 3, 2018. DOI: https://dx.doi.org/10.13005/bpj/1511.

[2] P. Chang, BWJ Grinband, MKM Bardis, G. Cadena, MY Su et al., “Deep-learning convolutional neural networks accurately classify genetic mutations in gliomas,” American Journal of Neuro Radiology, vol. 39, no. 7, pp. 1201-1207, 2018.

[3] D. J. Hemanth, J. Anitha, A. Naaji, O. Geman, DE. Popescu et al., “A modified deep convolutional neural network for abnormal brain image classification,” IEEE Access, vol. 7, pp. 4275 – 4283, 2019, doi: 10.1109/ACCESS.2018.2885639.

[4] LN Y. Liu, S. Stojadinovic, B. Hrycushko, Z. Wardak, L. Steven et al., “A deep convolutional neural network-based automatic delineation strategy for multiple brain metastases stereotactic radiosurgery,” PLoS ONE, vol. 12, no. 10, pp. 1–17, 2017.

[5] A. Sekhar, S. Biswas, R. Hazra, A. K. Sunaniya, A. Mukherjee and L. Yang, "Brain Tumor Classification Using Fine-Tuned GoogLeNet Features and Machine Learning Algorithms: IoMT Enabled CAD System," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 3, pp. 983-991, March 2022, doi: 10.1109/JBHI.2021.3100758.

[6] A. Kujur, Z. Raza, A. A. Khan and C. Wechtaisong, "Data Complexity Based Evaluation of the Model Dependence of Brain MRI Images for Classification of Brain Tumor and Alzheimer’s Disease," in IEEE Access, vol. 10, pp. 112117-112133, 2022, doi: 10.1109/ACCESS.2022.3216393.

[7] S. Asif, W. Yi, Q. U. Ain, J. Hou, T. Yi and J. Si, "Improving Effectiveness of Different Deep Transfer Learning-Based Models for Detecting Brain Tumors From MR Images," in IEEE Access, vol. 10, pp. 34716-34730, 2022, doi: 10.1109/ACCESS.2022.3153306.

[8] A. Vidyarthi, R. Agarwal, D. Gupta, R. Sharma, D. Draheim and P. Tiwari, "Machine Learning Assisted Methodology for Multiclass Classification of Malignant Brain Tumors," in IEEE Access, vol. 10, pp. 50624-50640, 2022, doi: 10.1109/ACCESS.2022.3172303.

[9] A. U. Haq et al., "IIMFCBM: Intelligent Integrated Model for Feature Extraction and Classification of Brain Tumors Using MRI Clinical Imaging Data in IoT-Healthcare," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 10, pp. 5004-5012, Oct. 2022, doi: 10.1109/JBHI.2022.3171663.

[10] Y. Ding et al., "MVFusFra: A Multi-View Dynamic Fusion Framework for Multimodal Brain Tumor Segmentation," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 4, pp. 1570-1581, April 2022, doi: 10.1109/JBHI.2021.3122328.

[11] S. Ahmad and P. K. Choudhury, "On the Performance of Deep Transfer Learning Networks for Brain Tumor Detection Using MR Images," in IEEE Access, vol. 10, pp. 59099-59114, 2022, doi: 10.1109/ACCESS.2022.3179376.

[12] T. Zhou, S. Canu, P. Vera and S. Ruan, "Latent Correlation Representation Learning for Brain Tumor Segmentation With Missing MRI Modalities," in IEEE Transactions on Image Processing, vol. 30, pp. 4263-4274, 2021, doi: 10.1109/TIP.2021.3070752.

[13] H. A. Shah, F. Saeed, S. Yun, J. -H. Park, A. Paul and J. -M. Kang, "A Robust Approach for Brain Tumor Detection in Magnetic Resonance Images Using FinetunedEfficientNet," in IEEE Access, vol. 10, pp. 65426-65438, 2022, doi: 10.1109/ACCESS.2022.3184113.

[14] M. Rizwan, A. Shabbir, A. R. Javed, M. Shabbir, T. Baker and D. Al-JumeilyObe, "Brain Tumor and Glioma Grade Classification Using

Gaussian Convolutional Neural Network," in IEEE Access, vol. 10, pp. 29731-29740, 2022, doi: 10.1109/ACCESS.2022.3153108.

[15] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim and A. Islam, "TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study," in IEEE Access, vol. 10, pp. 60039-60059, 2022, doi: 10.1109/ACCESS.2022.3179577.

[16] W. Wentao, D. Jiaoyang, G. Xiangyu, W. Gu, F. Zhao et al., “An intelligent diagnosis method of brain MRI tumor segmentation using deep convolutional neural network and SVM algorithm,” Hindawi, Computational and Mathematical methods in medicine, vol. 2020, 2020, https://doi.org/10.1155/2020/6789306.

[17] Zijianwang, Y. Sun, S. Qianzi and L. Cao, “Dilated 3d convolutional neural networks for brain MRI data classification,” IEEE Access, vol. 7, pp.134388–134398,2020,dol: https://doi.org/10.1109/ACCESS.2019.2941912.

[18] M. A. Sameer, O. Bayat and H. J. Mohammed, "Brain tumor segmentation and classification approach for MR images based on convolutional neural networks," in Proc. IEEE, Conference on, Information Technology To Enhance e-learning and Other Application (IT-ELA), pp. 138-143, 2020, Baghdad, Iraq, dol: 10.1109/IT-ELA50150.2020.9253111.

[19] M. Waqasnadeem, A. Mohammed, A. Ghamdi, M. Hussain, M. A. Khan, et al., “Brain tumor analysis empowered with deep learning: A review, taxonomy, and future challenges,” MDPI, Brain Sciences, vol. 10, no. 2, pp. 118, 2020, doi: https://doi.org/10.3390/brainsci10020118.

[20] M. Baskar, R. Renukadevi and J. Ramkumar, “Region centric minutiae propagation measure orient forgery detection with finger print analysis in health care systems,” Springer, Neural Process Letter, 2021, https://doi.org/10.1007/s11063-020-10407-4.(SCI).

[21] T. S. Arulananth, L. Balaji and M. Baskar, “PCA based dimensional data reduction and segmentation for DICOM images,” Springer, Neural Process Letter, 2020. https://doi.org/10.1007/s11063-020-10391-9.

An Algorithm Transform DNA Sequences to Improve Accuracy in Similarity Search

Hoang Do Thanh Tung, Phuong Vuong Quang
Institute of Information Technology, Vietnam Academy of Science and Technology, HaNoi, Viet Nam

***Abstract*—Similarity search of DNA sequences is a fundamental problem in the bioinformatics, serving as the basis for many other problems. In this, the calculation of the similarity value between sequences is the most important, with the Edit distance (ED) commonly used due to its high accuracy, but slow speed. With the advantage of transforming the original DNA sequences into numerical vector form that retaining unique features based on properties. The calculation processing on these transformed data will be much faster, many times faster than a direct comparison on the original sequence. Additionally, from a long DNA sequence, after transformation, it typically has a lower storage capacity, making it have good data compression. The challenge of this job is to develop algorithms based on features that maintain biological significance while ensuring search accuracy, which is also the problem to be solved. Previous methods often used pure mathematical statistics such as frequency statistics and matrix transformations to construct features. In this paper, an improved algorithm is proposed based on both biological significances and mathematical statistics to transforming gene data into numerical vectors for ease of storage and to improve accuracy in similarity search between DNA sequences. Based on the experimental results, the new algorithm improves the accuracy of similarity calculations while maintaining good performance.**

***Keywords*—Similarity search; data transformation; DNA sequence; big data**

I. Introduction

Bioinformatics is an interdisciplinary field that develops software methods and tools for understanding biological data, especially when the data sets are large and complex. It combines technologies of applied mathematics, statistics, computer science, biology, chemistry, physics… and biological mathematics. The term bioinformatics is a part of computational biology, and the combination of these sciences is intertwined and mutual. Thus, the research results in this field not only contribute to biology, but also to other fields. Biological data contains gene sequences (DNA - Deoxyribonucleic Acid) and briefly describes which is made up of four nucleotides: A, C, T and G. DNA is a crucial molecule for all living things, not just humans. The application of DNA in science and daily life is diverse and significant, such as crossbreeding, genetic mutation, comparison of species, prediction of phylogenetics, pedigree, genetic diseases, predict disease risk, etc. The fundamental problem in bioinformatics related to DNA sequence processing is similarity search, which is finding subsequences that are the same or similar to a sequence of interest. This leads to further problems such as calculating the similarity value between sequences to draw a phylogenetic tree and sequence alignment to achieve maximum similarity between sequences. These problems are manageable on small and medium data but become complex with big data.

In similarity search, the commonly used method to calculate similarity value between two sequences is Edit Distance (ED) (also known as Levenshtein). The *ED* similarity value between two sequences is the minimum number of steps required to transform one sequence to other, based on three transformations: adding, editing, and deleting each character in the sequence [1]. For example, similarity value between LOVE and MOVIE is ED(LOVE, MOVIE) = 2 because two steps of LOVE → **M**OVE → MOV**I**E are needed. The advantage of this method is can compare sequences of different lengths, thereby flexibly being applied in many problems. However, its computational complexity is *O(n*m)* where *m*, *n* are the lengths of two sequences being compared. With long data, the results take a long time to obtain, making comparisons in long sequences less efficient. Due to the rapid advancements in technology, biological databases generate vast amounts of information and are continuously growing, resulting in a rapid increase in their size.

The large size of the data leads to a high number of I/O operations, resulting in a high cost of storage space and decreased processing performance. To address this issue, researchers are exploring various methods to index or transform the gene data into numerical form to reduce database access and improve query performance. One of the challenges in this field is to develop algorithms for transforming the sequence in a way that retains the biological significance of the data while reducing its size. Previous methods often utilized simple mathematical statistics, such as frequency statistics or matrix transformations, to build features. In this study, an improved algorithm is proposed that not only takes into account frequency, position appearance, and correlation between position and distance of the characters but also considers the biological significance of the sequence, based on the role and function of amino acids in proteins. This approach helps to retain as many features of the sequence as possible, leading to improved accuracy when searching for similarity between two sequences.

The rest of the paper is organized as follows, Section II will present the related work to the research, Section III presents the proposed improved algorithm, Section IV presents the experimental results of the algorithm proposal, and Section V is the conclusion.

II. Related Work

A lot of research has been carried out to calculate similarity between sequences. Through the survey, a number of published methods were found. Some typical algorithms such as: Smith - Waterman [2] and BLAST [3] are well-known algorithms for performing sequence comparisons through representative sequences. This group of algorithms has high accuracy, but limitation is inefficient when size of database is too large or continuously added. Many other algorithms, improved from the famous BLAST algorithm, also allow performing calculations quickly such as Flash algorithm [4], ProperSearch tool [5], CAFE method [6]. BIS (Bitmap indexing structure) [7], IDC (Incrementally Decreasing Cover) [8], FRESCO (Framework for REferential Sequence Compresion) [9], Modified HuffBit Compress Algorithm [10]. These algorithms perform and compression search by interfering with data structures such as matrix analysis, genomic statistics, frequency changes, etc. In addition, Metric space indexing techniques [4], Williams & Zobel [11] and Ozturk and Ferhatosmanoglu [12] are indexing methods that use special transformations that convert the input DNA sequence into numerical vectors based on features. The objective of these methods is to find the most effective features to generate the vectors, so as to preserve the biological information of the original sequence and perform efficient similarity searches. Studies by Kahveci and Singh [13], Yongkun Li et al [14] have found that using frequency and position-distance characteristics resulted in better performance compared to other features.

The *Hfwd*^{2} method [1] is a technique that partitions a long DNA sequence into smaller subsequences using a window length, and then transforms these windows into numerical vectors that represent the frequency of appearance of the characters (A, C, T, G) as well as their combinations (N-grams). To enhance the features stored in the vector, this method also incorporates the position of each character in the sequence. To address the issue of uneven distribution of characters in the DNA sequence, the authors divide the sequence into two equal parts and generate the vectors based on these two parts to ensure that the comparison is more equal. While this method results in a lower accuracy compared to direct comparison due to the noise introduced during the transformation process, it offers a significant improvement in performance and speed during similarity searches. However, *Hfwd*^{2} has a few limitations, such as the use of a simple positional parameter, and the lack of consideration for the biological significance of the characters in the sequence. Below, we propose an improved algorithm from *Hfwd*^{2}, in which numerical vectors transformation is not only built with features based on frequency, position appearance, correlation between position and distance way of characters, but also interested in the biological significance of sequences based on amino acids, applied on wavelet transform to increase features stored after transformation. Theoretically, it is expected to improve accuracy of calculating similarity value between sequences.

Current methods are still continuing to research solutions to improve accuracy of similarity search with transformation vectors. In this paper, we propose an improved algorithm for transforming gene sequences into numerical vectors. This algorithm considers not only mathematical statistics such as frequency and position appearance, but also the biological significance of amino acids in the sequences. By doing so, we aim to maintain as much information as possible from the original sequence while improving the accuracy of similarity searches.

III. Proposed Method

The transformation of original sequences into numerical vectors has been discussed in the above sections. This approach can optimize storage in databases and increase computational performance. The challenge is to preserve the biological meaning of the sequences in the numerical vectors after transformation, in order to maintain search accuracy. To address this, we propose an improved algorithm compared to *Hfwd*^{2}. This new algorithm considers not just the frequency, position appearance, and correlation between position and distance of characters, but also the biological significance of the sequences based on amino acids. Additionally, the use of wavelet transform increases the amount of information stored in the numerical vectors. Theoretically, it is expected to improve accuracy of calculating similarity value between sequences.

In the pre-processing stage, we also partitioned the DNA sequences into equal length windows and transform each partition into numerical vectors using the feature parameters we have proposed. These vectors are then stored in a centralized database for later retrieval. When a new sequence needs to be compared for similarity, it will undergo the same pre-processing step and then be compared with the vectors in the database. The resulting set of similarities will be outputted. The general process of the method is shown in Fig. 1.

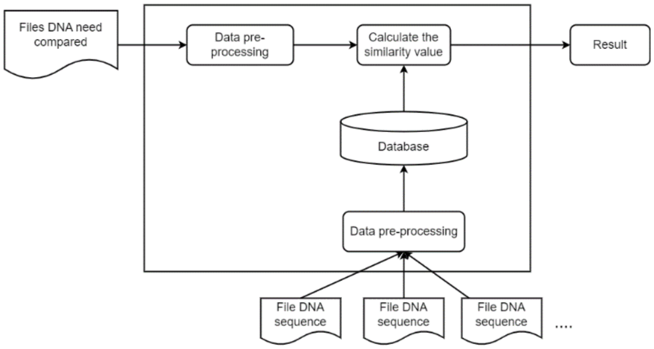


Fig. 1. General process of the method.

Files containing DNA sequences after being collected will be sent to the data pre-processing module. The system will remove any characters that are not A, C, T, G from the sequence, and partitioned the original sequence into equal length windows. The length of these windows can be adjusted to meet the user's needs, and we will explore the optimal range for best results in the experimental section. These operations can be easily performed using support functions from the C# language's String library. Finally, the partitions are subjected to transformation algorithms to generate multidimensional numerical vectors.

These vectors will be indexed so that when making comparisons based on similarity, they will be much faster than the original sequence. The pre-processing model is depicted in Fig. 2.

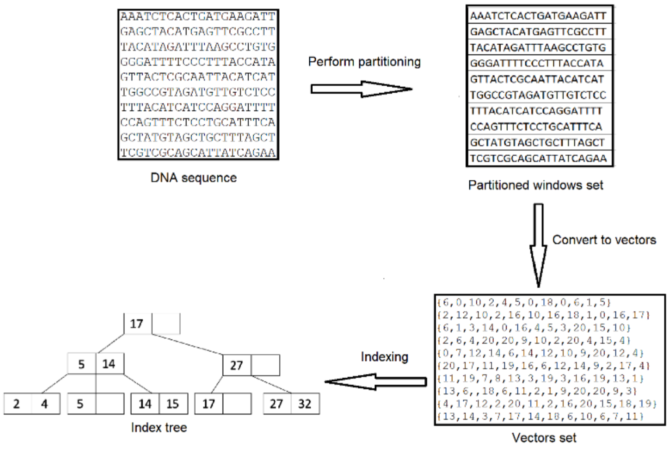


Fig. 2. Data pre-processing and indexing model.

A. Algorithms Feature Extraction for DNA Sequence Transformation

Just like feature extraction in natural language processing, the goal of this step is to show how to transform text data into vectors with numerical values. This is often the most important step that determines whether the final approach is successful for a real problem [15]. Here, we will present a proposal of four features that can be used to generate a numerical vector from the input DNA sequence. These features help to capture important information about the DNA sequence for perform similarity calculation.

Given the sequence S = {s_0, s_1, s_2, ..., s_{l-1}} where *l* is the length of the S, the algorithms to build feature of the sequence S are as follows:

1) Algorithm transform based on combinatoric: Let ω = {a_1, a_2, ..., a_α} where N-grams is the combination of characters built in the sequence S, α is length of the combination of characters. The DNA sequence S will only includes four nucleotides A, C, G, T, so when choosing N=1 we have the combination ω = {A, C, G, T}, when N=2 the combination ω = { AA, AC, AG, AT, TA, TC, TG, TT, GA, GC, GG, GT, TA, TC, TG, TT}, and so on. The larger N gets, the combination grows exponentially. Each element in the ω combination may or may not appear in the original sequence S, and may appear in different positions, which creates unique characteristics for each DNA sequence. This algorithm adds a parameter to calculate the correlation between position and distance compared to Hfwd[2]. The purpose is to store the parameters of frequency, position, and correlation between position and distance of characters that appear in the sequence, and use them as unique features of each sequence to calculate the similarity value later.

a) Frequency appearance

$$\mathrm{W}_{x\alpha} = \begin{cases} 1, x_\alpha = s_i \\ 0, x_\alpha <> s_i \end{cases} \tag{1}$$

Where i = 1,2…, l; $x_\alpha \in$ ω; if character x_α appear in position s_i under consideration, it has 1, otherwise it has 0.

$$a_{x\alpha} = \sum_{i=1}^{l} \mathrm{w}_{x\alpha}(s_i) \tag{2}$$

$a_{x\alpha} \geq 0$ is total appearance of the x_α in the sequence S.

b) Position appearance

$$b_{x\alpha} = \sum_{i=1}^{l} \mathrm{i} * \mathrm{w}_{x\alpha}(s_i) \tag{3}$$

$b_{x\alpha} \geq 0$ is sum of position values of x_α in the sequence S.

c) Correlation between position and distance

$$c_{x\alpha} = \frac{\sum_{i=1}^{l} (\mathrm{i} * \mathrm{w}_{x\alpha}(s_i)) * (\mathrm{i} * \mathrm{w}_{x\alpha}(s_i) - i_{pre} * \mathrm{w}_{x\alpha}(s_{i\_pre}))}{a_{x\alpha}} \tag{4}$$

$c_{\alpha\alpha} \geq 0$ is average of (position * distance), distance calculated as the difference between the current position and the previous position of x_α in the sequence S.

- N-grams selection

With DNA sequences, when N=1 the combination of characters has the value 4^1 = 4, when N=2 the combination has the value 4^2=16, when N=3 the combination has the value 4^3=64, etc. Increasing N can increase the amount of information stored in the vectors, but it also increases the computational cost of generating the vectors and calculating the similarity. According to a suggestion in [1], and based on our own experiments selecting N=1,2,3,4, we has been found that N=2 gives good results for comparisons and is also computationally efficient.

Algorithm 1. CombinatoricTranform(subsequence S)

```
INPUT: subsequence S
OUTPUT: a vector combinatoric 48 dimensions
BEGIN
    Initialize ω = {AA, AC, AG, AT, TA, TC, TG, TT, GA, GC, GG, GT, TA, TC, TG, TT}.
    Initialize vector vCombinatoric 48 dimensions
    FOR (i=0 to 15) {
        vCombinatoric[i] = Count frequency of element in ω (by fomula(2))  }
    FOR (i=16 to 31) {
        vCombinatoric[i] = Calculate sum of localtion of element in ω (by fomula(3))  }
    FOR (i=32 to 47) {
        vCombinatoric[i] = Calculate value of correlation between position and distance of element in ω (by fomula(4))  }
    RETURN vCombinatoric
END
```

The pseudocode for Algorithm 1 outlines the steps to transform an input DNA subsequence into a 48-dimensional numerical vector based on the formulas (1), (2), (3), and (4).

For example, with the sequence S = {AGTAGTGCTA}. We can calculate CombinatoricTranform(S) = {0, 0, 1, 0, 2, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0, 4, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 25, 0, 0, 0, 0, 4, 0, 6, 0, 0, 0, 0}.

2) Algorithm transform based on covariance: Covariance is a measure of how much two random variables vary together. It’s similar to variance, but where variance tells you how a single variable varies, covariance tells you how two variables vary together. Theoretically, covariance is used by analysts as a way to look at the overall for one or more related variables. In the context of biological and genetic data, variables are specific positions of nucleotides in the gene. Genetic covariance studies whether two positions of nucleotides on the gene evolve independently or they evolve together [16]. The authors in [17] also presented the idea of using covariance as a feature represented when transform the DNA sequence. This feature could provide additional information about the correlation between nucleotides in the gene, which could be useful in further analysis and comparison with other DNA sequences.

$$cov_{x,y} = \frac{\sum\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{N-1} \tag{5}$$

Where (x, y) ∈ {(A,T); (A;G); (A,C); (T,G); (T,C); (G,C)}, x_i, y_i is position of x, y on the sequence, $\bar{x}, \bar{y}$ is average position value of x, y in the sequence.

We use the covariance formula to build features through the vector { $cov_{A,C}, cov_{A,G}, cov_{A,T}, cov_{T,C}, cov_{T,G}, cov_{G,C}$ } showing correlation between the positions of the nucleotides together.

The pseudocode for Algorithm 2 explains how to transform an input subsequence into a 6-dimensional vector based on formula (5).

Algorithm 2. CovarianceTranform(subsequence S)

```
INPUT: subsequence S
OUTPUT: a vector covariance 6 dimensions
BEGIN
    Initialize ω = {(A,T); (A;G); (A,C); (T,G); (T,C); (G,C)}.
    Initialize vector vCovariance 6 dimensions
    FOR (i=0 to 15) {
        vCovariance[i] = Calculate value of correlation
        between position and distance of element in ω (by
        fomula(5))
    }
    RETURN vCovariance
END
```

For example, with the given sequence S, we can calculate CovarianceTranform(S) = { -1.33, -2.3, -3, -1.6, -2.8, -1.24}.

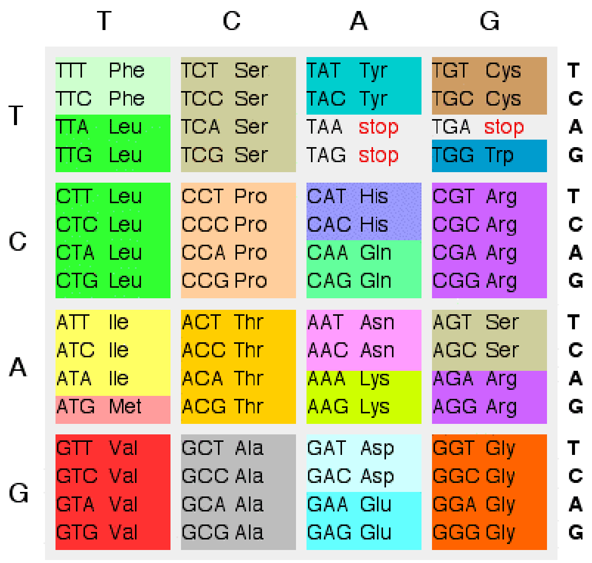
3) Algorithm transform based on Haar wavelet: The Discrete Wavelet Transform (DWT) is a widely used method in digital signal processing, data mining, information retrieval, text clustering and classification, digital image processing, etc. due to its simplicity and efficiency. DWT involves dividing a signal into two parts: high frequency and low frequency. The low frequency part is further divided into high and low frequency parts through a process called downsampling. The encoding complexity is linear and allows for multiple resolution levels. The Haar filter, being the simplest possible wavelet, is often used in the analysis of signals with abrupt transitions and is commonly used in analyzing time series or ordinal data. We use Haar filter to divide the original sequence into two parts: approximation (by summing) and detail (by subtracting) in pairs of values in in two vectors.

In the case that a character is unevenly distributed in different regions of the DNA sequence, for example, if character A is abundant in the first half but scarce in the second half, the algorithm for transforming based on these features can have lower efficiency. The goal of applying Haar wavelet is to overcome this limitation, improve the accuracy when calculating and comparing the similarity values of the transformed sequences. The pseudocode for Algorithm 3 presents the process of using Haar wavelet to transform vectors.

Algorithm 3. WaveletHaarTranform(vector u, vector v, integer *l*)

```
INPUT: vector u,v have l dimensions
OUTPUT: a vector l*2 dimensions
BEGIN
    Initialize vector vHigh, vLow l dimensions
    FOR (i=0 to l-1) {
        vHigh[i] = u[i] + v[i]
        vLow[i] = u[i] - v[i]
    }
    Initialize vector vWavelet l*2 dimensions
    FOR (i=0 to l-1) {
        vWavelet[i] = vHigh[i]
    }
    FOR (i=l to l*2 - 1) {
        vWavelet[i] = vLow[i]
    }
    RETURN vWavelet
END
```

4) Algorithm transform based on codon: The human body, along with other organisms, produces proteins through a process called transcription. Genes in biological cells store the information needed to construct proteins in the form of DNA sequences. During transcription, the DNA information is transcribed into messenger RNA (mRNA), which carries the genetic information. This mRNA sequence is then translated into a sequence of amino acids, which make up the proteins. Researchers have discovered that a group of three nucleotides encodes a single amino acid [18]. With four nucleotides, A, C, T, and G, there are 64 possible codons and 20 different amino acids.



	T	C	A	G	
T	TTT Phe	TCT Ser	TAT Tyr	TGT Cys	T
	TTC Phe	TCC Ser	TAC Tyr	TGC Cys	C
	TTA Leu	TCA Ser	TAA stop	TGA stop	A
	TTG Leu	TCG Ser	TAG stop	TGG Trp	G
C	CTT Leu	CCT Pro	CAT His	CGT Arg	T
	CTC Leu	CCC Pro	CAC His	CGC Arg	C
	CTA Leu	CCA Pro	CAA Gln	CGA Arg	A
	CTG Leu	CCG Pro	CAG Gln	CGG Arg	G
A	ATT Ile	ACT Thr	AAT Asn	AGT Ser	T
	ATC Ile	ACC Thr	AAC Asn	AGC Ser	C
	ATA Ile	ACA Thr	AAA Lys	AGA Arg	A
	ATG Met	ACG Thr	AAG Lys	AGG Arg	G
G	GTT Val	GCT Ala	GAT Asp	GGT Gly	T
	GTC Val	GCC Ala	GAC Asp	GGC Gly	C
	GTA Val	GCA Ala	GAA Glu	GGA Gly	A
	GTG Val	GCG Ala	GAG Glu	GGG Gly	G

Fig. 3. Table mapping three nucleotides to codon.

Fig. 3 shows how the various combinations of three bases in the coding strand of DNA are used to code for individual amino acids - shown by their three letter abbreviation. The combination of codons determines the sequence of amino acids in the resulting protein, which ultimately determines its structure and function. In this way, codons are the key to translating genetic information stored in DNA into the functional proteins that carry out vital cellular processes in all living organisms. Almost the same as above, we will use frequency appearance and correlation of position and distance to store the feature. The goal is to capture features related to biological significance and store them in vectors after transforming the input sequences. Thereby improving the efficiency when calculating and comparing the similarity value.

Let β a collection of amino acids, β = {Phe, Leu, Ile, Met, Val, Ser, Pro, Thr, Ala, Tyr, His, Gln, Asn, Lys, Asp, Glu, Cys, Trp, Arg, Gly}.

a) Frequency appearance

$$d_{y\alpha} = \sum_{i=1}^{l} w_{y\alpha}(s_i) \qquad (6)$$

$d_{y\alpha} \geq 0$ is total appearance of the y_α in the sequence S.

b) Correlation between position and distance

$$e_{y\alpha} = \frac{\sum_{i=1}^{l}(i * w_{y\alpha}(s_i)) * (i * w_{y\alpha}(s_i) - i_{pre} * w_{y\alpha}(s_{i\_pre}))}{d_{y\alpha}} \qquad (7)$$

$e_{y\alpha} \geq 0$ is average of (position * distance), distance calculated as the difference between the current position and the previous position of y_α in the sequence S.

The pseudocode of Algorithm 4 explains how to transform an input subsequence into a 40-dimensional vector based on formulas (6) and (7).

Algorithm 4. CodonTranform(subsequence S)

```
INPUT: subsequence S
OUTPUT: a vector codon 6 dimensions
BEGIN
    Initialize β = {Trp, Phe, Tyr, His, Asn, Asp, Cys, Gln, Lys,
    Glu, Ile, Val, Pro, Thr, Ala, Gly, Ser, Leu, Arg }.
    Initialize vector vCodon 40 dimensions
    FOR (i=0 to 19) {
        vCodon [i] = Count frequency of element in β (by
        fomula(6))
    }
    FOR (i=20 to 39) {
        vCodon [i] = Count frequency of element in β (by
        fomula(6))
    }
    RETURN vCodon
END
```

For example, with the given sequence S, will be calculate CodonTranform(S) = {0, 1, 0, 0, 2, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 7, 0, 0, 6.5, 4.5, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0}.

B. The Combine Algorithm Transforms DNA Sequences into Vectors

In Part A, we have presented four algorithms that will be used in vector transformation, this section will present an algorithm that combines the above algorithms to generate the final feature vector. Algorithm to calculate similarity value will be performed on these feature vectors.

Definition 1. Given the sequence S = {s0, s1, s2, ..., sl-1} where l is length of S, we define the vector FCC(S) as follows:

$$FCC(S) = [\text{CombinatoricTranform(S)}, \text{CovarianceTranform(S)}, \text{CodonTranform(S)}] \qquad (8)$$

The final feature vector *FCC(S)* is generated by merging the vectors produced by Algorithms 1, 2, and 4. Example, for sequence S = {AGTAGTGCTA}, we have *FCC(S) = { 0, 0, 1, 0, 2, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0, 4, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 25, 0, 0, 0, 0, 4, 0, 6, 0, 0, 0, 0, 0, 1, 0, 0, 2, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 7, 0, 0, 6.5, 4.5, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, -1.33, -2.33, -3, -1.6, -2.8, -1.24}*

Definition 2. Given the sequence S = {s0, s1, s2, ..., sl-1} where l is length of S, split S into Sa, Sb have equal length, we define the vector FCCW(S) as follows:

$$Va = [FCC(S_a)]$$

$$Vb = [FCC(S_b)]$$

$$FCCW(S) = [V_a + V_b, V_a - V_b] \qquad (9)$$

In case where S has an odd number of characters then length $S_b = S_a+1$, that is the last odd character will be in the subsequence S_b. The vectors V_a, V_b can be calculated from the FCC formula in Definition 1. Applying the Haar wavelet filter according to Algorithm 3, concatenate 2 vectors V_a, V_b to generate an *FCCW* vector containing the features of the original sequence. In this vector, it will contain information about the frequency, the position and the correlation between position and distance of the features mentioned above, through which the vector will be able to store the unique feature of the original sequence. In this way, an original subsequence of any length will be stored under a fixed length vector have 188 dimensions. Pseudocode for Algorithm 5 describes how to transform an input DNA sequence into a list of output final vectors.

Algorithm 5. TranformDNA(sequence S)

```
INPUT: DNA sequence S need tranform, window size W
OUTPUT: feature vectors
BEGIN
        //Calculate the number of windows will be cut from the Input string
        IF (Length(S) % W > 0){wCount = Length(S)/W +1}
        ELSE { wCount = Length(S)/W }
        //Partition sequence S into subsequences with window size
        FOR (i=0 to (wCount – 1))  {
                position = i*W
                listSub += Substring(S, position, W)
        }
        //Transform subsequences to vectors
        FOR (each subsequence in listSub)   {
                Devide a subsequence into two areas Sa, Sb
                Calculation Va = [FCC(Sa)], Vb = [FCC(Sb)] (by fomula(8))
                vectors += WaveletHaarTranform(Va,Vb) (by fomula(9)) }
        RETURN listVectors
END
```

Algorithm 6. *FCCWD*(vector u, vector v)

```
INPUT: vector u, v
OUTPUT: similarity FCCWD between u and v
BEGIN
        posDis = 0; negDis = 0;
        FOR EACH (each vector $u_i$, $v_i$ in u, v){
        IF($u_i > v_i$) {
                posDis += $u_i - v_i$ }
        ELSE {
                negDis += abs($u_i - v_i$) } }
        m = min (posDis, negDis)
        µ = | posDis  - negDis | / 2
        IF(m < µ) {
                RETURN µ/N   }
        ELSE {
                RETURN (µ + (m - µ))/N    }
END
```

The similarity value of two vectors is determined by the distance between the two vectors. To calculate this distance, the algorithm presented in [1] is employed. This algorithm calculates the distance between two vectors by finding the maximum number of operations required to transform from vector *u* to vector *v*. Algorithm 6 will calculate these values for each pair of *(u,v)* and storing the result in the variables posDis and negDis.

- Algorithmic complexity

The generated vectors will only contain numbers, which results in a computational complexity of O(*l*), much improved compared to O(l^2) of the *ED* algorithm.

IV. Experimental Results

In test model, we will take a text input DNA sequence and partition it into smaller subsequences using a selected window length. Next, we will use the *FCCWD* and *Hfwd*^{2} algorithms to transform these subsequences into a set of vectors, which will be stored in a general database. With these vectors, it will be possible to calculate the similarity values between them, and obtain the results. Lastly, we will compare the average results of the improved algorithm with the reference *ED* algorithm to evaluate the efficiency improvement achieved, as illustrated in Fig. 4.

We employed the use of Visual Studio 2019 and the C# programming language, as well as SQL Server 2019 for database management. The tests were conducted on a computer with the following configuration: CPU: Intel Xeon E2224G at 3.5 GHz, RAM: 16 GB, Hard Disk: Intel SSD 250 GB, Operating System: Windows Server 2019.

The test dataset is 171 genes of the family Poxviridae virus (32.9 MB), 172 genes of the family Asfarviridae virus (31.4 MB), 203 genes of the family Herpesviridae virus (32.6 MB), 23 genes of the family Corona virus (0.9 MB) available for download at the website https://www.ncbi.nlm.nih.gov/.

As the scenario has been built, we will compare the results of the similarity value calculation using three methods, with the standard *ED* method being the target to aim for. We will partition the data file into subsequences of different window lengths W = (1000, 2500, 5000, 7500, 10000, 15000, 20000) and evaluate the efficiency based on the average of the calculations. Since the parameters differ, the similarity values may not be on the same frame of reference. To compare the results, we will use the *k* nearest neighbor method on the resulting data. This means, given an input DNA sequence, we will find the cluster of *k* sequences with the closest similarity value for each method. The intersection of the resulting *FCCWD* clusters with *ED* and *Hfwd*^{2} with *ED* will represent the accuracy of the algorithms. As an example, if we search for similarities using the standard *ED* method, the result will be a set A with *k* elements having minimum values. The *FCCWD* method will find a set B also with *k* elements. The intersection of A and B (A ∩ B) will be the elements that *FCCWD* correctly found compared to *ED* as illustrated in Fig. 5.

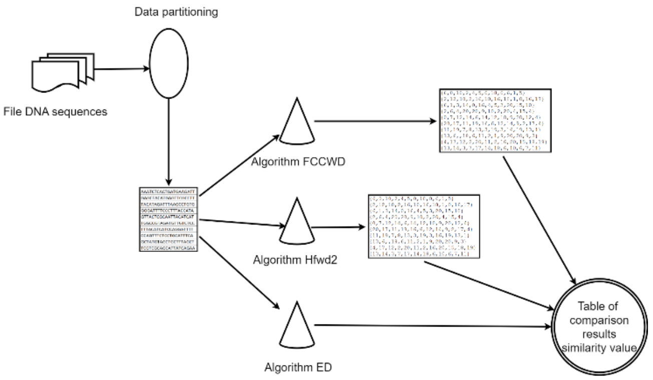


Fig. 4. Model of experiment.

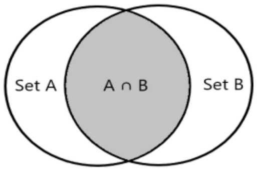


Fig. 5. Example of intersection of 2 result sets.

Fig. 6 and Fig. 7 illustrate the results for scenarios where k equals 5 and 10. The average true positive rate of *FCCWD* is higher than *Hfwd*^{2} under these scenarios. The efficiency of method is highest when the window length is between 5000 and 10000, but noticeably decreases when the window length is smaller than 5000 or larger than 10000.

Fig. 8 and Fig. 9 present the results for scenarios where k is equal to 15 and 20. Although performing worse than *Hfwd*^{2} at W = 1000 with k=20, but the average true positive rate of *FCCWD* still better than *Hfwd*^{2} in the remaining cases. The efficiency of method remains low for window lengths smaller than 5000, but proves to be stable when window lengths are larger than 5000.

The results indicate that when the window length is small, accuracy is lower and gradually improves as the window length increases. With small window lengths, the small number of characters creates high noise levels, leading to larger errors. When the sequence is longer, the accuracy of *FCCWD* is higher because we have added more parameters to store the eigenstate better than the *Hfwd*^{2} method. Therefore, this method is also more efficient when comparing large sequences than *Hfwd*^{2} tested on small windows from 100 to 1000. The best accuracy of the proposed method is the windows length from 5000 to 10000.

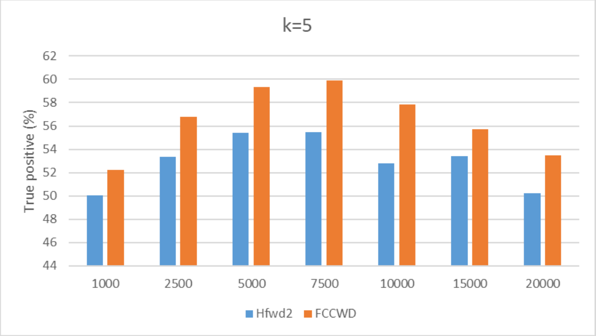


Fig. 6. Compare result with k = 5.

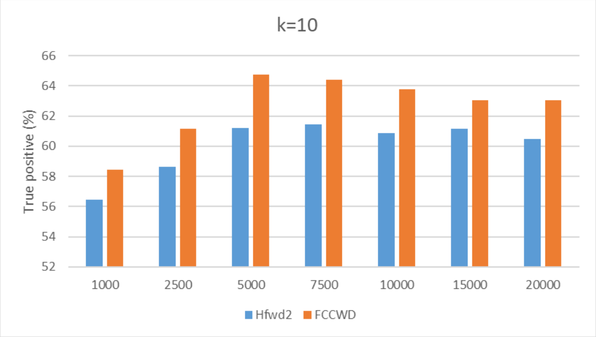


Fig. 7. Compare result with k = 10.

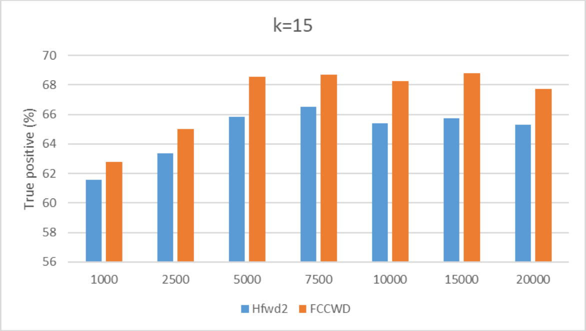


Fig. 8. Compare result with k = 15.

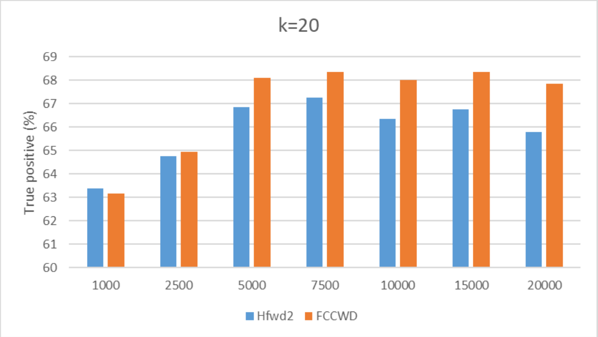


Fig. 9. Compare result with k = 20.

TABLE I. EXECUTION TIME OF METHODS (SECONDS)

W size **Method**	**1000**	**2500**	**5000**	**7500**
ED	25.0029	155.7917	619.9818	1387.816
FCCWD	0.0003	0.0007	0.0014	0.0021
Hfwd^2	0.0003	0.0007	0.0013	0.002
W size **Method**	**10000**	**15000**	**20000**	
ED	2457.2346	5496.9212	9730.9857	
FCCWD	0.0028	0.0042	0.0055	
Hfwd^2	0.0027	0.004	0.0053	

Regarding the computational performance shown in Table I and Fig. 10, the computational performance of both *FCCWD* and *Hfwd*^{2} methods is observed to be highly efficient, with computation times only being a few hundredths of a second. On the other hand, *ED* is significantly slower, ranging from a few tens to tens of thousands of seconds. Although *FCCWD* uses more parameters and has a larger vector dimension, the similarity algorithm has linear complexity, making the increase in cost minimal compared to *Hfwd*^{2}. As the window length of the data increases, the computation time of *FCCWD* increases very little, while *ED* increases very quickly.

The experiment also shows that the data compression ratio is quite good, with the larger window length the more compression, as shown in Fig. 11. Because how large a window is, it will also be transformed into a vector with 188-dimensions. This can help the data after transformation to be easily stored and reused in the next times.

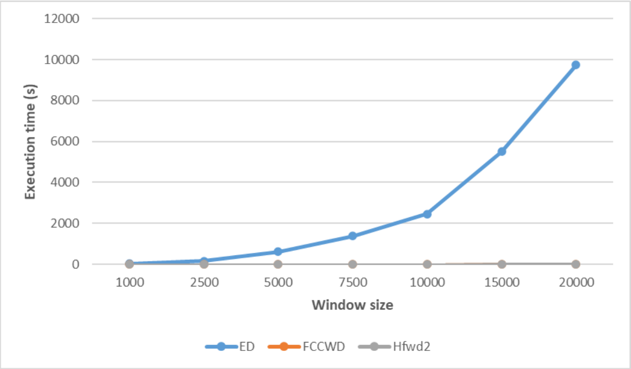


Fig. 10. Time comparison results.

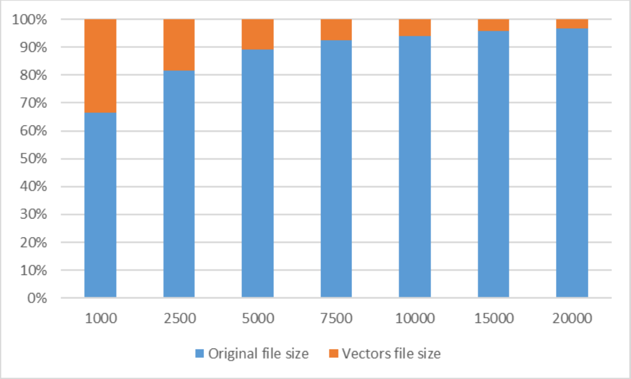


Fig. 11. Compare files size after transformation.

V. Conclusion

With the goal of improving the accuracy of similarity search, this paper propose an improved algorithm that transforms DNA sequences into numerical vectors using multiple feature parameters. These features are a blend of mathematical statistics and biological characteristics of genetic genes, allowing for a better representation of the original sequence information in the transformed vectors. Additionally, the method also enables efficient storage and reuse of the transformed data through reduction of its size. Experimental results demonstrate that the new algorithm improves similarity calculation accuracy while maintaining good performance. Although some cases of high noise still affect accuracy, the algorithm performs better in the case of long window length. In the future, we will continue to look for valuable parameters to further improve accuracy, as well as apply other indexing methods to process large data better.

Acknowledgment

This paper is supported by project CSCL02.06/22-23 of Institute of Information Technology (IoIT) - Vietnam Academy of Science and Technology (VAST), Hanoi, Vietnam.

References

[1] In-Seon Jeong, Kyoung-Wook Park, Seung-Ho Kang, Hyeong-Seok Lim, “An efficient similarity search based on indexing in large DNA databases,” Computational Biology and Chemistry, vol 34, pp.131-136, 2010.

[2] Zeyu Xia, Yingbo Cui, Ang Zhang, Tao Tang, Lin Peng, Chun Huang, Canqun Yang & Xiangke Liao, “A Review of Parallel Implementations for the Smith–Waterman Algorithm,” Interdisciplinary Sciences: Computational Life Sciences, vol 14, pp.1–14, 2022.

[3] Samer Mahmoud Wohoush, Mahmoud Hassan Saheb, “Indexing for Large DNA Database Sequenes,” International Journal of Biometrics and Bioinformatics (IJBB), vol 5, pp.202-215, 2011.

[4] T. Magoc and S. Salzberg, “FLASH: Fast length adjustment of short reads to improve genome assemblies,” Bioinformatics, pp.2957-2963, 2011.

[5] Xianyang Jiang, Peiheng Zhang, Xinchun Liu, Stephen S.-T.Yau, “Survey on index based homology search algorithms,” Springer Science + Business Media, LLC, pp.185-212, 2007.

[6] Hugh Williams, Justin Zobel, “Compression of nucleotide databases for fast searching,” Bioinformatics, pp.549–554,1997.

[7] Ooi BC, Pang HH, Wang H, Wong L, Yu C, “Fast filter-and-refine algorithms for subsequence selection,” Proceedings of the 6th international database engineering and applications symposium (IDEAS’02), Edmonton, Canada, pp.243–254, July 2002.

[8] Lee HP, Tsai YT, Sheu TF, Tang CT, “An IDC-based algorithm for efficient homology filtration with guaranteed seriate coverage,” Fourth IEEE symposium on bioinformatics and bioengineering (BIBE’04), Taichung, Taiwan, pp.395-402, 2004.

[9] Jim Dowling, KTH, “Reference Based Compression Algorithm”, Scalable, Secure Storage of Biobank Data, Work Package 2, pp.23 – 44, June 2014.

[10] Nahida Habib, Kawsar Ahmed , Iffat Jabin, Mohammad Motiur Rahman, “Modified HuffBit Compress Algorithm - An Application of R,” Journal of Integrative Bioinformatics, pp.1-13, Feb 2018 .

[11] Williams, Zobel, “Indexing and retrieval for genomic databases,” IEEE Transactions on Knowledge and Data Engineering Vol 14, pp.63–78, 2002.

[12] Ozturk, Ferhatosmanoglu, “Effective indexing and filtering for similarity search in large biosequence database,” Third IEEE Symposium on Bioinformatics and Bioengineering, Proceedings, pp.359–366, 2003.

[13] Kahveci, Singh, “An efficient index structure for string databases,” Proceedings of 27th International Conference on Very Large Data Base, Roma, Italy, pp.351–360, 2001.

[14] Yongkun Li, Lily He, Rong Lucy He & Stephen S.-T.Yau, “A novel fast vector method for genetic sequence comparison,” Scientific Reports, pp.1-11, 2017.

[15] Qing Zhou and Jun S. Liu, “Extracting sequence features to predict protein–DNA interactions: a comparative study,” Nucleic Acids Research, 36(12), pp.4137–4148, 2008.

[16] Stanley Maloy. Kelly Hughes, “Brenner's Encyclopedia of Genetics (Second Edition) ,” Elsevier, pp.242-245, 2013.

[17] Rui Dong, Lily He, Rong Lucy He, and Stephen S.-T. Yau1, “A Novel Approach to Clustering Genome Sequences Using Inter-nucleotide Covariance,” Frontiers in Genetics, pp.1-12, 2019.

[18] N. N. Kozlov, “The Study of the Secrets of the Genetic Code,” Journal of Computer and Communications, pp.64-83, 2018.

An Automated Text Document Classification Framework using BERT

Momna Ali Shah[1], Muhammad Javed Iqbal[2], Neelum Noreen[3], Iftikhar Ahmed[4]
Department of Computer Sciences, UET Taxila, Pakistan[1, 2]
Department of Computer and Information Sciences, Gulf Colleges, Hafr Al Batin, Saudi Arabia[3]
Department of Information Technology, King Abdul Aziz University, Jeddah, Saudi Arabia[4]

***Abstract*—Due to the rapid advancement of technology, the volume of online text data from numerous various disciplines is increasing significantly over time. Therefore, more work is needed to create systems that can effectively classify text data in accordance with its content, facilitating processing and the extraction of crucial information. Since these non-automated systems use manual feature extraction and classification, which is error-prone and time-consuming by choosing the best appropriate algorithms for feature extraction and classification, traditional procedures are typically resource intensive (computational, human, etc.), which is not a viable solution. To address the shortcomings of traditional approaches, we offer a unique text categorization strategy based on a well-known DL algorithm called BERT. The proposed framework is trained and tested using cutting-edge text datasets, such as the UCI email dataset, which includes spam and non-spam emails, and the BBC News dataset, which includes multiple categories such as tech, sports, politics, business, and entertainment. The system achieved the highest accuracy of 91.4% and can be used by different organizations to classify text-based data with a high performance. The effectiveness of the proposed framework is evaluated using multiple evaluation metrics such as Accuracy, Precision, and Recall.**

Keywords—Deep learning; text classification; BERT

I. Introduction

Text classification is a common problem in Natural Language Processing (NLP) that aims to classify the text data based on its content. This field has become drastically important due to increase in text based data. The increase in internet usage has resulted in the creation of diversified text data that is made available by numerous social media platforms and websites in different languages. This has resulted in exponential rise in the number of complex documents and texts that demand a deeper understanding of machine learning approaches to effectively identify texts in numerous applications. This field has wide range of applications such as sentiment analysis, email classification, news classification, movie review prediction, etc. [1, 2].

In NLP, numerous ML techniques have been developed over the past few years. A typical text classification system has four stages: preprocessing, feature extraction, feature selection, and classification. These applications must solve a number of issues relating to the nature and organization of the underlying textual information by condensing word variants into short representations while retaining the majority of the linguistic properties. However, there are certain limitations in the traditional methods. Firstly, it is difficult to capture text semantics using these techniques since they solely focus on word frequency attributes and completely ignore the contextual information stored in text. Second, the success of these statistical approaches in machine learning is often dependent on hand-crafted feature extraction and classification, which is time-consuming and error-prone. Moreover, it can be difficult for researchers to develop such pipelines and methods for text classification that can perform better [3, 4].

Hence, due to these problems, recent years have seen a complete shift from these traditional text classification methods towards much stronger state-of-the-art DL based methods. These algorithms do not require a feature extraction phase prior to data classification, as these systems are completely automated because these models are highly capable of extracting robust features from the dataset themselves during the learning phase. Due to which, the deep learning algorithms have achieved state-of-the-art performance in a variety of NLP tasks, hence, the researchers are keen in exploring the applicability of these algorithms in different tasks like question/answering, email classification, news categorization and much more [5, 6].

In this paper, we proposed a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) architecture for text classification. BERT is a brand-new language representation model that Google has introduced in 2018 [6, 7]. The model has succeeded in achieving state-of-the-art performance on text classification problems, hence, has increased the interest of researchers in fine-tuning and deployment of BERT on various text classification problems. In this paper, we fine-tuned BERT architecture on two state-of-the art datasets composed of email and news. The proposed framework is discussed in detail in Section III. The contributions of the proposed study are as follows:

- We developed an automated text classification framework to classify different types of text data.
- The proposed method initially preprocesses data by removing stop-words and extra characters so that classification performance can be improved.
- The preprocessed text is then classified via widely known DL based architecture called BERT by fine-tuning the architecture on our problem.

- The performance of the proposed technique using different evaluation parameters and compare its performance with existing systems.
- In this study, we performed extensive experiments on publically available datasets to show the efficiency and robustness of our algorithm.
- Both of the proposed methods can accurately detect and classify text data effectively and can be deployed by various organizations to classify the text data.

The remaining paper is organized as follows. The literature is critically analyzed in Section II. The proposed methodology is discussed in detail in Section III. Whereas, Section IV evaluates the performance of the proposed technique and compares it with state-of-the-art methods. The study is concluded in Section V.

II. Literature Review

Text classification is a common NLP task that has a wide range of uses, such as sentiment analysis, email classification, detection of offensive language, spam filtering, etc. Now-a-days ML has become a subject of interest for text classification tasks, as these algorithms have shown considerable potential for acquiring linguistic knowledge.

A typical text classification system has four stages: preprocessing, feature extraction, feature selection, and classification. These applications must address a number of issues relating to the nature and structure of the underlying textual information for languages by translating word variants into compact representations while retaining the majority of the linguistic properties. However, these systems have several issues. Firstly, it is difficult to capture text semantics using these techniques since they solely focus on word frequency attributes and completely ignore the contextual structure information in text. Second, the effectiveness of these statistical methods for machine learning frequently depends on challenging technical features and the usage of vast linguistic resources.

The authors in Jang et al. [8] employed MLP to classify textual data. The authors succeeded in achieving 71% accuracy on MLP. However, the performance should be improved. She et al. [3] proposed a hybrid technique that solves CNN's fundamental limitation in expressing long-term contextual information while utilizing CNN's capacity to extract local data. Additionally, the model makes an effort to address LSTM's inherent flaws, which include its tendency to process data sequentially and rank as the poorest feature extractor. When compared to counterpart models, the hybrid model performed better, but its findings lagged below models that make use of an attention mechanism in terms of interest.

Urdu editorials were also classified by Sattar et al. [1] using NB. The authors reduced the dimensionality by eliminating terms with common frequency. With their study, they were able to prove that when Naive Bayes classifier is supplied text with frequent terms it outperforms the model when it isn’t supplied those terms. However, these studies need to be incorporated on multiple Urdu categories rather than only headline classification. The authors in Antoun, et al. [7] used BERT model for Arabic text classification called Arabert which was trained on 24 gigabytes of data. Similarly, in another research, Abdul-Mageed, et al. [4] trained Arabic BERT architecture called MARBERT on 1B tweets. However, the systems mentioned in [4, 7] are computationally expensive.

Koswari et al. [5] proposed an ensemble approach using deep learning algorithms to classify text from news dataset and achieved 87% accuracy. However, the system obtained a low overall performance, hence, its accuracy should be improved. Cai et al. [9] classified news data by employing several deep learning architectures such as RCNN, CNN and RNN. Similarly, the study presented by Lenc et al. [6] proposed the use of CNNs as well as a simple multi-layer perceptron to extract features from Czech newspaper documents before applying multi-label document classification. This technique achieved F1 score of 0.84 using MLP with sigmoid functions. However, these studies only classify news data, hence need to test their architectures on other types of text data before deploying in real-world scenario.

III. Materials and Methods

Due to its vast applicability in businesses and organizations, text classification has become a very significant research area in NLP. The text classification algorithms aim to classify the text data based on its content and meta-data contained in it. This can be achieved by using ML and DL based algorithms to automate the process with an increase in data volumes. In this paper, we propose a novel and robust text classification framework employing a well-known DL based algorithm called BERT. The pipeline of proposed architecture is illustrated in Fig. 1.

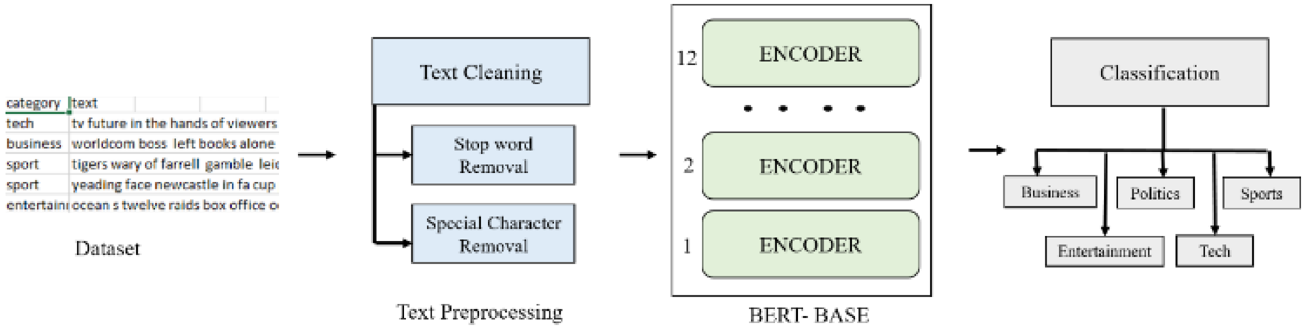

category
text
tech
tv future in the hands of viewers
business
worldcom boss left books alone
sport
tigers wary of farrell gamble leic
sport
yeading face newcastle in fa cup
entertaini
ocean s twelve raids box office o
Dataset
Text Cleaning
Stop word Removal
Special Character Removal
Text Preprocessing
12
ENCODER
2
ENCODER
1
ENCODER
BERT- BASE
Classification
Business
Politics
Sports
Entertainment
Tech

Fig. 1. Pipeline of proposed framework.

The proposed architecture is trained and evaluated on two publically available datasets. Initially, we cleaned the datasets by removing stop-words and special characters. Furthermore, we also converted the entire text in small case before actual processing. The cleaned dataset is then supplied to BERT architecture for feature extraction and classification.

A. Data Collection

The proposed framework is trained and evaluated on publically available datasets obtained from different sources. The BBC News dataset consists of a total of 2225 documents that consists of five classes namely business, entertainment, politics, sport, tech that was obtained from 2004-2005 [10]. Fig. 2 shows the dataset distribution showing the class name and number of documents in that class.

The second dataset is gathered from UCI Database [11]. We also performed exploratory data analysis of the second dataset. The database is composed of 5569 emails, of which 745 are spam and others are non-spam. Hence, non-spam emails count for 12% of the dataset and spam emails count for 88% of the whole database. The dataset is highly imbalanced, hence, in these recall and precision as an evaluation metrics are very useful. However, it may be noted that before supplying the database to proposed BERT architecture, we balanced the dataset and randomly chose equal numbers of instances from both classes to avoid the biasness in the classification architecture.

B. Data Preparation

Data cleaning is an essential phase in any NLP task, which aims to modify data in a format that is much easier for the algorithm to analyze or predict. In this phase, we cleaned the dataset by removing special characters and stop words. Special characters and symbols consist of non-alphabet letters such as “([/ (] [| @],]”. Whereas, stop-words are a group of terms that are used frequently in a sentence or used to link sentences, some of these include "a," "the," "is," and "are." These terms need to be eliminated because they provide no information to the model but can be a cause of poor performance of any text classification model. Furthermore, the entire dataset is also converted in small case to help remove any ambiguity during learning process. Fig. 3 shows the preprocessing steps applied to the datasets.

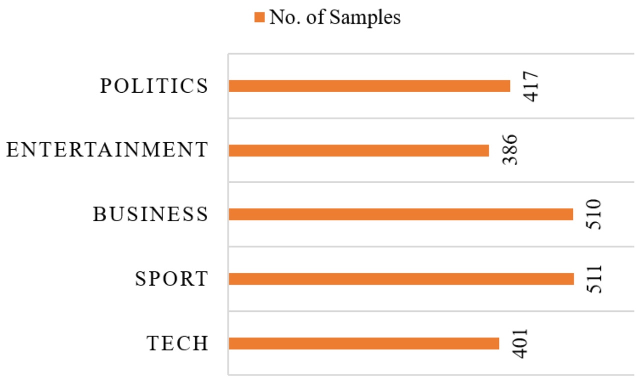
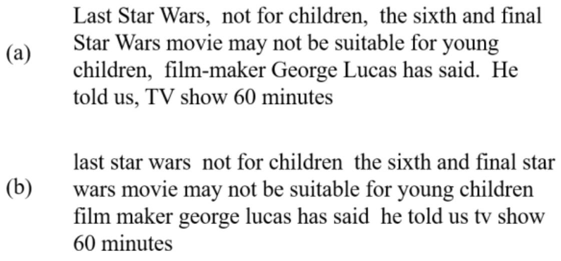


Fig. 2. No. of text samples in BBC news dataset.



(a) Last Star Wars, not for children, the sixth and final Star Wars movie may not be suitable for young children, film-maker George Lucas has said. He told us, TV show 60 minutes

(b) last star wars not for children the sixth and final star wars movie may not be suitable for young children film maker george lucas has said he told us tv show 60 minutes

Fig. 3. Dataset preparation, (a) Non-preprocessed text, (b) Preprocessed text.

C. Proposed Framework Design

The field of NLP focuses on developing computing methods to automatically interpret and represent human language. For a very long period, the bulk of approaches to examine NLP issues relied on labor-intensive, hand-crafted features and shallow machine learning models. As a result of linguistic information being represented via sparse representations, issues like the curse of dimensionality began to arise due to high-dimensional feature vectors. However, these issues in the traditional methodologies have been solved, thanks to advent of DL based algorithms such as Convolutional Neural Networks, Recurrent Neural Networks, etc. [12]. But, one of the major issues faced in DL architectures is lack of training data. The majority of task-specific datasets only contain some human-labeled training samples because NLP is a diverse area with numerous separate jobs. Modern DL-based NLP models, on the other hand, have improved on larger volumes of data containing millions, or billions, of annotated instances. Over the past decade, researchers have created a number of methods for training general purpose language representation models using the huge volume of content from the web in order to close this data gap. The models trained on massive datasets can now be utilized on smaller problem such as question/answering or sentiment analysis, etc. rather than training models from scratch [2].

BERT, proposed in 2018 by Google AI Language researchers created quite a stir in the ML community as it achieved good results in a wide range of NLP tasks. The framework is intended to assist computers in understanding the meaning of ambiguous words in textual data by establishing context through the use of surrounding material [13, 14]. The architecture of BERT is built using Transformers, where each output element is coupled to each input element and the weights between them are dynamically calculated based on their connection. Earlier language models could only read text input in one of two directions i.e. either left to right or from right to left, but not both simultaneously. However, BERT can read data simultaneously in both directions mainly due to transformers that help its enhanced understanding of linguistic ambiguity and context. Furthermore, earlier approaches like word2vec would map every word to a vector, which only captures a small fraction of its meaning in one dimension which is known as word embedding. But BERT is the first NLP technique that completely relies on self-attention techniques because of the bidirectional Transformers at its core that helps it understand complete meaning as the paragraph develops [15]. This capability of directionality enables the BERT to eliminate the left-to-right momentum due to which the words are usually biased towards a particular meaning as a phrase proceeds, hence reading from both directions, accounts for the impact of all other words on the focus word, and compensates for the augmented meaning [13].

In this paper, a refined BERT base architecture is suggested (shown in Fig. 4) for the text classification problem. BERT-base has 110 parameters and was trained on an English language corpus. BERT-base contains 12 encoders layered on top of each other. BERT-Base features a larger feed forward network with 768 hidden units. In addition, the structure contains 12 attention heads. The system gets computationally expensive as the number of encoders and parameters rises. For these reasons, we chose the BERT-base model because it is lightweight and quick to train.

D. Experimental Configuration and Setup

We trained and evaluated our proposed BERT architecture on publically available datasets i.e. UCI Email dataset composed of Spam and Non-Spam emails. The second dataset consists of BBC News text dataset composed of 5 different classes namely tech, entertainment, sport, politics and business. The model is tested on different hyper-parameters and performed the best on 10 epochs, mini-batch size of 32, a learning rate of 0.001 and a dropout rate of 0.1 (meaning 10% of the random nodes are dropped during training process to lighten up the network). In this study, 75% of the dataset is used for training the model, whereas 25% of the dataset is used for testing purposes. The entire experiment is performed on Python using Anaconda software on a PC with 8GB RAM and Intel Core i5 processor.

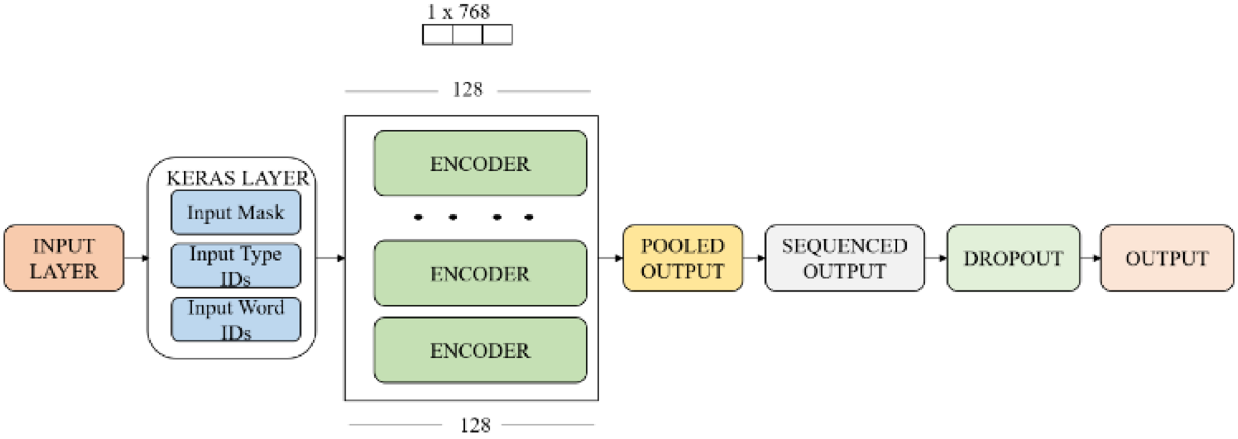


Fig. 4. Proposed framework architecture.

IV. Proposed Method Results

A. Evaluation Parameters

The proposed method is evaluated using different metrics such as precision, recall and accuracy. Confusion matrices help in showing tabular counts of observed and expected values. Different evaluation matrices such as True Positives, True Negatives, False Positives and False Negatives can be calculated using confusion matrices as well. TN depicts the total number of negative cases that were correctly identified. Similar to this, TP denotes the accurately identified positive cases. FP shows the negative cases that were by mistake classified as positive, while FN shows the positive instances wrongly classified as a negative [16, 17]. The value for accuracy, precision and recall can be calculated from the following equations.

$$\text{ACC} = \frac{\text{TN+TP}}{\text{TP+FN+TN+FP}} \quad (1)$$

$$\text{REC} = \frac{\text{TP}}{\text{FN+TP}} \quad (2)$$

$$\text{PRE} = \frac{\text{TP}}{\text{FP+TP}} \quad (3)$$

B. Experiment # 01: Classification of Emails using BERT

In this study, we employed BERT on email dataset containing both spam and non-spam emails. We initially cleaned the dataset before feeding it to BERT architecture. In this study, we employed cased BERT architecture so we changed the text to smaller case and then removed the stop words, keywords, etc.

The proposed method achieved a training accuracy of 92.3% whereas values obtained from precision, recall and f1score are 0.92 and 0.91 respectively as shown in Fig. 5. Whereas, the system achieved testing accuracy, precision and recall of 91.2%, 0.91 and 0.91 respectively as shown in Fig. 6. The confusion matrix of the proposed technique is shown in Fig. 7.

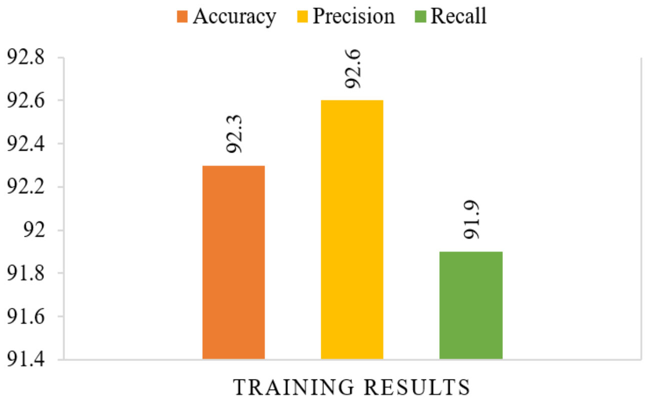


Fig. 5. Training results on UCI email dataset.

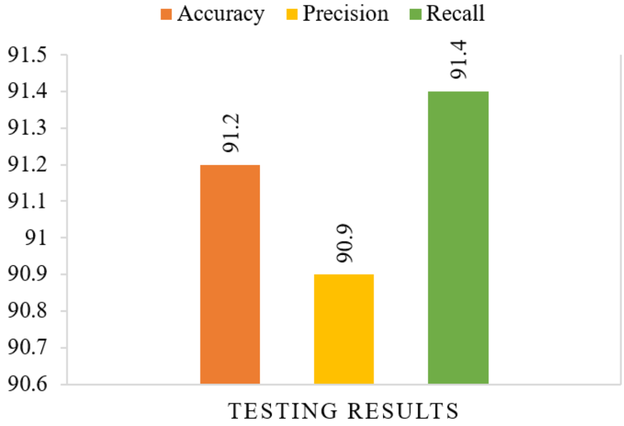
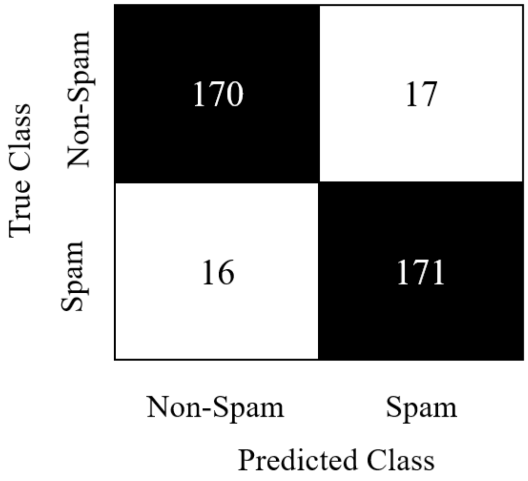


Fig. 6. Testing results on UCI email dataset.


True Class
Predicted Class

	Non-Spam	Spam
Non-Spam	170	17
Spam	16	171

Fig. 7. Confusion matrix obtained from the proposed method.

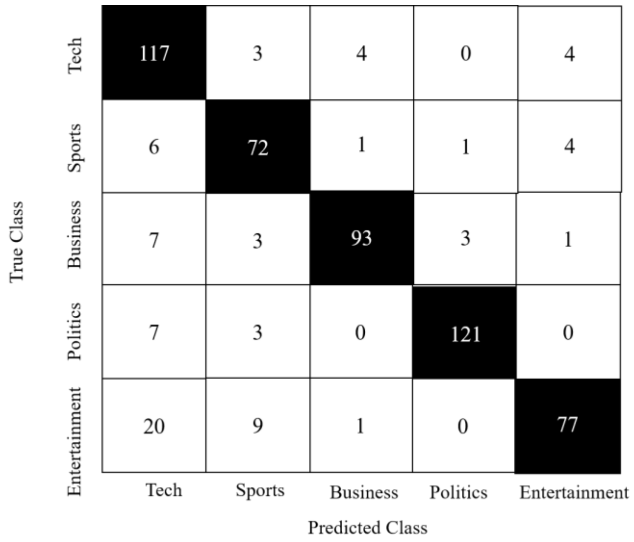
The proposed approach is evaluated on a publicly available dataset comprised of spam and non-spam emails. We trained the BERT architecture by testing out various hyper-parameters and reached the final conclusion on 10 epochs, 32 mini-batch size and Adam optimizer. The different tested hyper-parameters are shown in Table I. The optimal values are also highlighted in the table.

TABLE I. HYPER-PARAMETER OPTIMIZATION ON UCI EMAIL DATASET

Hyper-parameter	**Value/s**
Epochs	4, 8, **10**
Mini-batch	8, 16, **32**
Learning Rate	0.1, 0.01, **0.001**
Optimizer	RMSProp, **Adam**

C. Experiment # 02: Classification of News using BERT

In this section, we discuss the results obtained from the proposed BERT architecture on BBC News dataset. The dataset is composed of five different news categories such as tech, entertainment, sport, politics and business. The proposed method achieved a training accuracy of 89.1% and a testing accuracy of 88.8%. The confusion matrix of the proposed technique is illustrated in Fig. 8. We also evaluated the performance of our proposed framework on precision and recall obtained from the confusion matrix. Training scores of precision and recall are 0.66 and 0.90 respectively as shown in Fig. 9, on the other hand, testing scores for precision and recall are 0.66 and 0.89 respectively as shown in Fig. 10.


True Class
Predicted Class

	Tech	Sports	Business	Politics	Entertainment
Tech	117	3	4	0	4
Sports	6	72	1	1	4
Business	7	3	93	3	1
Politics	7	3	0	121	0
Entertainment	20	9	1	0	77

Fig. 8. Confusion matrix obtained on BBC news dataset.

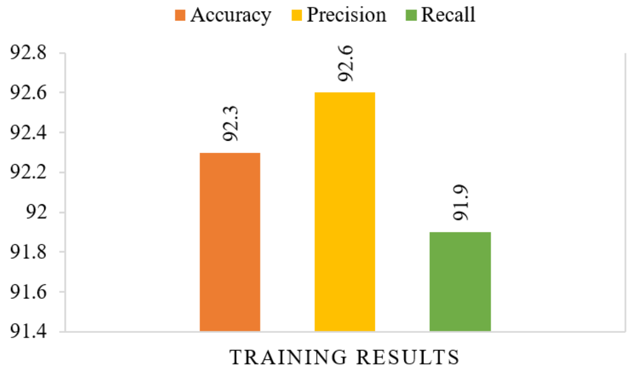


Fig. 9. Training results in terms of accuracy, precision and recall on BBC news database.

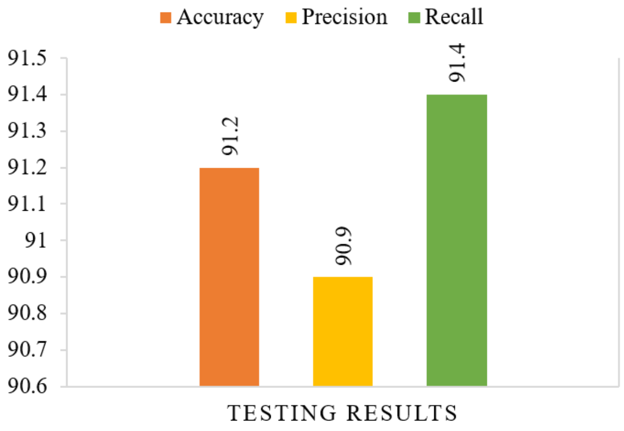


Fig. 10. Testing results on BBC news dataset.

In this experiment, we tested our different hyper-parameter settings to train the BERT architecture before reaching the final conclusion. The final parameters are 10 epochs, 32 mini-batch size and adam optimizer. The different tested hyper-parameters are shown in Table. II Moreover, the optimal values are also highlighted in the table.

TABLE II. HYPER-PARAMETER OPTIMIZATION ON BBC NEWS DATASET

Hyper-parameter	Value/s
Epochs	4, 8, **10**
Mini-batch	8, 16, **32**
Optimizer	RMSProp, **Adam**
Learning Rate	0.1, 0.01, **0.001**

D. Comparison with Existing Systems

One of the very common problems in NLP is text classification that aims to classify text data according to its contents. With the emergence of ML and DL based approaches, the researchers are keen to explore the results of these algorithms to solve this classification problem. However, most of the systems have certain limitations such as poor accuracy, use of single datasets or no data preparation prior to classification. Hence, there is a need to develop a robust and efficient system that can classify text data based on its content accurately.

Hence, in this thesis, we propose a novel and robust text classification method employing one of the very famous DL architecture known as BERT. The proposed method is trained and evaluated on publically available datasets and achieved the 91% and 89% accuracy on different datasets. Since accuracy as a single metric is not sufficient to assess the performance of a classification system, hence, we also evaluated the performance of our proposed strategy using other evaluation parameters namely precision and recall. The results prove the efficacy and robustness of the proposed technique and our devised framework can be deployed by organizations to classify text data. The comparison of our proposed framework with existing methods is described in Table III.

TABLE III. COMPARISON OF PROPOSED METHOD WITH EXISTING SYSTEMS

Reference	Technique	Result/s
Pappagari et al. [18]	RoBERTa & CNN	ACC= 84.7% & 86%
Briskilal et al. [19]	BERT	ACC= 85%
Jang et al. [8]	MLP	ACC= 71%
Semberecki et al. [20]	LSTM	ACC= 86.2%
Lenc et al. [6]	MLP	F1-Score=0.84
Proposed Method	**BERT**	**ACC=91.4%**

V. CONCLUSION AND FUTURE WORK

With the increase in data volumes, automatic text classification has become a necessity for organizations and businesses. The automated systems help them improve their performances overtime and save a lot of time and resources compared to manual systems. This has resulted in increased interest of researchers in this domain of NLP. In this thesis, we propose a novel and completely automated text classification technique employing DL frameworks. The proposed framework uses a fine-tuned BERT architecture to classify text data based on its content. The architecture proposed in this study is case sensitive, hence, the text is preprocessed by changing it in small case. Moreover, additional keywords and stop words are also removed because they can result in poor overall performance.

The preprocessed text data is then fed to fine-tuned BERT architecture for classification. The proposed technique is trained and evaluated on publically available text datasets i.e. BBC News Dataset and UCI Email dataset. The proposed technique achieved accuracy of 91.4% on UCI Email database and 89.1% on BBC News Dataset. We also compared the proposed system’s performance with existing techniques. The results prove the efficiency and robustness of our method. Hence, it can be deployed in businesses to reduce the workload of manual text classification that will save time and energy required in the manual procedure. In the future, we would like to explore text data in various other languages and also explore other DL architectures.

ACKNOWLEDGMENTS

This research work was funded by Postgraduate Studies and Scientific Research, Gulf Colleges, Hafr Al Batin, Saudi Arabia. The authors acknowledge technical and financial support from Gulf Colleges, Hafr Al Batin, Saudi Arabia.

REFERENCES

[1] S. A. Sattar, S. Hina, N. Khursheed, A. J. I. J. o. S. Hamid, and Technology, "Urdu documents classification using naïve bayes," vol. 10, p. 29, 2017.

[2] J. Devlin and M.-W. Chang. Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. Available: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

[3] X. She and D. Zhang, "Text classification based on hybrid CNN-LSTM hybrid model," in 2018 11th International Symposium on Computational Intelligence and Design (ISCID), 2018, vol. 2, pp. 185-189: IEEE.

[4] M. Abdul-Mageed, A. Elmadany, and E. M. B. J. a. p. a. Nagoudi, "ARBERT & MARBERT: deep bidirectional transformers for Arabic," 2020.

[5] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, "Rmdl: Random multimodel deep learning for classification," in Proceedings of the 2nd international conference on information system and data mining, 2018, pp. 19-28.

[6] L. Lenc and P. Král, "Deep neural networks for Czech multi-label document classification," in International Conference on Intelligent Text Processing and Computational Linguistics, 2016, pp. 460-471: Springer.

[7] W. Antoun, F. Baly, and H. J. a. p. a. Hajj, "Arabert: Transformer-based model for arabic language understanding," 2020.

[8] B. Jang, M. Kim, G. Harerimana, S.-u. Kang, and J. W. J. A. S. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," vol. 10, no. 17, p. 5841, 2020.

[9] J. Cai, J. Li, W. Li, and J. Wang, "Deeplearning model used in text classification," in 2018 15th international computer conference on wavelet active media technology and information processing (ICCWAMTIP), 2018, pp. 123-126: IEEE.

[10] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 377-384.

[11] C. Kaul, S. Manandhar, and N. Pears, "Focusnet: An attention-based fully convolutional network for medical image segmentation," in 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), 2019, pp. 455-458: IEEE.

[12] Elvis. Deep Learning for NLP: An Overview of Recent Trends. Available: https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d.

[13] B. Lutkevich. BERT language model. Available: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model.

[14] R. Horev. (2018). BERT Explained: State of the art language model for NLP. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270?gi=928a1e3b0ac9.

[15] I. Tenney, D. Das, and E. J. a. p. a. Pavlick, "BERT rediscovers the classical NLP pipeline," 2019.

[16] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in data democracy: Elsevier, 2020, pp. 83-106.

[17] A. J. I. J. o. R. S. Hay, "The derivation of global estimates from a confusion matrix," vol. 9, no. 8, pp. 1395-1398, 1988.

[18] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, "Hierarchical transformers for long document classification," in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 838-844: IEEE.

[19] J. Briskilal, C. J. I. P. Subalalitha, and Management, "An ensemble model for classifying idioms and literal texts using BERT and RoBERTa," vol. 59, no. 1, p. 102756, 2022.

[20] P. Semberecki and H. Maciejewski, "Deep learning methods for subject text classification of articles," in 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), 2017, pp. 357-360: IEEE.

Experimental Evaluation of Genetic Algorithms to Solve the DNA Assembly Optimization Problem

Hachemi Bennaceur[1], Meznah Almutairy[2], Nora Alqhtani[3]
Faculty of Computer and Information Sciences-Computer Science Department,
Al Imam Mohammad ibn Saud Islamic University (IMSIU),
Riyadh, Saudi Arabia[1, 2, 3]

***Abstract*—This paper aims to highlight the motivations for investigating genetic algorithms (GAs) to solve the DNA Fragment Assembly (DNAFA) problem. DNAFA problem is an optimization problem that attempts to reconstruct an original DNA sequence by finding the shortest DNA sequence from a given set of fragments. This paper is a continuation of our previous research paper in which the existence of a polynomial-time reduction of DNAFA into the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP) was discussed. Taking advantage of this reduction, this work conceptually designed a genetic algorithm (GA) platform to solve the DNAFA problem. This platform offers several ingredients enabling us to create several variants of GA solvers for the DNAFA optimization problems. The main contribution of this paper is the designing of an efficient GA variant by carefully integrating different GAs operators of the platform. For that, this work individually studied the effects of different GAs operators on the performance of solving the DNAFA problem. This study has the advantage of benefiting from prior knowledge of the performance of these operators in the contexts of the TSP and QAP problems. The best designed GA variant shows a significant improvement in accuracy (overlap score) reaching more than 172% of what is reported in the literature.**

***Keywords*—Genetic algorithms; traveling salesman problem; quadratic assignment problem; DNA fragments assembly problem**

I. Introduction

The DNA fragment assembly problem is the process of reconstructing an original DNA sequence from a given set of DNA fragments. This is achieved by ordering and aligning these DNA fragments such that the resulting DNA sequence is as short as possible. It is a complex combinatorial optimization problem belonging to the class of NP-hard problems, where there is a need to find the right order of the DNA fragments to assemble them. Several metaheuristic techniques have been developed to solve this problem [1], [2], [3]. This paper exploit the genetic algorithm platform (GAP) developed in the former preliminary paper [4], in which the existence of a polynomial-time reduction of DNAFA into the Traveling Salesman Problem and the Quadratic Assignment Problem was discussed. Then, conceptually designed a GA platform for solving the DNAFA problem, inspired by the existing efficient GAs in the literature for solving the TSP and QAP problems. This platform gathers and offers several GA operators designed to solve hard optimization problems such as TSP, QAP, and DNAFA. The GAP enables the researchers to easily design an adequate variant GA algorithm for hard optimization problems in particular. This work implementing and experimenting on some GA variants judiciously built from the platform (GAP) aims to identify the best variant that efficiently deals with the DNAFA problem. Using this platform, this work is able to individually study the effects of genetic algorithm components on selected metrics, which were presented in terms of time and overlap score. This work focused on examining and discussing the effects of population size, population generation methods, selection types, and crossover types and figure out which component has the most impact on GA performance. Some of these GA components have never been tested in the context of the DNAFA problem, such as SCX crossover, which is worth to be investigated experimentally. Other components have been tested before, but when retest was done on them, a different result was found, such as greedy as a population initialization method. Because of SCX effectiveness in TSP and QAP, we believe the SCX crossover is a smart crossover that will outperform other crossovers. As a result of these comprehensive experiments, this work identifies the best-designed GA variant that outperforms the existing GA algorithms in solving the DNAFA problem. This GA variant features the use of 200 individuals for the population size, along with the greedy method for initializing the population, tournament selection, and SCX crossover. This GA variant showed a significant increase in overlap score compared to what is reported in the literature. The results showed that the SCX crossover was the best crossover among the studied crossovers and gave good results. Furthermore, the results showed that the greedy method is a very powerful method that improved the algorithm's performance by 37%, demonstrating that the population generation method has the greatest impact on improving the results than the other GA components. The experimental results demonstrated the efficiency of the designed approach, as it got a better result for the overlap score ranging from 56.16% to 172.74% than the previous recorded results for most data sets. This work demonstrate experimentally that the best designed GA variant outperforms existing GA algorithms in solving the DNAFA problem for some data sets.

A. The DNAFA Problem

The DNAFA problem is defined as follows: Given a set of fragments, f1, f2,…, fn, drawn from a finite alphabet Σ = $\{A,C,G,T\}$, the goal is to find the shortest superstring that contain all the input fragments $F' =< f'_1, f'_2, \dots, f'_n >$ that maximizes the number of overlaps between every pair of two consecutive fragments and thus minimizes the length of F'.

$$MAX_{<f'_1,f'_2,\dots,f'_n>} \sum overlapping\ (f'_i f'_{i+1}) \qquad (1)$$

$$where\ 1 \leq i \leq n-1.$$

B. Assembly Process

To understand the assembly process, we've defined some key terms.

- Fragment: A short sequence of DNA bases. It is also called read.
- Coverage: The number of fragments at a specific position in the DNA.
- Prefix: A substring from the first characters of a fragment.
- Suffix: A substring from the last characters of a fragment.
- Overlap: Common sequence between the suffix of one fragment and the prefix of another.
- Layout: An arrangement of the collection of fragments based on their overlapping order.
- Contig: Contiguous overlapped fragments without gaps.
- Scaffold: The overlapped contigs, which may contain gaps.
- Consensus: Reconstruction of the complete sequence

In the assembly process, the input for the DNA fragment assembly is a set of fragments. The traditional assembly approach works in the following order: overlap, layout, and consensus [5].

- *Overlap stage:* Finding the overlapping fragments and computing their similarity score (overlap score). This means finding the longest match between the suffix of one fragment and the prefix of another.
- *Layout stage:* Finding the order of fragments based on the computed overlap score.
- *Consensus stage*: Reconstructing the complete sequence from the layout.

This paper is organized as follows: the second section discusses the related works, then the proposed design is discussed in detail in the third section. In the fourth section, the experiments and the method of conducting the investigations are detailed, and then the results are listed in the fifth section. The sixth section discusses these results by comparing them with previous works, and finally, the paper is concluded in the seventh section.

II. Related Works

This section presents the previous related works organized into two subsections: the first subsection summarizes the works solving the DNAFA with GA, and the second subsection introduces the works solving the TSP and QAP with GA.

A. Genetic Algorithm for DNAFA Problem

The basic genetic algorithm schema contains various concepts such as population encoding, population initialization, fitness function, selection, crossover, and mutation. Each concept has its own importance in the algorithm. By studying previous works, it can be noted that each concept can be done in a different way. In more detail, the population can be encoded in different ways in the GA, one such way is through segmented permutation [6], identity permutation [7]. Random generation, as in [8], the greedy approach, and the 2-opt heuristics, as proposed by Minetti et al. [8] and [9], are common strategies for generating the initial populations.

For the fitness function, the most commonly used fitness function is to maximize the overlap score, where the smith-waterman algorithm is used to calculate the overlap between the fragments [7]. The smith water man algorithm takes a lot of time but, even though it is the most precise algorithm for identifying similarity regions between fragments. Overlap score is considered the best measure for measuring the quality of the solution. It was used in most of the previous works [7], [8], and [9].

The crossover operator is the main operator of GA, as it plays a crucial role in efficiently exploring the search space of the optimization problem. The parents' characteristics are mainly inherited by crossover operators. Among the crossovers that were used in solving DNAFA, there was the order-based crossover (OX) as in [5], [10], [6], the edge-recombination (ER) [5], and the partially mapped crossover (PMX) [7], [9]. For the mutation, inversion mutation operators [10], and swap mutation [7], [9]were used for the DNAFA.

GA can be combined with other metaheuristics to achieve good results. For this purpose, Minetti et al. [10] designed a hybrid method named SAX that combined the GA with a simulated annealing metaheuristic. Another work, by Hughes et al. [7] combines different variations of GA in different ways. Also, the authors in [5] applied multiple algorithms, such as simulated annealing and scatter search with the GA. Another recent work is provided by Uzma and Halim [9] they combine GA and Power Aware Local Search (PALS).

The studies of Bucur [6], [11] focus on minimizing the total length of the scaffold (summing the length of the overlapped contigs). Unlike previous works, Bucur used simulated data sets where the fragments were of uniform length, they were able to measure the accuracy since they had the reference genome. However, as they mentioned, the main disadvantage of their method is its increased time complexity.

B. Genetic Algorithm for TSP and QAP

This section reviews GA algorithms designed to solve the TSP and QAP problems. Different types of encoding were used for the optimization problems TSP and QAP. Most works of the wide literatures used the identity permutation such as for TSP in [12], [13] and QAP in [14]. Another advanced types of population encoding were used for TSP such as value encoding [15], and real number encoding [16]. The common strategies of generating the initial populations are the random generation as investigated for TSP in [16] and the greedy method as in [17]. Recently, more advanced strategies have been developed, the author in [13] proposed Multi-Agent Reinforcement Learning (MARL) for solving TSP problems, and [14] implemented the sequential sampling method for solving QAP problems.

For the selection, the roulette wheel is the common selection operator used for optimization problems [15], [13], [18], [16], the tournament selection was implemented for TSP [17], and the stochastic remainder selection was used for QAP [19]. More recently, in [16], a greedy method was designed as a selection operator for TSP.

Several advanced crossover operators have been designed for solving TSP as well as QAP using GA algorithms. The Sequential Constructive Crossover (SCX) is an intelligent crossover designed by Ahmed [12] to solve the TSP. Recently, a modified version of sequential constructive crossover, named greedy SCX (GSCX), was proposed for solving TSP [20]. The reverse greedy sequential constructive crossover (RGSCX) and the comprehensive sequential constructive crossover (CSCX) are two new crossover operators that enhance SCX for solving TSP [21]. Other types of advanced crossover operators were designed in [18] to solve the QAP, relying on the idea of a frequency model. Three crossover operators were introduced for enhancing GA, namely, the Highest Frequency Crossover (HFX), the Greedy HFX (GHFX), and the Highest Frequency Minimum Cost Crossover (HFMCX).

Various types of mutations have been investigated for the TSP and QAP problems, including the exchange mutation [13], [16], [17], and the reciprocal exchange mutation [12]. More advanced mutation operators have been designed for the TSP and QAP problems, such as the interchange mutation in [15], and the inversion mutation in [17]. In [22], the adaptive and combined mutation operators were proposed for solving QAP.

C. *Other Metaheuristics Algorithms for DNA Fragments Assembly*

Particle swarm optimization (PSO) was reported in the literature for the DNA fragment assembly problem. Verma and Kumar [2] used the PSO with the smallest position value (SPV) rule. The PSO can be enhanced when combined with other algorithms, such as in Huang et al. [23], who proposed a hybrid particle swarm optimization algorithm (HPSO). The algorithm was divided into two parts: (1) Tabu search combined with PSO to improve solution quality and (2) simulated annealing combined with variable neighborhood local search (VNS). Additionally, the parallel approach can reduce the computation time, so, Mallén-Fullerton and Fernández-Anaya [2] presented a parallel heuristic based on the PSO and the differential evolution (DE), which is similar to GA, but DE relies on mutation operation, while GA relies on crossover operation to assemble better solutions. Mallén-Fullerton and Fernández-Anaya used a variation of the TSP (the Lin-Kernighan algorithm [24]) with some modifications to be applied for DNA fragment assembly. Another study is that of Huang et al [25], who presented a memetic PSO algorithm with a variable neighborhood search (VNS) approach as well as TS and SA, each of these algorithms is used in different ways and in different combinations. Indumathy and Maheswari [26] used a variant of the standard PSO called the constriction factor PSO (CPSO). Another proposed metaheuristic algorithm for solving the DNA fragment assembly problem is the problem aware local search (PALS) [27]. The main drawback of PALS is its quick convergence to local optima but combining it with other algorithms can overcome this drawback. Minetti et al. [28] used PALS by combining it with SA, this suggested method shows improved performance on the largest data sets when compared with SA and PALS separately. Another algorithm founded for the DNA fragment assembly is the bee colony, Firoz et al. [29] presented the artificial bee colony (ABC) algorithm and the queen bee evolution based on the genetic algorithm (QEGA). Majid al-Rifaie [30] investigated a new algorithm, stochastic diffusion search (SDS), which follows a different strategy for calculating the overlaps, picking a model from given fragments and trying to find the same model in the rest of the fragments. Among the fragments containing the model, the one with the highest similarity is picked, assembled, and then removed from the search space.

The previous paper [4] showed that the DNAFA optimization problem is a special case of two well-known optimization problems: the traveling salesman problem and the quadratic assignment problem. Particularly, that paper theoretically demonstrated that all three optimization problems have a similar topological structure and that they need to explore a search space of solutions with the same complexity to find an optimal solution. For this reason, the GA platform designed to solve the DNAFA problem is inspired by the efficient GA approaches developed for the famous combinatorial optimization problems, TSP and QAP. The GA platform gathers several advanced GA operators and tools that have demonstrated their effectiveness in the context of TSP and QAP.

Table I illustrates the GA parameters' settings from the literature for DNA, TSP, and QAP.

III. The GA Platform

The GA platform consists of the best and most advanced GA tools for the DNAFA problem (shown in Fig. 1.). One could build several variants of the GA to solve it by judiciously integrating the ingredients of this platform in different ways.

A. *The GA Operations of the Designed Platform*

This section describe the different GA operations involved in the platform that suggested earlier [4]. This paper will study all these operations, test them experimentally, and try different versions by combining different tools from the platform to create the best version that will be compared with other algorithms from the literature.

1) Encoding: For the encoding, the work will use the integer encoding, where the fragments encode as numbers, such that fragment one encodes as "1", fragment two encodes as "2" and so on.

TABLE I. GA PARAMETERS SETTING FROM THE LITERATURE FOR DNA, TSP AND QAP

GA Design and Experimental Settings	DNA_FA	TSP	QAP
Individual encoding	Integer numbers (an ordered sequence of integer numbers, each of which represents a fragment number).	Integer numbers (an ordered sequence of integer numbers, each of which represents a city to be visited)	Integer numbers (an ordered sequence of integer numbers, each of which represents an assignment of a task to a resource).
Population initialization	Random, greedy, 2-opt heuristics	Random, greedy, MARL (Multiagent Reinforcement Learning) [13].	Sequential sampling, random
Population size	Varies from 11 to 2500 Individuals.	Varies from 20 to 200 individuals.	Varies from 30 to 200 individuals.
Selection	Tournament.	Roulette wheel, tournament, greedy [31]	Roulette wheel, stochastic reminder selection.
Crossover	OX, ER, PMX, one-point order. CX	SCX, ERX, GNX, PMX, smart multi point crossover, order insert crossover.	SCX, OPX, SPX, HFX, GHFX, HFMCX, MPX.
Mutation	Inversion mutation, swap mutation	Reciprocal mutation, exchange mutation, interchange mutation, inversion mutation	Reciprocal exchange mutation, combined mutation, adaptive mutation, swap mutation [12], [14].
Crossover probability	Varies from (60% to 100%)	Varies from (90% to 100%)	100%
Mutation probability	2%	Varies from (1% to 20%)	Varies from (5% to 15%)
Stopping condition	No improvement for number of iterations.	Optimal rout, number of generations.	Number of generations, CPU time.
Number of runs	From 5 runs to 30 runs	From 10 runs to 30 runs.	20 runs.
Number of generations	Varying from (1000 to 512 K) generations	Varying from (20 to 10,000) generations.	Varying from (5000 to 10k) generations.

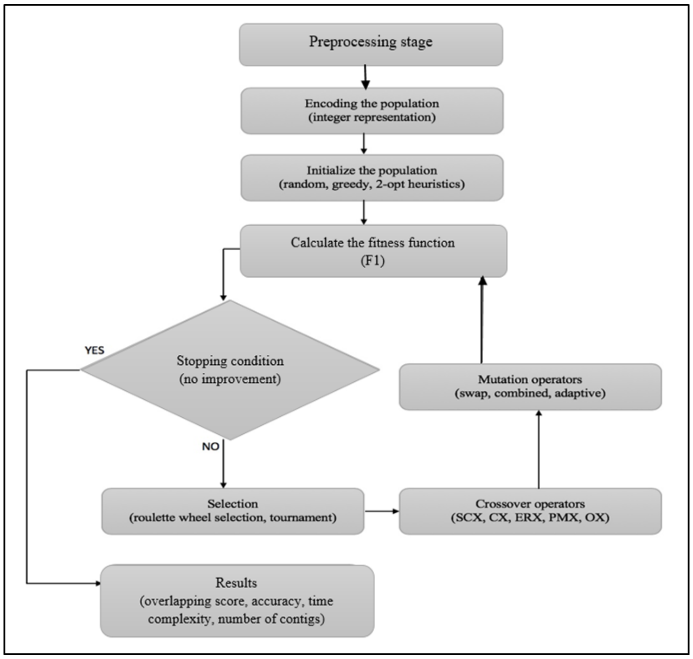

Preprocessing stage
Encoding the population
(integer representation)
Initialize the population
(random, greedy, 2-opt heuristics)
Calculate the fitness function
(F1)
YES
Stopping condition
(no improvement)
NO
Mutation operators
(swap, combined, adaptive)
Selection
(roulette wheel selection, tournament)
Crossover operators
(SCX, CX, ERX, PMX, OX)
Results
(overlapping score, accuracy, time
complexity, number of contigs)

Fig. 1. GA platform design for the DNA fragments assembly problem.

2) Initial Population: Initial population includes the population size and the population generation method. For the population size, the work selects two sizes (200 and 500 individuals) and discusses how much time is saved if the population is small and how accurate it is.

For the population generation method, the GA platform design includes the random, greedy, and 2-opt heuristic strategies that have previously yielded good performances as shown in [7], [31], and [9]. Since the primary results showed that the greedy initialization method gave the best solutions, this paper displays its results. However, because the greedy method searches and generates populations intelligently, further experiments will investigate whether the greedy can find the solution from the beginning without relying on the rest of the GA operator.

3) Fitness Functions: As the fitness function is repeatedly applied to each individual of each generation, it should be relatively easy to compute and should also accurately evaluate the quality of each individual [12]. A simple fitness function aims to maximize the overlap score by summing the overlap for each of the adjacent fragment pairs, as expressed by the expression (2) in [9].

$$F = \sum_{i=1}^{n-1} w[i, i+1] \qquad (2)$$

where w $[i,$ i + 1] is the overlap score between fragment i and fragment i + 1. F is simple in complexity since it takes O(n).

To measure the solution quality, the work will use the following formula, which is often used in TSP and QAP problems to measure the solution quality.

Gap = ((ASV-BSV)/BSV) *100

Where ASV refers to the average solution value (average overlap) and BSV refers to the best-known solution value reported in the literature.

4) Selection operators: As roulette wheel selection is widely used and consumes the least amount of time and tournament selection can maintain diversity by giving an equal chance to all the individuals to compete [32], the roulette wheel and the tournament selections are selected to be added to the platform.

5) Crossover operators: Several crossover operators, including SCX, OX, CX, PMX, and ERX, have been chosen for inclusion in the platform. Special attention should be paid to SCX, as it is a smart crossover and was one of the best operators for the TSP and QAP problems and is expected to have the same performance in the DNA_FA context. Moreover, SCX has never been used to solve the DNA assembly problem before; therefore, this paper will present the results related to this crossover.

6) Mutation operator: The swap mutation operator and its variants were widely used for DNA_FA, TSP, and QAP [9] and [17]. Combined and adaptive mutations were designed for the QAP problem [22]. These three mutation types are included in the platform.

7) Stopping condition: The GA platform will stop if the solution is not improved at all during a certain number of iterations or a time limit is reached.

IV. Experiments

This section describes the data sets, the experimental setting, and the variable's values in this study.

A. Data Sets

The GA platform will be assessed on data sets produced by next-generation sequencing, the data sets are obtained from the National Center for Biotechnology Information (NCBI)[1]. These data sets are the same benchmarks used in the previous works mentioned in the related works section. This work used 17 data sets with a varying number of fragments, from 25 fragments to 352 fragments. The mean length of the fragment varies between 286 and 512 pb, the description of these data sets is given in Table II.

B. Experimental Setting

The designed algorithm has been implemented in C++ on a Windows 10 computer with a 2 GHz CPU and 16 GB of RAM. The work maintained the parameter values used in TSP and QAP that led to the best results. Based on Table I, this work chose the values for the GA operators, which are described in detail in Table III. The work applied to each data set 60 experiments using different GA parameters and operators. In detail, it applied two types of population size, three types of initialization, two types of selection, and five types of crossover (2*3*2*5 = 60). Since the GA is a stochastic process, each experiment was run 30 times to ensure the satiability of the given results (30*60=1800 experiments). Since there were 17 datasets, the total number of experiments reached more than 30 thousand (17 * 1800=30,600 experiments)

TABLE II. Benchmarks' Datasets, where Total Data Size is Number of Fragments * Mean Fragment Length

Benchmark	Mean fragment length	Number of fragments	Total data size
x60189 4	395	39	15405
x60189 5	286	48	13728
x60189 6	343	66	22638
x60189 7	387	68	26316
m15421 5	398	127	50546
m15421 6	350	173	60550
m15421 7	383	177	67791
j02459 7	405	352	142560
f25_305	307	25	7675
f25_400	400	25	10000
f25_500	500	27	13500
f50_315	315	50	15750
f50_412	412	50	20600
f50_498	498	50	24900
f100_307	307	100	30700
f100_415	415	100	41500
f100_512	512	100	51200

TABLE III. Parameters' and Operators' Values Used in the Experiments

Parameter	Value
Population size	200, 500 individuals.
Initialize type	Randomly, 2-opt heuristics, and greedy
Number of runs	30 runs.
Stopping condition	No improvement for 300 consecutive generations or running time < 5 second.
Selection type	Tournament with size 5, roulette wheel
crossover	SCX, OX, CX, PMX, ERX.
mutation	Randomly pick one of (Swap, adaptive, and combined mutation)
Mutation probability	0.001
Crossover probability	1.0
Total experiments	30,600

[1] The National Center for Biotechnology Information (NCBI) is part of the United States National Library of Medicine (NLM), a branch of the National Institutes of Health (NIH). The NCBI houses a series of databases relevant to biotechnology and biomedicine and is an important resource for bioinformatics tools and services. Major databases include GenBank for DNA sequences. https://www.ncbi.nlm.nih.gov/guide/

C. *Evaluation Metrics*

The performance of the GA algorithm will be measured in terms of the following:

- Overlapping scores: should be high. The overlap score measured by calculating the length of the overlap between each fragment and all the existing fragments. The overlap scores were computed using the Smith-Waterman algorithm. Two forms of overlap scores were reported in the results: the best overlap scores out of 30 runs, and the average overlap score for 30 runs.
- Computational complexity (time complexity): should be minimized. The time for the complete assembly process was divided into two stages: the time for calculating the overlap score in the preprocessing stage, and the time for the GA to find the best solution. This paper only showed the time of the GA because this work studying the change in the performance of the GA and also because the time for SW is constant for each dataset regardless of which GA variant is studying.

D. *Aspects of Investigations*

This work study the effect of some algorithm components on two metrics: the GA running time and the overlap score. It will discuss the effect of population generation, including the population size and the population generation method, the effect of the crossover types, as well as the effect of the selection types. According to the comprehensive experiments, below are the major interesting investigations aspects.

- Crossover types on overlap score when varying the population generation method.
- Population size on overlap score and GA running time.
- Selection types on overlap score and GA running time when varying the population generation method.

V. Results

This section presents and discusses the results obtained from the experiments conducted in this study. The results in this section are organized in subsections. Each subsection reports the results of a specific investigation, as mentioned earlier. In each subsection, the results will be illustrated with tables or pictures and discussed, in addition to summarizing the findings at the end of each subsection.

A. *The Effect of Crossover Type on Overlap Score when Varying the Population Generation Method*

This section studies the effect of the crossover types on the overlap score when varying the population generation methods while the other GA operators remain constant. For simplicity, the type of selection operation is the tournament, and the population size is 200.

Each of the figures below represents the effect of a specific population generation method on the best overlap score.

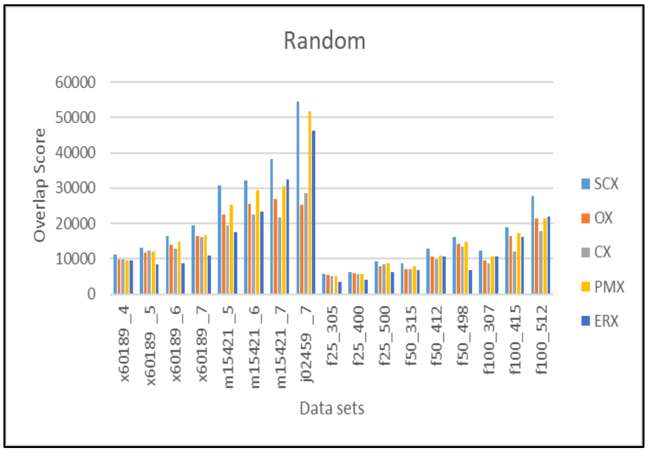


Fig. 2. The effect of crossover types on overlap score when generation population method is random.

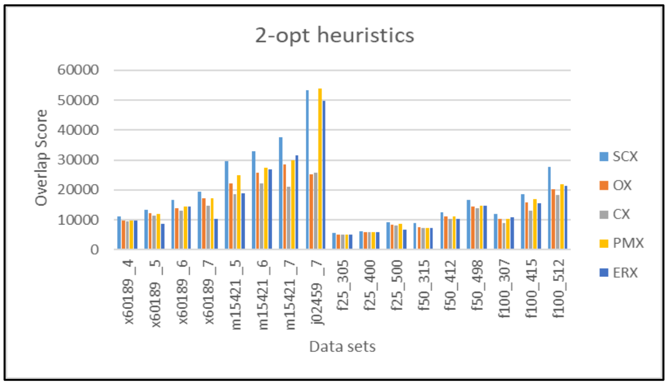


Fig. 3. The effect of crossover types on overlap score when generation population method is 2-opt heuristics.

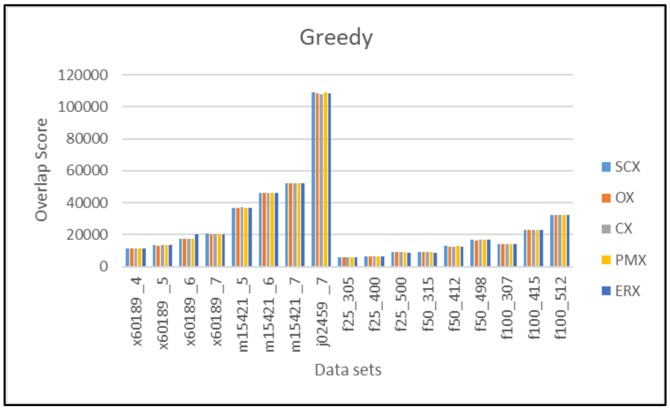


Fig. 4. The effect of crossover types on overlap score when generation population method is greedy.

From Fig. 2, Fig. 3, and Fig. 4, it can be seen that the greedy initialization type is the one that gives the highest overlap score for all the data sets. Moreover, the 2-opt heuristics and random method gave different results but clearly showed that SCX is the best crossover in the majority of cases. Clearly, the SCX has the best accuracy regardless of the population generation methods. SCX is less sensitive to the type of initialization, whatever the type of initialization, it gives good results in every case. Also, the greedy approach improves the performance of all crossover operators. It seems only when population generation method is greedy, it is not clear if SCX is still the best. This reveals the impact of the population generation method and how creating the population in a smart way from the beginning has been a strong factor in improving the results and reducing the differences between the types of crossovers. In more detail, the greedy improved the SCX overlap results by 11.7% compared to random and 2_opt heuristics, also improved the OX and CX results by 25.9% and 31.01%, respectively. And improve the PMX and ERX by 22.01% and 37.7%, respectively. This demonstrates that the SCX was the least sensitive crossover to the population generation method among the crossover types and maintained the highest overlap score. Thus, further investigation is done by checking the time cost for the greedy method using small and large population sizes.

The results of this study are summarized as follows:

- SCX is not sensitive to the type of initialization, whatever the type of initialization, it gives good results in every case.
- OX, CX, PMX, and ERX are sensitive to the type of generating population, as they perform better with the greedy initialization than with the random and the 2-opt heuristics. Because the greedy is good at generating a good initial population.
- The population generation method has a strong impact on improving the results.

B. *The Effect of the Population Size on Overlap Score and GA Running Time*

This subsection investigates the effect of population size on the overlap score, including the best overlap score and average overlap score. In addition, the effect of population size on the GA's running time.

1) The effect of population size on overlap score: Recall that this paper only report for the SCX; Table IV shows the effect when the initialization method is greedy, the crossover type is SCX, and the selection type is tournament. The "gap" column represents the gap on overlap which calculated by the formula in Section III and the "absolute difference" represents the pure difference between 500 and 200 individuals.

In this investigation, when the crossover is SCX, and the generation method is greedy, and the selection type is tournament, the results show that when the initial population is 60% less, the GA still gives high overlap score in all datasets with an average difference of 0.14%. This is important since decreasing the initial population size decreases the computation significantly.

Moreover, for datasets (f*) this work show a significant increase in accuracy compared to what is reported in the literature. The increase reaches 172.57%, with the 200-population size, and 171.9% with the 500-population size. Moreover, when computing the difference between the best overlap score and average overlap score, there was not a significant difference in performance, where the best overlap score is only 0.44% and 0.42% better than the average overlap score on 200 and 500, respectively. Thus, the paper only reports the best overlap score in the rest of the paper. With regard to the gap, the table shows that our results are better in 9 data sets out of 17. When the absolute difference is negative, that means the 200 size is better than the 500. This was clear for eight data sets, and their performance was equal in four data sets, this makes the smaller size more suitable.

The results of this study are summarized as follows:

- Increase the size of the population increase the computational time, however it may give chance to have good results for the big data sets.
- The population sizes of 200 and 500 individuals do not have a noticeable difference in the quality of the solution; therefore, it is preferable to take the smaller size.

TABLE IV. THE EFFECT OF POPULATION SIZE ON OVERLAP SCORE FOR SCX WHEN THE GENERATION METHOD IS GREEDY AND SELECTION TYPE IS TOURNAMENT

Data set	**BSV**	**200 individuals-greedy**			**500 individuals-greedy**			**Gap on the overlap (%)**		**Absolute difference (500, 200)**
		Best overlap	**Average overlap**	**Standard deviation**	**Best overlap**	**Average overlap**	**Standard deviation**	**200-individuals**	**500-individuals**	
x60189 4	11478	**11298**	**11203.8**	46.99	11272	11195.9	**41.95**	-2.39	-2.46	-26
x60189 5	14161	**13594**	**13498.4**	67.11	13540	13478.4	**12.38**	-4.68	-4.82	-54
x60189 6	18301	**17401**	**17318.9**	41.02	17304	17273.8	**6.96**	-5.37	-5.61	-97
x60189 7	21271	**20546**	**20433.8**	30.40	20527	20431.6	**30.03**	-3.94	-3.95	-19
m15421 5	38746	36972	**36916.6**	20.08	**36975**	36908	**12.55**	-4.72	-4.74	3
m15421 6	48052	**46304**	46226.2	32.61	46240	**46233.7**	**18.38**	-3.80	-3.78	-64
m15421 7	55171	52069	51977.7	74.56	**52077**	**52005.6**	**37.98**	-5.79	-5.74	8
j02459 7	116700	109043	108855	**58.85**	**109056**	**108876**	76.40	-6.72	-6.70	13
f25_305	2271	**5594**	**5594**	**0**	**5594**	**5594**	**0**	146.32	146.32	0
f25_400	3139	**6307**	**6307**	**0**	**6307**	**6307**	**0**	100.92	100.92	0
f25_500	5777	**9170**	**9021.5**	71.45	**9170**	8974.13	**45.71**	56.16	55.34	0
f50_315	4013	**9076**	9072.87	16.87	**9076**	**9076**	**0**	126.09	126.16	0
f50_412	5835	12990	12896.4	**29.14**	**13095**	**12915.3**	48.60	121.02	121.34	105
f50_498	9050	**17070**	**16935.8**	54.22	17012	16910	**27.48**	87.14	86.85	-58
f100_307	7035	**14319**	**14265.3**	16.78	14282	14260.7	**3.96**	102.78	102.71	-37
f100_415	9202	**23008**	**22896.5**	45.55	22993	22894.2	**27.49**	148.82	148.80	-15
f100_512	11881	**32384**	32285.2	41.36	**32340**	**32305**	**26.11**	172.57	171.90	1

2) *The effect of population size on GA running time:* This section studies the effect of population size on the GA running time. The following tables reveal this effect when the type of crossover is SCX, the type of initialization is greedy, and the selection type is tournament. In addition, Table VI compare the use of complex SCX crossover operators with small population size to the use of simple PMX, ERX, OX and CX crossover operators with large population sizes. The 200 and 500 refer to small and large population sizes, respectively.

- The "GA time" column represents the time for the whole GA to find the result,
- The "greedy time" column represents the time for initializing the population with the greedy method.
- The "overlap after greedy" column represents the overlap score after creating the population.
- The "overlap after GA" column represents the overlap score when the algorithm is done.
- The "Increase in overlap "column represents the percentage increase in the overlap score between the overlap score after greedy and the overlap score at the end of the GA.
- The "Absolute difference in time" column shows the difference in time between 500 and 200 individuals for the GA time.

As indicated earlier, the question posed for discussion is whether the greedy might override the algorithm's performance, is the solution comes from the greedy or the GA finds the solution, is the time spent in creating the population or in finding the solution. For this matter, the following table illustrate the overlap score after creating the population with the greedy method, as well as the overlap score when the algorithm is done. In addition to the time taken to create the population and the time taken by the algorithm to find the solution, the least time and the best solution for every data set are marked in bold.

TABLE V. The Effect of Population Size on GA Running Time (in Sec.). For SCX when the Generation Method is Greedy and Selection Type is Tournament

Data sets	**200 individual-greedy**					**500 individual-greedy**					**Absolute difference in time (500-200)**
	GA time	**Greedy initialization time**	**Overlap after greedy**	**Overlap after GA**	**Increase in overlap (%)**	**GA time**	**Greedy initialization time**	**Overlap after greedy**	**Overlap after GA**	**Increase in overlap (%)**	
x60189 4	1.93	0.36	10765	**11298**	4.72%	**1.07**	0.81	10865	11252	3.44%	-0.86
x60189 5	1.6	0.59	12250	**13594**	9.89%	**1.83**	1.4	12250	13493	9.21%	0.23
x60189 6	**2.5**	0.99	16272	**17401**	6.49%	**2.5**	2.2	16275	17295	5.90%	0
x60189 7	**1.9**	1.1	19419	20546	5.49%	3.3	2.6	19529	**20641**	5.39%	1.4
m15421 5	**5.1**	4.1	36606	**36972**	0.99%	6.9	6.4	36616	36951	0.91%	1.8
m15421 6	**6.4**	4.7	46154	**46304**	0.32%	13.1	12.3	46154	46262	0.23%	6.7
m15421 7	**5.4**	3.6	51708	**52069**	0.69%	13.7	12.5	51738	52012	0.53%	8.3
j02459 7	**27.4**	23.1	107398	**109094**	1.55%	60.2	56.7	107504	109051	1.42%	32.8
f25_305	**0.2**	0.1	5594	**5594**	0.00%	0.43	0.36	**5594**	**5594**	0.00%	0.23
f25_400	**0.23**	0.17	6107	**6307**	3.17%	0.5	0.4	**6307**	**6307**	0.00%	0.27
f25_500	1.6	0.2	8550	**9170**	6.76%	**1.2**	0.5	8640	**9170**	5.78%	-0.4
f50_315	**0.7**	0.6	8576	**9076**	5.51%	1.6	1.4	8576	8982	4.52%	0.9
f50_412	**1.03**	0.7	12486	12990	3.88%	2.3	1.8	12492	**13011**	3.99%	1.27
f50_498	**1.5**	0.8	16802	**17051**	1.46%	2.2	1.9	16802	17014	1.25%	0.5
f100_307	**2.6**	2	14100	**14319**	1.53%	5.5	5.3	14103	**14260**	1.10%	2.9
f100_415	**3.6**	1.9	22154	**23008**	3.71%	6.2	4.7	22154	**22993**	3.65%	2.6
f100_512	**4.8**	2.9	31187	**32384**	3.56%	8.6	7.6	31189	32339	3.56%	3.8

The table above shows that a small population size takes less time in the majority of the data sets and gives more opportunity for the algorithm to improve the solution. However, the larger population size takes more time to generate but less time to find the solution. In the case of small data sets, most of the time is taken to create population, while the time taken to find the solution is very small. These results showed that when using a larger population size with the greedy method, the improvement in the solution is small and may be nonexistent in the case of small data sets such as (f25_305, f25_400). Small population sizes are better suited to the greedy method because they allow the algorithm to improve the solution while also taking less time. When the datasets (x60189_4 and f25_500) are excluded, the results in Table V for total time show that there is a 49.21% reduction in time when using a 40% smaller population. Moreover, this table showed us that the greedy method contributed 95% to improving the solution, and the GA improved the solution by 3.51% for the 200-population size and 2.99% for the 500 size. This supports the previous investigation, that SCX with a smaller population size is better.

But it may come to mind that if we choose the larger population size with a simple crossover, could it give an overlap higher than the SCX with the small population size in a reasonable time? So, recalling what previously raised for discussion the following table compare SCX with the smaller population size, with other types of crossovers with a larger population size, for time and overlap score.

The results in Table VI show in 14 out of 17 data sets, using SCX with 40% less population size leads to better results than other crossovers with larger size. In addition to having a 40% smaller population size but comparable accuracy, SCX is also significantly faster than the other crossovers. The results show that SCX is 26.92% faster than PMX in all data sets, except for “x60189 _6” and “f25_500” datasets, and in some datasets, it is 59.46% faster (as in f50_498). Also, SCX is 38.38% faster than ERX in all data sets, except for “x60189 _4” and “f25_500” datasets, and in some datasets, it is 68.75% faster (as in f50_498). Also, SCX is 34.64% faster than OX in all data sets except for x60189_4, and SCX is 32.89% faster than CX in all data sets except for “x60189_4”, “x60189_6”, and “f25_500”. There is a similarity in the performance of all crossovers in the small data sets. But the SCX is still the dominant one. This confirms the results obtained previously that with smallest population size SCX still gives the best solution.

TABLE VI. COMPARING SCX WITH THE SMALLER POPULATION SIZE, WITH OTHER CROSSOVERS WITH A LARGER POPULATION SIZE

Data sets	**SCX- 200 individual**		**PMX-500 individual**		**ERX-500 individual**		**OX-500 individual**		**CX-500 individual**	
	Best overlap	**Time**	**Best overlap**	**Time**	**Best overlap**	**Time**	**Best overlap**	**time**	**Best overlap**	**time**
x60189 4	11298	1.93	***11318***	2.2	10817	**1.4**	**11316**	**1.4**	**11252**	**1.6**
x60189 5	***13594***	**1.6**	13304	1.8	12942	1.7	13474	2.1	13324	1.7
x60189 6	***17401***	2.5	16997	**1.8**	16997	3.5	17344	2.6	16272	1.4
x60189 7	***20546***	**1.9**	20536	2.9	19898	3.3	20467	2.8	20477	2.8
m15421 5	***36972***	**5.1**	36964	6.2	36662	6.4	36922	7.3	36899	7
m15421 6	***46304***	**6.4**	46240	7.3	46162	7.4	46287	9.3	46240	7.3
m15421 7	***52069***	**5.4**	52011	11.1	51728	11.9	52011	8.1	52011	11.4
j02459 7	***109094***	**27.4**	108832	34.8	107557	52.1	108832	50	109012	36.3
f25_305	***5594***	**0.2**	***5594***	0.3	**5594**	0.3	5594	0.4	5594	0.3
f25_400	***6307***	**0.2**	***6307***	0.3	6114	0.4	6307	0.4	6307	0.4
f25_500	***9170***	1.6	9013	0.8	8650	**0.4**	9130	2	9122	0.7
f50_315	***9076***	**0.6**	***9076***	1.4	8585	1.7	9076	1.5	9076	1.4
f50_412	***12990***	**1.03**	12960	1.3	12564	1.5	12192	1.7	12886	1.3
f50_498	17051	**1.5**	***17163***	3.7	16868	4.8	17089	4.3	17163	3.2
f100_307	***14319***	**2.6**	14260	3.9	14105	4.8	14314	4.6	14314	6.5
f100_415	***23008***	**3.6**	22854	**3.6**	22163	6.3	22854	3.7	22878	3.9
f100_512	32384	**4.8**	***32352***	5.1	31823	6.5	32266	6.3	32254	6.3

The results of this study are summarized as follows:

- The greedy method had a clear impact on the GA performance and contributed to improving the solution by 95%.
- Most of the time is spent on creating the population especially with the large population size.
- Smart crossover like SCX with small population size is better than simple crossovers with large population size.

C. The Effect of Selection Types Varying the Population Generation on the Overlap Score

This section studies the effect of the initialization and selection types on the overlap score. The following figures show this effect when the type of crossover is SCX and the population size is 200, since previous investigations show that a small population size is more suitable.

Fig. 5, Fig. 6, and Fig. 7 show the effect of initialization types and selection types on the best overlap score. Clearly, the greedy initialization type gives a better overlap score than the random and 2-opt heuristics in 15 data sets out of 17.

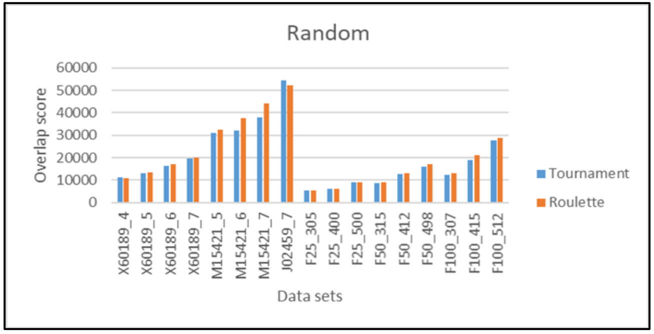


Fig. 5. Effect of selection types with random initialization on best overlap score.

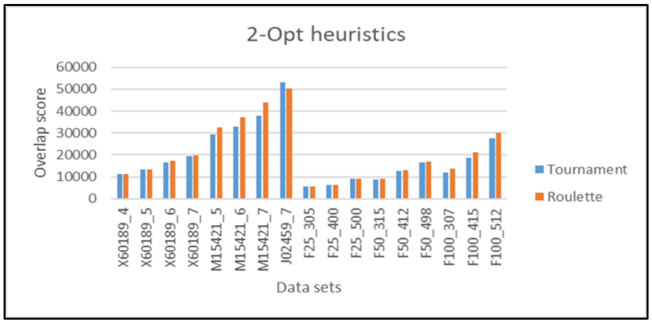


Fig. 6. Effect of selection types with 2-opt heuristics initialization on best overlap score.

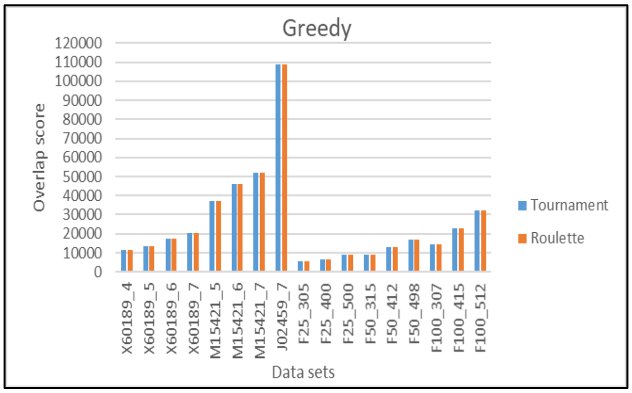


Fig. 7. Effect of selection types with greedy initialization on best overlap score.

Fig. 8, Fig. 9, and Fig. 10 show the effect of initialization and selection types on the average overlap score. The greedy initialization still dominates the random and 2-opt heuristics for the average overlap score as well, giving a higher average in 16 out of 17 data sets. Additionally, the roulette wheel selection is still better than the tournament with the random and 2-opt heuristics. However, with the greedy initialization, the tournament is better.

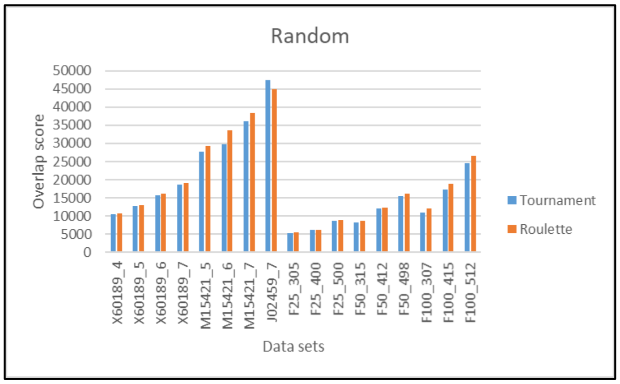


Fig. 8. Effect of selection types with random initialization on average overlap score.

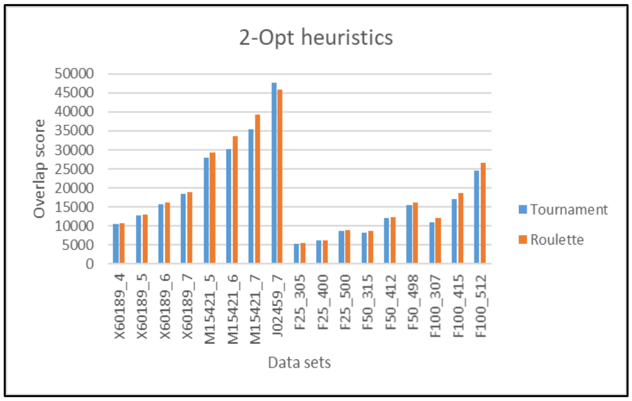


Fig. 9. Effect of selection types with 2-opt heuristics initialization on average overlap score.

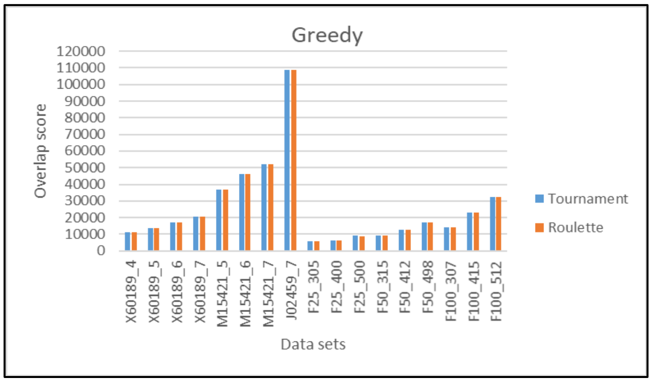


Fig. 10. Effect of selection types with greedy initialization on average overlap score.

The results of this study are summarized as follows:

- The greedy initialization type is the best among the majority of the data sets for the best overlap scores and the best among all the data sets for the average overlap scores.
- The random initialization type is better than the 2-opt for the best overlap score, but their performance is almost similar for the average overlap score.
- The tournament selection type is better than the roulette wheel selection for the best and average overlap scores with the greedy.

D. *The Effect of Selection Types Varying the Population Generation Method on GA Running Time*

Table VII illustrates the results obtained when studying the effect of initialization and selection types on the GA running time. It shows this effect when the type of crossover is SCX, and the population size is 200. The GA time column represents the time it took for the GA to find the result. The least time for every data set is marked in bold green when the selection type is the tournament and marked in bold blue if the selection type is the roulette wheel. The gap column shows the difference in time between the generating methods (i.e., greedy, random, 2-opt heuristic).

Table VII shows that the greedy initialization type is the best from the viewpoint of time complexity. The random and 2-opt heuristics types take more time than the greedy, but the random-type records less time for 12 data sets out of 17, while the 2-opt heuristics record less time for three data sets and equal time for two. As for the selection type, the roulette wheel selection dominates the tournament selection by recording the least time for 14 data sets out of 17. The gap confirms as in the previous section that the greedy generating method is better than the random and 2-opt, and the random is better than 2-opt. In more detail, the results show that the greedy initialization results are fast compared to both random and the 2-opt heuristic in most of the datasets. Except for datasets M15421_7 and J02459_7, the greedy approach is 47.39% and 48.17% faster than random, and 2-opt heuristics when selection type is tournament. Also, greedy approach is 66.90% and 67.19% faster than random, and 2-opt heuristics when selection type is roulette. For the other crossovers OX, CX, PMX, and ERX, the tournament was the best in terms of solution quality, and the roulette was the fastest, because the tournament chose multiple parents every time and compared them to pick the better one.

The results of this study are summarized as follows:

- The greedy initialization type takes less time than the random and the 2-opt heuristics in almost all cases except for the large data set (J02459_7), because the greedy method takes time to create the population, the larger the data, the more comparisons that greedy makes, and therefore it takes longer time.
- The random initialization type is faster than the 2-opt heuristics in 12 data sets out of 17, this is an expected result.
- The roulette wheel selection type is faster than the tournament selection, but the tournament is better for solution quality.

TABLE VII. THE EFFECT OF INITIALIZATION TYPES AND SELECTION TYPES ON THE GA RUNNING TIME (IN SEC.) FOR SCX CROSSOVER ON REAL DATASETS THE BOLD GREEN COLOR REFERS TO THE LEAST TIME FOR THE CORRESPONDING DATA SET WHEN THE SELECTION TYPE IS TOURNAMENT, AND THE BOLD BLUE COLOR REFERS TO THE LEAST TIME WHEN THE SELECTION IS ROULETTE

Data set	Select	random	2-opt	greedy	gap	gap	gap
		GA time	GA time	GA time	Greedy-random	Greedy-2-opt	Random-2-opt
X60189_4	Tournament	4.27	4.3	2.17	-2.1	-2.13	-0.03
	Roulette	4.27	4.43	**0.93**	-3.34	-3.5	-0.16
X60189_5	Tournament	4.4	4.57	3.63	-0.77	-0.94	-0.17
	Roulette	4.7	4.77	**1.37**	-3.33	-3.4	-0.07
X60189_6	Tournament	4.6	4.73	3.63	-0.97	-1.1	-0.13
	Roulette	4.73	4.77	**1.27**	-3.46	-3.5	-0.04
X60189_7	Tournament	4.7	4.7	2.57	-2.13	-2.13	0
	Roulette	4.8	4.8	**1.3**	-3.5	-3.5	0
M15421_5	Tournament	4.87	4.97	4.1	-0.77	-0.87	-0.1
	Roulette	5	5	**3.7**	-1.3	-1.3	0
M15421_6	Tournament	5.03	5	5	-0.03	0	0.03
	Roulette	4.87	5	**4.13**	-0.74	-0.87	-0.13
M15421_7	Tournament	4.93	4.93	6.13	1.2	1.2	0
	Roulette	**4.77**	5	**4.77**	0	-0.23	-0.23
J02459_7	Tournament	**5.1**	**5.1**	25.97	20.87	20.87	0
	Roulette	5.17	5.17	38.27	33.1	33.1	0
F25_305	Tournament	4	4.07	0.23	-3.77	-3.84	-0.07
	Roulette	4.57	4.4	**0.2**	-4.37	-4.2	0.17
F25_400	Tournament	4.33	4.4	**0.2**	-4.13	-4.2	-0.07
	Roulette	4.47	4.53	**0.2**	-4.27	-4.33	-0.06
F25_500	Tournament	4.3	4.47	1.6	-2.7	-2.87	-0.17
	Roulette	4.6	4.5	**1.13**	-3.47	-3.37	0.1
F50_315	Tournament	4.4	4.53	0.87	-3.53	-3.66	-0.13
	Roulette	4.47	4.63	**0.73**	-3.74	-3.9	-0.16
F50_412	Tournament	4.3	4.63	**1.17**	-3.13	-3.46	-0.33
	Roulette	4.6	4.67	1.23	-3.37	-3.44	-0.07
F50_498	Tournament	4.47	4.63	2.37	-2.1	-2.26	-0.16
	Roulette	4.73	4.63	**0.8**	-3.93	-3.83	0.1
F100_307	Tournament	4.77	4.67	**2.53**	-2.24	-2.14	0.1
	Roulette	4.93	4.97	2.97	-1.96	-2	-0.04

F100_415	Tournament	4.8	4.77	4.1	-0.7	-0.67	0.03
	Roulette	4.93	4.93	**2.27**	-2.66	-2.66	0
F100_512	Tournament	4.77	4.8	4.53	-0.24	-0.27	-0.03
	Roulette	4.93	4.97	**3.8**	-1.13	-1.17	-0.04

VI. Discussion

This section discusses the findings that emerged from the results presented in the Results section. And conclude that the small population size (i.e., 200 individuals) is more suitable for most cases. And the greedy type of initialization is the best when look for good overlap score results and time. Furthermore, the results show that the roulette wheel selection type is more suitable than the tournament selection in the context of time, but the tournament is better in the quality of the solution. Also, this work shows that the SCX crossover is the best in the context of best overlap score and average overlap score.

This study has multiple GA versions, but in comparison to the previous works, we selected the best version we got. Moreover, the comparisons are divided as follows:

- Previous works that used the GA, the comparison is presented in Table VIII.
- Previous works that used other metaheuristics algorithms, the comparison is presented in Table IX.

Table VIII compares the designed GA results and the other previous GA work's results in the context of the overlap score. The best results are marked in bold. The "difference in percentage" column shows the difference between our best results and those of the previous works. Clearly, our results for the F-series data sets (from F25_305 to F100_512) dominate all the previous work's results. This work got less than the best results of previous works in eight data sets out of 17, however, our results are still better than [23], [25], and [9] for these data sets.

Moreover, this work obtained better results than the results of all the previous works in nine data sets out of 17.

With regard to the time, the results were given in a reasonable time and there is no significant change or difference in time, because the dominant time is actually not the GA time but the assembly time (i.e., in our case, the Smith-Waterman algorithm.). GA is useful when the data set is large, and this is expected because GA avoids large search space. The results show that the designed GA gives the results in less time for large data sets such as M15421_6, M15421_7, and J02459_7, which have several fragments that vary from 173 to 352 characters.

TABLE VIII. Comparison of Best Solutions between our GA Results and other GA Algorithms Results from the Literature in the Context of Overlap Score

Data sets	Our best GA	REF. [9]	REF. [10]	REF. [31]	Difference in percentage %
X60189_4	11272	6488	**11478**	**11478**	-1.79%
X60189_5	13475	8655	14027	**14161**	-4.84%
X60189_6	17357	9943	**18301**	**18301**	-5.16%
X60189_7	20559	11546	**21268**	21212	-3.33%
M15421_5	36972	22598	**38726**	38694	-4.53%
M15421_6	46240	29469	48048	**48052**	-3.77%
M15421_7	52077	32744	**55072**	55071	-5.44%
J02459_7	109043	68736	115301	**116487**	-6.39%
F25_305	**5594**	2271	-	596	146.32%
F25_400	**6307**	3139	-	777	100.92%
F25_500	**9170**	5777	-	921	58.73%
F50_315	**9076**	4013	-	1578	126.16%
F50_412	**12967**	5835	-	1572	122.23%
F50_498	**16902**	9050	-	1570	86.76%
F100_307	**14318**	7035	-	2780	103.53%
F100_415	**22911**	9202	-	2846	148.98%
F100_512	**32384**	11881	-	2717	172.57%

TABLE IX. COMPARISON OF BEST SOLUTIONS BETWEEN OUR GA RESULTS AND OTHER METAHEURISTICS NON-GA ALGORITHMS RESULTS FROM THE LITERATURE IN THE CONTEXT OF OVERLAP SCORE

Data set	**Our best GA**	**REF. [23]**	**REF. [3]**	**REF. [25]**	**REF. [28]**	**REF. [27]**	**REF. [33]**	**Difference in Percentage %**
X60189_4	11272	-	**11478**	3046	**11478**	11451	**11478**	-1.79%
X60189_5	13475	-	13642	-	**14161**	13932	**14161**	-4.84%
X60189_6	17357	-	**18301**	-	**18301**	18204	**18301**	-5.16%
X60189_7	20559	-	20921	3022	**21271**	20968	**21271**	-3.35%
M15421_5	36972	5821	38686	6443	**38746**	38454	**38746**	-4.58%
M15421_6	46240	6713	47669	7041	**48052**	-	**48052**	-3.77%
M15421_7	52077	6291	54891	6537	**55171**	54666	**55171**	-5.61%
J02459_7	109043	-	114381	-	**116700**	115405	**116700**	-6.56%
F25_305	**5594**	-	-	-	596	-	596	838.59%
F25_400	**6307**	-	-	-	777	-	777	711.71%
F25_500	**9170**	-	-	-	921	-	921	895.66%
F50_315	**9076**	-	-	-	1581	-	1581	474.07%
F50_412	**12967**	-	-	-	1573	-	1573	724.35%
F50_498	**16902**	-	-	-	1570	-	1570	976.56%
F100_307	**14318**	-	-	-	2793	-	2793	412.64%
F100_415	**22911**	-	-	-	2860	-	2860	701.08%
F100_512	**32384**	-	-	-	2732	-	2732	1085.36%

VII. CONCLUSION

This paper is a continuation of our previous work [4] to solve the DNA fragment assembly problem. As was pointed out in the introduction to this paper, the DNAFA is an optimization problem that attempts to reconstruct an original DNA sequence by finding the shortest DNA sequence from a given set of fragments. We have designed a platform for the genetic algorithm, from which more than one version of the genetic algorithm can be deduced to solve this problem. The design was inspired by the good designs that solved TSP and QAP problems. This study is the first to our knowledge that examines the genetic algorithm for the DNAFA problem from this perspective. In more detail, this study has gone a long way towards investigating the effect of genetic algorithm operators on the quality of the solution to the DNAFA problem. The study focused on investigating the effect of the initial population, size of the population, selection types, and crossover types.

This paper recorded the important results and came out with some findings, the most obvious finding to emerge from this study is that the SCX crossover is a smart crossover and has never been used before with DNA_FA, SCX crossover gave better results compared to the rest of the studied crossover types. Furthermore, the results show that the population generation method has the greatest influence on GA performance in terms of time and solution quality. Also, we configured the best-designed GA variant that outperforms the existing GA algorithms solving the DNAFA problem. This GA variant features the use of 200 individuals for the population size along with the greedy method for initializing the population, tournament selection, and SCX crossover. This study has found that generally, the size of the population does not significantly affect the quality of the solution, especially if the type of initialization is good. The results were good and competitive compared to the results of previous works. Our design showed that the results were better than all previous results from the literature for some data sets.

There is still ample scope to study and solve this problem, an interesting point will be how to find a way to create the population intelligently and without consuming a lot of time, given that the greedy is time consuming. Moreover, further research might explore or investigate the effect of other GA operators (e.g., mutation types and stooping conditions). Also, investigate the effect of combining different types of GA operators (initialization types, crossover operators, and mutation operators) on the results. Another possible area of future research would be to combine the data sets (next generation with the third generation).

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding and supporting this work through Graduate Students Research Support Program.

REFERENCES

[1] R. S. Verma and S. kumar, “DSAPSO: DNA sequence assembly using continuous Particle Swarm Optimization with Smallest Position Value rule,” in 2012 1st International Conference on Recent Advances in

Information Technology (RAIT), Mar. 2012, pp. 410–415. doi: 10.1109/RAIT.2012.6194455.

[2] G. M. Mallén-Fullerton and G. Fernández-Anaya, "DNA fragment assembly using optimization," in 2013 IEEE Congress on Evolutionary Computation, Jun. 2013, pp. 1570–1577. doi: 10.1109/CEC.2013.6557749.

[3] E. Alba and G. Luque, "A Hybrid Genetic Algorithm for the DNA Fragment Assembly Problem," in Recent Advances in Evolutionary Computation for Combinatorial Optimization, vol. 153, C. Cotta and J. van Hemert, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 101–112. doi: 10.1007/978-3-540-70807-0_7.

[4] H. Bennaceur, M. Almutairy, and Alqhtani, Nora, "An Investigative Study of Genetic Algorithms to Solve the DNA Assembly Optimization Problem," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 10, p. 12, 2020.

[5] G. Luque and E. Alba, "Metaheuristics for the DNA Fragment Assembly Problem," 2005. doi: 10.5019/j.ijcir.2005.28.

[6] D. Bucur, "De Novo DNA Assembly with a Genetic Algorithm Finds Accurate Genomes Even with Suboptimal Fitness," in Applications of Evolutionary Computation, vol. 10199, G. Squillero and K. Sim, Eds. Cham: Springer International Publishing, 2017, pp. 67–82. doi: 10.1007/978-3-319-55849-3_5.

[7] J. Hughes, S. Houghten, G. M. Mallén-Fullerton, and D. Ashlock, "Recentering and Restarting Genetic Algorithm variations for DNA Fragment Assembly," in 2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, May 2014, pp. 1–8. doi: 10.1109/CIBCB.2014.6845500.

[8] G. Minetti, E. Alba, and G. Luque, "Seeding strategies and recombination operators for solving the DNA fragment assembly problem," Inf. Process. Lett., vol. 108, no. 3, pp. 94–100, Oct. 2008, doi: 10.1016/j.ipl.2008.04.005.

[9] Uzma and Z. Halim, "Optimizing the DNA fragment assembly using metaheuristic-based overlap layout consensus approach," Appl. Soft Comput., vol. 92, p. 106256, Jul. 2020, doi: 10.1016/j.asoc.2020.106256.

[10] G. Minetti, G. Leguizamon, and E. Alba, "SAX: a new and efficient assembler for solving DNA Fragment Assembly Problem," p. 12, 2012.

[11] D. Bucur, "A stochastic de novo assembly algorithm for viral-sized genomes obtains correct genomes and builds consensus," Inf. Sci., vol. 420, pp. 184–199, Dec. 2017, doi: 10.1016/j.ins.2017.07.039.

[12] Z. H. Ahmed, "GENETIC ALGORITHM FOR THE TRAVELING SALESMAN PROBLEM USING SEQUENTIAL CONSTRUCTIVE CROSSOVER," 2010.

[13] M. M. Alipour, S. N. Razavi, M. R. Feizi Derakhshi, and M. A. Balafar, "A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem," Neural Comput. Appl., vol. 30, no. 9, pp. 2935–2951, Nov. 2018, doi: 10.1007/s00521-017-2880-4.

[14] Z. H. Ahmed, "An improved genetic algorithm using adaptive mutation operator for the quadratic assignment problem," in 2015 38th International Conference on Telecommunications and Signal Processing (TSP), Jul. 2015, pp. 1–5. doi: 10.1109/TSP.2015.7296481.

[15] S. S. Juneja, P. Saraswat, K. Singh, J. Sharma, R. Majumdar, and S. Chowdhary, "Travelling Salesman Problem Optimization Using Genetic Algorithm," in 2019 Amity International Conference on Artificial Intelligence (AICAI), Feb. 2019, pp. 264–268. doi: 10.1109/AICAI.2019.8701246.

[16] W. Xueyuan, "Research on Solution of TSP Based on Improved Genetic Algorithm," in 2018 International Conference on Engineering Simulation and Intelligent Control (ESAIC), Aug. 2018, pp. 78–82. doi: 10.1109/ESAIC.2018.00025.

[17] R. Liu and Y. Wang, "Research on TSP Solution Based on Genetic Algorithm," in 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), Jun. 2019, pp. 230–235. doi: 10.1109/ICIS46139.2019.8940186.

[18] H. Bennaceur and Z. Ahmed, "Frequency model based crossover operators for genetic algorithms applied to the quadratic assignment problem," Int Arab J Inf Technol, vol. 14, pp. 138–145, 2017.

[19] Z. H. Ahmed, "A Simple Genetic Algorithm using Sequential Constructive Crossover for the Quadratic Assignment Problem," vol. 73, p. 4, 2014.

[20] Zakir Hussain Ahmed, "Solving the Traveling Salesman Problem using Greedy Sequential Constructive Crossover in a Genetic Algorithm," Fabruary 2020.

[21] Zakir Hussain Ahmed, "Genetic Algorithm with Comprehensive Sequential Constructive Crossover for the Travelling Salesman Problem," IJACSA Int. J. Adv. Comput. Sci. Appl. Vol 11 No 5, 2020.

[22] Z. H. Ahmed, H. Bennaceur, M. H. Vulla, and F. Altukhaim, "A Hybrid Genetic Algorithm for the Quadratic Assignment Problem," p. 7.

[23] K. Huang, J. Chen, and C. Yang, "A Hybrid PSO-Based Algorithm for Solving DNA Fragment Assembly Problem," in 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications, Sep. 2012, pp. 223–228. doi: 10.1109/IBICA.2012.8.

[24] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem," Oper. Res., vol. 21, no. 2, pp. 498–516, 1973.

[25] K.-W. Huang, J.-L. Chen, C.-S. Yang, and C.-W. Tsai, "A memetic particle swarm optimization algorithm for solving the DNA fragment assembly problem," Neural Comput. Appl., vol. 26, no. 3, pp. 495–506, Apr. 2015, doi: 10.1007/s00521-014-1659-0.

[26] R. Indumathy, S. Uma Maheswari, and G. Subashini, "Nature-inspired novel Cuckoo Search Algorithm for genome sequence assembly," Sadhana, vol. 40, no. 1, pp. 1–14, Feb. 2015, doi: 10.1007/s12046-014-0300-3.

[27] E. Alba and G. Luque, "A New Local Search Algorithm for the DNA Fragment Assembly Problem," in Evolutionary Computation in Combinatorial Optimization, vol. 4446, C. Cotta and J. van Hemert, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–12. doi: 10.1007/978-3-540-71615-0_1.

[28] G. Minetti, G. Leguizamón, and E. Alba, "An improved trajectory-based hybrid metaheuristic applied to the noisy DNA Fragment Assembly Problem," Inf. Sci., vol. 277, pp. 273–283, Sep. 2014, doi: 10.1016/j.ins.2014.02.020.

[29] J. S. Firoz, M. S. Rahman, and T. K. Saha, "Bee algorithms for solving DNA fragment assembly problem with noisy and noiseless data," in Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12, Philadelphia, Pennsylvania, USA, 2012, p. 201. doi: 10.1145/2330163.2330192.

[30] F. Majid al-Rifaie and M. Majid al-Rifaie, "Maximising Overlap Score in DNA Sequence Assembly Problem by Stochastic Diffusion Search," in Intelligent Systems and Applications, vol. 650, Y. Bi, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2016, pp. 301–321. doi: 10.1007/978-3-319-33386-1_15.

[31] J. A. Hughes, S. Houghten, and D. Ashlock, "Restarting and recentering genetic algorithm variations for DNA fragment assembly: The necessity of a multi-strategy approach," Biosystems, vol. 150, pp. 35–45, Dec. 2016, doi: 10.1016/j.biosystems.2016.08.001.

[32] E. Çela, V. G. Deineko, and G. J. Woeginger, "The multi-stripe travelling salesman problem," Ann. Oper. Res., vol. 259, no. 1, pp. 21–34, 2017, doi: 10.1007/s10479-017-2513-4.

[33] A. Ben Ali, G. Luque, and E. Alba, "An efficient discrete PSO coupled with a fast local search heuristic for the DNA fragment assembly problem," Inf. Sci., vol. 512, pp. 880–908, Feb. 2020, doi: 10.1016/j.ins.2019.10.026.

Fake News Classification Web Service for Spanish News by using Artificial Neural Networks

Patricio Xavier Moreno-Vallejo[1], Gisel Katerine Bastidas-Guacho[2],
Patricio Rene Moreno-Costales[3], Jefferson Jose Chariguaman-Cuji[4]
Escuela Superior Politécnica de Chimborazo (ESPOCH), Riobamba, Ecuador[1, 2, 3]
Independent Researcher, Riobamba, Ecuador[4]

Abstract—**The use of digital media, such as social networks, has promoted the spreading of fake news on a large scale. Therefore, several Machine Learning techniques, such as artificial neural networks, have been used for fake news detection and classification. These techniques are widely used due to their learning capabilities. Besides, models based on artificial neural networks can be easily integrated into social media and websites to spot fake news early and avoid their propagation. Nevertheless, most fake news classification models are available only for English news, limiting the possibility of detecting fake news in other languages, such as Spanish. For this reason, this study proposes implementing a web service that integrates a deep learning model for the classification of fake news in Spanish. To determine the best model, the performance of several neural network architectures, including MLP, CNN, and LSTM, was evaluated using the F1 score., and LSTM using the F1 score. The LSTM architecture was the best, with an F1 score of 0.746. Finally, the efficiency of web service was evaluated, applying temporal behavior as a metric, resulting in an average response time of 1.08 seconds.**

Keywords—Fake news; LSTM; classification; web service; machine learning

I. Introduction

In the last few years, the use of technology and digital media has increased. Besides, due to the pandemic, people have chosen to use digital media, such as social networks, to get news [1]. However, a large amount of unverified news could be fake [2]. Fake news has always existed, but there is a greater irruption in open and unrestricted access platforms nowadays. Therefore, the propagation of fake news can cause information theft, scams, collective hysteria, or discredit the honor of a person or an institution [3]. In order to solve the problem of fake news propagation, some techniques based on machine learning, Natural Language Processing, and information retrieval have been proposed to detect this kind of news automatically and make decisions [4]. Machine learning models have been used effectively in text classification to determine whether a text is racist, positive or negative, fake or genuine, and so on.

There are many classification models to determine whether the news is authentic or fake [5]–[10]. In [11], the authors propose a Naïve Bayes model for fake news classification with an accuracy of 84%. On the other hand, in [12], web scrapping and crawler techniques are applied in order to create a dataset with tweets and Facebook web links to train models. Since there are many machine learning models like decision tree, Support Vector Machine (SVM), Naïve Bayes, random forest, and Logistic Regression, some works have compared their performance in the task of news classification as fake or real. On the one hand, in [13], the comparison reveals that Naïve Bayes achieves the best performance on the tested datasets achieving an accuracy of 66%, and, in [14], a Random Forest model allows the classification of news into fake or real with an accuracy of 76.94%. On the other hand, in [15], the authors propose a neural network architecture that outperforms previous approaches with an accuracy of 94.21% on test data. Nonetheless, achieving these high-performance values is challenging for detecting fake news in Spanish due to the limited datasets in this language.

Most of the machines learning models aim to classify news in English. Therefore, this study presents a machine learning based framework to classify news in Spanish into real or fake. The web service with the classifier model was developed following the CRISP-DM and SCRUM methodologies. The main contributions of this study are listed as follows:

- Machine learning model to classify fake news in Spanish. To the best of our knowledge, this model is the first one to classify news in Spanish by using MLP architecture.
- A web site based on web service architecture is built to access the classifier. That is, the machine learning model is integrated to the web service to be available in real-time.

The remainder of this paper is organized as follows. Section II describes previous studies related to fake news classification. Section III then presents the methodology applied in this study. Section IV describes the classification model with the integration into the web service. Section V presents the experimental results and evaluation metrics of the proposed classifier, while Section VI presents the conclusion of this work.

II. Related Work

Since fake news can be harmful to individuals or organizations, effective ways to detect this kind of news have been developed through the years. The problem can be framed as a two-class (real/fake) classification. Therefore, this study reviews approaches for fake news classification based on machine learning classifiers, such as decision trees, support vector machines (SVM), and Naïve Bayes. In [16] , the researchers propose a fake news detection system that uses the decision tree algorithm to classify the news from two sources. Then, they compare the results against the result obtained with the SVM algorithm showing that the results obtained with the decision tree are more accurate than with SVM, with an accuracy of 97.67% and a precision of 94.60% against the SVM results of 91.74% in accuracy and 90.12% in precision.

On the other hand, the work done in [17] compares the SVM with an apriori algorithm performance on a dataset that contains four attributes and 311 instances. In this case, the results show that SVM achieves a better accuracy of 91.87% while the apriori algorithm only gets an accuracy of 31.76%. It means it is better to project data into a hyperplane to look for discriminative features in this projected space rather than look for them directly in the source data space [18]–[20]. In [21], the authors apply a machine learning algorithm and use Natural Language Processing (NLP) to pre-process the data to increase the accuracy of the machine learning algorithms. The researchers pre-process a dataset from the Kaggle website of 20,800 news articles containing 10,387 real news and 10,413 fake news. The pre-processing consists of tokenization, removing stop words, stemming, and vectorizer through term frequency-inverse document frequency (TF-IDF). Then the pre-processed data is used to feed six ML algorithms: logistic regression, Naïve Bayes, K-Nearest Neighbor (KNN), SVM, random forest, and decision tree. They use the K-fold cross-validation, confusion matrix, and classification report (precision, recall, and F1-score) for model evaluation. Thus, after comparing the various models, the accuracy score is used to determine the best one giving the result that random forest and decision tree models achieve the best performance with over 99% accuracy. In comparison, the KNN got the worst accuracy, with ~52% accuracy, and the other models' accuracies were over 96%. The results indicate that NLP contributes to models' training improving their accuracy.

Helmstetter et al. [22] use a dataset collected from Twitter. From the dataset, they use the text features as well as the metadata from Twitter, which corresponds to user-level features (e.g., number of followers) and tweet-level features (e.g., number of retweets). Moreover, they extract additional features, such as the sentiment features using SentiWordNet [23] and the topic features using Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process (HDP) models. Then, the features are scaled and selected in order to feed the learning algorithms that include some ML algorithms; however, the best performance is achieved by the ensemble method XG-Boost which achieves a 0.94 F1-score. Another approach, like the one presented in [24], also proposes a voting ensemble mechanism with three classifiers: Decision Trees, Naïve Bayes, and KNN to achieve a lower error rate than using the models separately. Besides, Wynne et al. [25] proposed a large ensemble model called the two-layer ensemble model, where there is a first layer that contains two sets of voting classifiers with five classifiers each, and the output of these two voting classifiers are used as input for the second layers which contains a third final voting classifier. In this case, the final voting classifier performs best on the LIAR dataset. However, this approach can be time-consuming and needs a good number of resources since one needs to train each of the ten classifiers. Furthermore, Tian et al. [26] use a feature selection method called Genetic and Evolutionary Feature Selection (GEFES) [27] to identify a subset of features by means of a steady-state genetic algorithm. The selected features are used as input for a KNN model to classify news as fake or genuine, improving the model's performance. For experiments, the authors use the dataset BuzzFace which consists of 2282 news from Facebook related to the 2016 US presidential elections. On this dataset, the authors achieved an accuracy of 91.3% for the classification task using the proposed approach.

Some researchers have proposed methods that work with news in other languages different than English. For instance, Billones et al. [28] train ML models with a dataset of news in Filipino and compare the performance of the models with models trained with English news. The results show that the models perform better on the English dataset than on the other language dataset due to the limitation of labeled data amount in other languages. Khalil et al. [29] tackle the problem of few data in other languages creating a large Arabic fake news corpus of 606912 articles. Then they train and test some deep learning algorithms getting the best accuracy of 78.3% with the capsule network [30] that is based on a convolutional neural network. Additionally, many other works use deep learning strategies for English news classification. For instance, Alameri et al. [31] use neural networks and Long Short-Term Memory (LSTM) networks and compare them with classic ML approaches giving as the result that the LSTM outperforms the other models in terms of accuracy, precision, recall, and F1-score. Gupta et al. [32] also use deep learning approaches that include Convolutional Neural Networks (CNN) and LSTM, which are tested individually and, after, are used as an ensemble model to get the probability of truth. In this case, individually, CNN gets better results than LSTM. Mahmud et al. [33] use Graph Neural Networks (GNN) to make predictions based on text and graph data where the text is extracted using text representation learning techniques, and the graph contains the news propagation data. The addition of graph data boosts the performance of the model.

The literature review highlights the importance of deep learning models in the task of news classification and emphasizes the need for datasets in languages other than English, such as Spanish. Therefore, this study proposes conducting a comparative analysis of machine learning models and deep learning models for detecting fake news in Spanish to develop a tool that implements the best method to determine the authenticity of news in real-time.

III. Methodology

This section describes the applied methodologies for creating the fake news classification web service. CRISP-DM was applied to implement the fake news classifier model, while SCRUM was used to develop the web service (see Fig. 1).

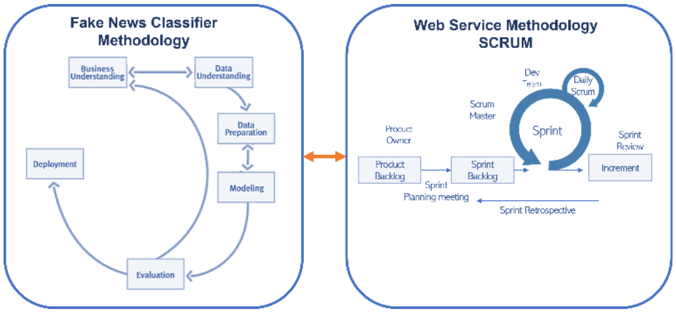


Fig. 1. Methodologies used to implement the proposed classifier.

A. CRISP-DM Methodology

In order to develop the fake news classifier based on deep learning techniques, CRISP-DM Methodology is applied. This methodology consists of six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

The phases are summarized in the following points:

- Business Understanding: In this phase, the fake news problem is analyzed. To this end, the state-of-the-art frameworks for fake news classification are reviewed, allowing to determine that there are few fake news classifiers in Spanish since most existing approaches focus on English news. Thus, this study aims to develop a fake news classifier for Spanish news.

- Data Understanding: Several datasets were analyzed to create a suitable dataset in Spanish for training the model. The final dataset consists of news extracted from the following sources: Fake-news-in-Spanish[1], Fake and real news[2, 3] and Fake news Corpus Spanish[4]. Besides, news from a website with fake Ecuadorian news was extracted to complement the dataset. The dataset consists of 9294 news and the corresponding labels: 4081 fake news and 5213 real news.

- Data Preparation: In this phase, the data are integrated from multiple sources using Python. Each news is preprocessed by removing stop words; that is, the dataset does not contain words such as “the”, “is”, “are”, etc. Then, stemming is applied to each word to extract the base form of the word. For instance, the stemming of the word “eating” is “eat”. Finally, tokenization is applied for each news.

- Modeling: The review of state-of-the-art approaches for fake news classification (e.g., Naïve Bayes, Logistic Regression, Support Vector Machine, K-nearest neighbor, Decision Trees, and deep-learning-based approaches, to name a few) shows that deep learning models are the most suitable for fake news classification. Therefore, artificial neural network architectures such as MLP (Multilayer Perceptron), CNN (Convolutional Neural Networks), and LSTM (Long Short-Term Memory) were implemented in order to determine the best model for detecting fake news in a Spanish corpus. After the implementation, the models were trained using 80% of news from the dataset. The 20% of data was used for testing.

- Evaluation: The models based on MLP, CNN and LSTM are evaluated by using F1 score. According to the results of this quantitative evaluation, the LTSM model was the model with the best F1 score for classifying Spanish news as fake or real.

- Deployment: In this phase, the classifier was integrated with a web service to be available online.

B. SCRUM

In order to develop a web service that allows the execution of the fake news classifier, SCRUM methodology is followed. This methodology mainly consists of:

- Product Backlog, where is defined all the system requirements.

- Sprint Backlog refers to the requirements to be developed in the spring. They are defined during the spring planning meeting.

- Sprint is the period that the Scrum Team (developers) uses to complete the development of the requirements within the Sprint Backlog. After a Sprint is completed, there are the Sprint Review and the Sprint Retrospective. Each sprint had duration of 50 hours.

The architectural design SOA (Service Oriented Architecture) is defined for the system and a user interface is designed (see Fig. 2). Then, the system that integrates the fake news classifier model was developed using the Flask framework. In addition, HTML, CSS, and JavaScript are used for the front end.

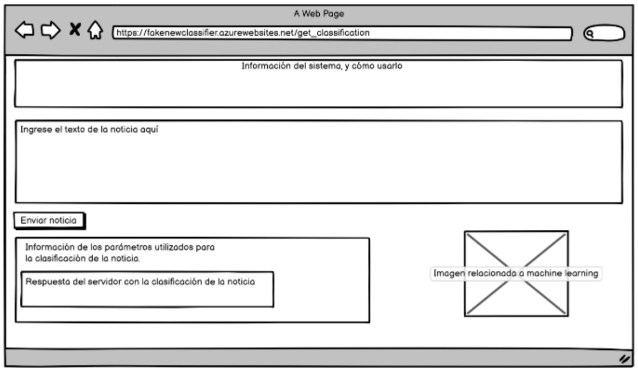

A Web Page
https://fakenewclassifier.azurewebsites.net/get_classification
Información del sistema, y cómo usarlo
Ingrese el texto de la noticia aquí
Enviar noticia
Información de los parámetros utilizados para la clasificación de la noticia.
Respuesta del servidor con la clasificación de la noticia
Imagen relacionada a machine learning

Fig. 2. User interface design.

IV. The Proposed Method

This section introduces the fake news classifier integrated into a web service available through the Internet. The overview of the proposed solution is shown in Fig. 3.

[1] https://www.kaggle.com/datasets/arseniitretiakov/noticias-falsas-en-espaol

[2] https://www.kaggle.com/datasets/zulanac/fake-and-real-news

[3] https://huggingface.co/datasets/julien-c/impressionists/blob/main/data/train-00000-of-00001.parquet

[4] https://huggingface.co/datasets/sayalaruano/FakeNewsCorpusSpanish

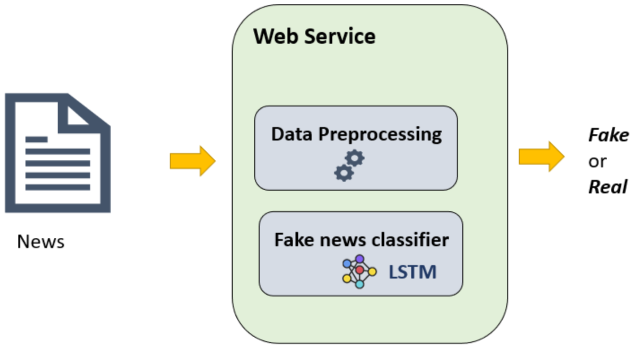


Fig. 3. Overview of the proposed solution.

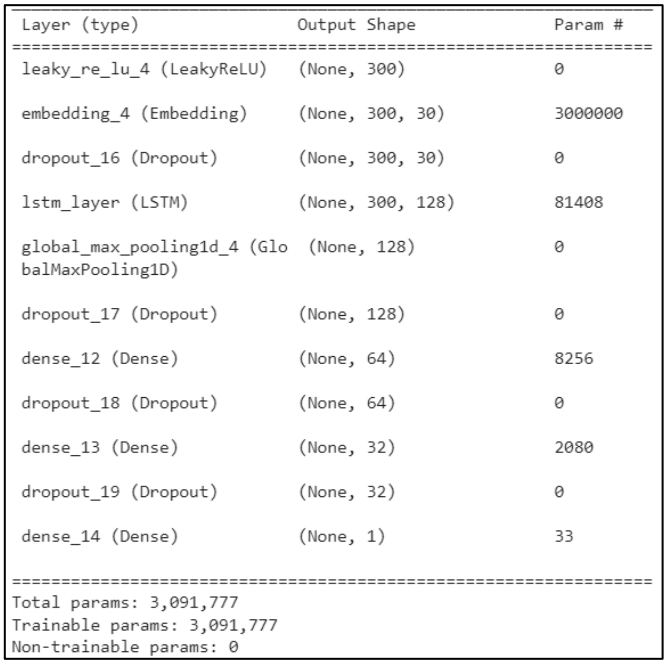
The proposal includes a binary classifier that indicates whether the news is fake or real. The model's input is pre-processed text by removing stopwords, applying stemming, and tokenization. On the other hand, the classifier model is based on an LSTM architecture, and different hyperparameters were evaluated to select the best model configuration. Fig. 4 summarizes the final architecture of the model.

V. Results

This section presents the results of the evaluation of the proposed model. As aforementioned, there was a comparison of several shallow models with deep learning models, specifically with artificial neural networks (ANN). Since ANNs had a better performance, three ANN architectures were evaluated: MLP, CNN, and LSTM to select the best model for the proposed solution.

The evaluation metric used to evaluate the model performance was the F1 score, whose formula is:

$$F1 = 2 * \frac{Precision*recall}{Precision+recall} \quad (1)$$



```
 Layer (type)                Output Shape              Param #
=================================================================
 leaky_re_lu_4 (LeakyReLU)   (None, 300)               0

 embedding_4 (Embedding)     (None, 300, 30)           3000000

 dropout_16 (Dropout)        (None, 300, 30)           0

 lstm_layer (LSTM)           (None, 300, 128)          81408

 global_max_pooling1d_4 (Glo  (None, 128)              0
 balMaxPooling1D)

 dropout_17 (Dropout)        (None, 128)               0

 dense_12 (Dense)            (None, 64)                8256

 dropout_18 (Dropout)        (None, 64)                0

 dense_13 (Dense)            (None, 32)                2080

 dropout_19 (Dropout)        (None, 32)                0

 dense_14 (Dense)            (None, 1)                 33

=================================================================
Total params: 3,091,777
Trainable params: 3,091,777
Non-trainable params: 0
```

Fig. 4. LTMS Architecture.

TABLE I. Evaluation Results

Model Architecture	***F1 Score***
MLP	0.738
CNN	0.678
LSTM	**0.746**

Table I shows the results of the models' evaluation. LSTM achieves the best performance with an F1 score of 0.746, followed by MLP and CNN with an F1 score of 0.738 and 0,678, respectively. Since LSTM is the best model, the developed web service integrates this model.

On one hand, the CNNs performance is poorly against MLP and LSTM since it is more oriented to work with images than with text. On the other hand, LSTM networks perform better than others since these kinds of networks are capable of learning sequences that allow a classification based on the context of the whole text.

During the training stage the values returned by the loss function were decreasing almost constantly all the time, however, the validation loss values just decreased until 100-ish epoch as shown in Fig. 5. It means that after epoch 100 the model starts to overfit then it was trained only until this epoch to avoid overfitting.

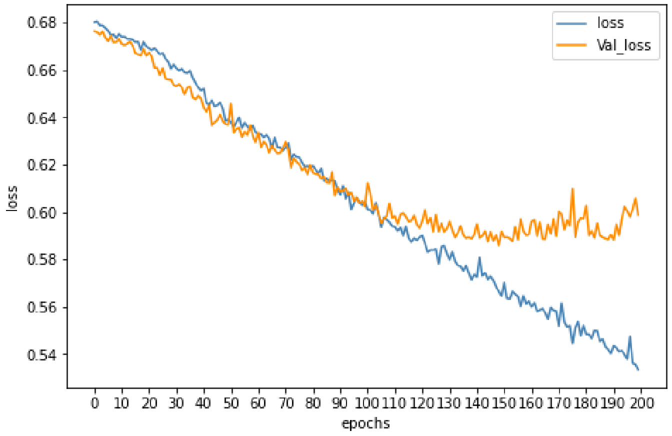


Fig. 5. Loss functions values updates through epochs.

Afterwards, the obtained results are compared with some previous work that have trained machine learning and deep learning models for fake news detection with datasets containing news in Spanish. Table II presents the F1 Scores obtained by six methods on the task of fake news classification in Spanish. The results show that the proposed method achieves the best performance while the random forest approach gets the worst performance. Additionally, it should be noted that there are few works that use a Spanish corpus and in the existing works the used corpus are small containing at most a set of 1500 news against the set of 9294 news that was used in the current research.

Furthermore, the performance of the web service was evaluated to determine the time required for news classification. The web service was deployed on Microsoft Azure, and its time response was considered as a metric, starting from the moment the user sends the request to the web service until it returns a classification response. Moreover, three scenarios were defined based on the length of the news content, which are described in Table III.

TABLE II. F1-SCORE OF DIFFERENT METHODS USED FOR FAKE NEWS CLASSIFICATION

Method	***F1 Score***
Logistic Regression [34]	0.693
Random Forest [35]	0.560
Sentence BERT + Heur[36]	0.701
Ensemble [37]	0.686
XLM-RoBERTa[38]	0.705
LSTM (ours)	**0.746**

TABLE III. SCENARIOS

Scenario	***News length interval (Number of characters)***
Scenario 1 – Small news	1300-2000
Scenario 2 – Medium news	2001-3200
Scenario 3 – Large news	>= 3201

Each scenario consists of 1386 requests to the web service for classifying news. Table IV summarizes the results of time response (seconds) after executing all requests for each scenario with news content of small, medium, and large lengths. The mean response time of the web service is 1.08 seconds, indicating good performance according to [39], as websites with good time behavior must respond within a maximum of 10 seconds before users leave the site.

TABLE IV. RESPONSE TIME

Scenario	***Response Time (seconds)***		
	Minimum	***Mean***	***Maximum***
Scenario 1 – Small news	0.71	1.01	1.43
Scenario 2 – Medium news	0.75	1.05	1.53
Scenario 3 – Large news	0.82	1.17	1.70
Total	0.76	**1.08**	1.55

VI. CONCLUSION

The study proposes a Fake Spanish News Classifier trained in a supervised fashion, utilizing LSTM architecture to determine the authenticity of Spanish news. The selection of LSTM architecture is based on its superior performance compared to MLP and CNN architectures since this architecture allows having a memory state, maintaining a relationship between the sequences of words in the text, which implies better performance when making the classifications. The method was evaluated using both private and public datasets, with experiments showing an F1 score of 0.746 which is good for a dataset with limited news in Spanish. However, this value is lower than the values in other studies that use datasets in English, noticing that the predictions in languages different than English are challenging mainly due to the limitation of data. Besides, a web service integrating the classifier has been implemented to detect fake news in real-time. For future work, the news can be classified into additional labels such as satire or junk science, for a more comprehensive understanding of fake news available on the internet that allows detecting them prior to propagation.

REFERENCES

[1] A. Wong, S. Ho, O. Olusanya, M. V. Antonini, and D. Lyness, “The use of social media and online communications in times of pandemic COVID-19,” https://doi.org/10.1177/1751143720966280, vol. 22, no. 3, pp. 255–260, Oct. 2020, doi: 10.1177/1751143720966280.

[2] G. Di Domenico, J. Sit, A. Ishizaka, and D. Nunan, “Fake news, social media and marketing: A systematic review,” J Bus Res, vol. 124, pp. 329–341, Jan. 2021, doi: 10.1016/J.JBUSRES.2020.11.037.

[3] V. Bakir and A. McStay, “Fake News and The Economy of Emotions: Problems, causes, solutions,” Digital Journalism, vol. 6, no. 2, pp. 154–175, Feb. 2018, doi: 10.1080/21670811.2017.1345645.

[4] R. Zafarani, X. Zhou, K. Shu, and H. Liu, “Fake news research: Theories, detection strategies, and open problems,” Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 3207–3208, Jul. 2019, doi: 10.1145/3292500.3332287.

[5] S. R. Sahoo and B. B. Gupta, “Multiple features based approach for automatic fake news detection on social networks using deep learning,” Appl Soft Comput, vol. 100, p. 106983, Mar. 2021, doi: 10.1016/J.ASOC.2020.106983.

[6] B. Probierz, P. Stefanski, and J. Kozak, “Rapid detection of fake news based on machine learning methods,” Procedia Comput Sci, vol. 192, pp. 2893–2902, Jan. 2021, doi: 10.1016/J.PROCS.2021.09.060.

[7] K. Poddar, G. B. D. Amali, and K. S. Umadevi, “Comparison of Various Machine Learning Models for Accurate Detection of Fake News,” 2019 Innovations in Power and Advanced Computing Technologies, i-PACT 2019, Mar. 2019, doi: 10.1109/I-PACT44901.2019.8960044.

[8] R. K. Kaliyar, A. Goswami, and P. Narang, “FakeBERT: Fake news detection in social media with a BERT-based deep learning approach,” Multimed Tools Appl, vol. 80, no. 8, pp. 11765–11788, Mar. 2021, doi: 10.1007/S11042-020-10183-2/TABLES/22.

[9] Z. Khanam, B. N. Alwasel, H. Sirafi, and M. Rashid, “Fake News Detection Using Machine Learning Approaches,” IOP Conf Ser Mater Sci Eng, vol. 1099, no. 1, p. 012040, Mar. 2021, doi: 10.1088/1757-899X/1099/1/012040.

[10] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, “Fake News Detection Using Machine Learning Ensemble Methods,” Complexity, vol. 2020, 2020, doi: 10.1155/2020/8885861.

[11] G. E. R. Agudelo, O. J. S. Parra, and J. B. Velandia, “Raising a model for fake news detection using machine learning in Python,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11195 LNCS, pp. 596–604, 2018, doi: 10.1007/978-3-030-02131-3_52/FIGURES/3.

[12] K. Shu, S. Wang, and H. Liu, “Exploiting Tri-Relationship for Fake News Detection,” ArXiv, 2017.

[13] A. H. Hoti, M. H. Hoti, H. Hoti, and A. Salihu, “Identifying Fake News written on Albanian language in social media using Naïve Bayes, SVM, Logistic Regression, Decision Tree and Random Forest algorithms,” 2022 11th Mediterranean Conference on Embedded Computing, MECO 2022, 2022, doi: 10.1109/MECO55406.2022.9797147.

[14] K. Martínez-Gallego, A. M. Álvarez-Ortiz, and J. D. Arias-Londoño, “Fake News Detection in Spanish Using Deep Learning Techniques,” Oct. 2021, doi: 10.48550/arxiv.2110.06461.

[15] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, “Fake News Detection: A Deep Learning Approach,” SMU Data Science Review, vol. 1, no. 3, Aug. 2018, Accessed: Dec. 30, 2022. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol1/iss3/10

[16] N. L. S. R. Krishna and M. Adimoolam, “Fake News Detection system using Decision Tree algorithm and compare textual property with Support Vector Machine algorithm,” 2022 International Conference on Business Analytics for Technology and Security, ICBATS 2022, 2022, doi: 10.1109/ICBATS54253.2022.9758999.

[17] M. Renuka and T. P. Anithaashri, “Enhancing the Detection of Fake News in Social Media using Support Vector Machine Algorithms

Comparing over Apriori Algorithms," Proceedings of International Conference on Technological Advancements in Computational Sciences, ICTACS 2022, pp. 51–54, 2022, doi: 10.1109/ICTACS56270.2022.9988701.

[18] A. Vora and N. Shekokar, "Fake News Detection Using Intelligent Techniques," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021, 2021, doi: 10.1109/ICRITO51393.2021.9596438.

[19] A. Jain, A. Shakya, H. Khatter, and A. K. Gupta, "A smart System for Fake News Detection Using Machine Learning," IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, ICICT 2019, Sep. 2019, doi: 10.1109/ICICT46931.2019.8977659.

[20] N. F. Baarir and A. Djeffal, "Fake News detection Using Machine Learning," 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-Being, IHSH 2020, pp. 125–130, Feb. 2021, doi: 10.1109/IHSH51661.2021.9378748.

[21] V. Gupta, R. S. Mathur, T. Bansal, and A. Goyal, "Fake News Detection using Machine Learning," 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, COM-IT-CON 2022, pp. 84–89, 2022, doi: 10.1109/COM-IT-CON54601.2022.9850560.

[22] S. Helmstetter and H. Paulheim, "Weakly supervised learning for fake news detection on Twitter," Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018, pp. 274–277, Oct. 2018, doi: 10.1109/ASONAM.2018.8508520.

[23] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," Proceedings of the Seventh International Conference on Language Resources and Evaluation. 2010. Accessed: Jan. 28, 2023. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

[24] T. S. Reshmi and D. M. Raja S, "Fake News Detection using Voting Ensemble Classifier," 5th International Conference on Inventive Computation Technologies, ICICT 2022 - Proceedings, pp. 1241–1244, 2022, doi: 10.1109/ICICT54344.2022.9850814.

[25] H. E. Wynne and K. T. Swe, "Fake News Detection in Social Media using Two-Layers Ensemble Model," ITC-CSCC 2022 - 37th International Technical Conference on Circuits/Systems, Computers and Communications, pp. 411–414, 2022, doi: 10.1109/ITC-CSCC55581.2022.9894967.

[26] Z. Tian and S. Baskiyar, "Fake News Detection using Machine Learning with Feature Selection," Proceedings of the 2021 6th International Conference on Computing, Communication and Security, ICCCS 2021, 2021, doi: 10.1109/ICCCS51487.2021.9776346.

[27] G. Dozier et al., "GEFeS: Genetic & evolutionary feature selection for periocular biometric recognition," IEEE SSCI 2011 - Symposium Series on Computational Intelligence - CIBIM 2011: 2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management, pp. 152–156, 2011, doi: 10.1109/CIBIM.2011.5949211.

[28] P. J. R. Billones, D. D. MacAsaet, and S. U. Arenas, "Bilingual Fake News Detection Algorithm Using Naïve Bayes and Support Vector Machine Models," Proceedings - 2022 IET International Conference on Engineering Technologies and Applications, IET-ICETA 2022, 2022, doi: 10.1109/IET-ICETA56553.2022.9971596.

[29] A. Khalil, M. Jarrah, M. Aldwairi, and Y. Jararweh, "Detecting Arabic Fake News Using Machine Learning," 2021 2nd International Conference on Intelligent Data Science Technologies and Applications, IDSTA 2021, pp. 171–177, 2021, doi: 10.1109/IDSTA53674.2021.9660811.

[30] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," Neurocomputing, vol. 376, pp. 214–221, Feb. 2020, doi: 10.1016/J.NEUCOM.2019.10.033.

[31] S. A. Alameri and M. Mohd, "Comparison of Fake News Detection using Machine Learning and Deep Learning Techniques," 2021 3rd International Cyber Resilience Conference, CRC 2021, Jan. 2021, doi: 10.1109/CRC50527.2021.9392458.

[32] A. Gupta, V. Mishra, and B. J, "Fake news detection using Ensemble model," International Conference on Interdisciplinary Cyber Physical Systems (ICPS), pp. 34–39, Nov. 2022, doi: 10.1109/ICPS55917.2022.00014.

[33] F. B. Mahmud, M. M. S. Rayhan, M. H. Shuvo, I. Sadia, and M. K. Morol, "A comparative analysis of Graph Neural Networks and commonly used machine learning algorithms on fake news detection," Proceedings - 2022 7th International Conference on Data Science and Machine Learning Applications, CDMA 2022, pp. 97–102, 2022, doi: 10.1109/CDMA54072.2022.00021.

[34] Reyes-Magaña Jorge and Argota-Vega Luis, "Analysis of Text Features Applied to Fake News Detection in Spanish," XXXVII International Conference of the Spanish Society for Natural Language Processing, vol. 2943, 2021, Accessed: Mar. 19, 2023. [Online]. Available: https://ceur-ws.org/Vol-2943/

[35] V. Lomas Barrie, N. Perez, V. M. Lara, and A. Neme, "Fake news detection based on random forests, k-nearest neighbors, and n-grams for a Spanish corpora," XXXVII International Conference of the Spanish Society for Natural Language Processing, 2021.

[36] P. Gamallo, "A Hybrid Strategy for Fake News Detection," XXXVII International Conference of the Spanish Society for Natural Language Processing, 2021.

[37] Z. Guan, "Fake News Detection in Spanish Using Multi-Model Ensemble Learning," XXXVII International Conference of the Spanish Society for Natural Language Processing, 2021, Accessed: Mar. 19, 2023. [Online]. Available: https://sites.google.com/view/toxicspans

[38] K. Zhao, S. Zhou, − 098x, and W. Li, "Fake news detection based on Pre-training Model," XXXVII International Conference of the Spanish Society for Natural Language Processing, 2021.

[39] J. Nielsen, "Website Response Times," Nielsen Norman Group, 2010. https://www.nngroup.com/articles/website-response-times/ (accessed Dec. 26, 2022).

Support Vector Regression based Localization Approach using LoRaWAN

Saeed Ahmed Magsi[1], Mohd Haris Bin Md Khir[2], Illani Bt Mohd Nawi[3], Abdul Saboor[4], Muhammad Aadil Siddiqui[5]
Dept. of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia[1, 2, 3, 5]
High Performance Computing Lab (HPC3), Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia[4]
Faculty of Information and Communication Technology, BUITEMS, Quetta, Pakistan[4]

***Abstract*—The Internet of Things (IoT) domain has experienced significant growth in recent times. There has been extensive research conducted in various areas of IoT, including localization. Localization of Long Range (LoRa) nodes in outdoor environments is an important task for various applications, including asset tracking and precision agriculture. In this research article, a localization approach using Support Vector Regression (SVR) has been implemented to predict the location of the end node using LoRaWAN. The experiments are conducted in the outdoor campus environment. The SVR used the Received Signal Strength Indicator (RSSI) fingerprints to locate the end nodes. The results show that the proposed method can locate the end node with a minimum error of 36.26 meters and a mean error of 171.59 meters.**

Keywords—LoRaWAN; localization; RSSI; fingerprinting; support vector regression

I. Introduction

In 2016, everything appeared to spin around the development of the Internet of Things (IoT), where anything from vehicles to washroom scales is connected to the internet to offer additional services to customers [1]. However, it is most likely that the industry applications evolving from machine-to-machine (M2M) technologies are the main driving force for IoT. IoT is the evolution of M2M communications, where a larger number of nodes are connected using ethernet in the backend to reroute the data as needed.

This is a crucial step towards creating smart city applications and the fourth industrial revolution, where experts say that the physical, digital, and biological boundaries will blur in industries [1]. IoT companies are trying to launch their solution for networks because of the increasing demand for Machine-to-Machine communication. While Machine-to-Machine largely depends upon 2G networks for deployment, the IoT emerged with entirely different requirements, such as low costs for the IoT chips and the dense deployment of nodes on a single cell [2][3].

The deployment of IoT has increased the demand for finding the locations of the end devices. It is crucial in the field of IoT to have the localization done with low power and long range [4][5][6][7]. This can be accomplished by implementing low-power wide area networks (LPWAN) technologies. Long-range wide area networks (LoRaWAN), the LPWAN technology, have the significance of providing location-based services with low power and long range.

Localization using LoRaWAN can be performed using multiple techniques or approaches. The simplest of which is the trilateration technique [8][9]. This technique uses at least three gateways to find the location of the end node. It uses the received signal strength indicator (RSSI) to determine the distance between the gateways and the end nodes. It then applies the trilateration algorithm to find the end node's location. The second approach is to find the angle of the received signal on the receiving antenna. Using that angle, the angle of arrival (AoA) technique helps in finding out the location of the end node [10][11][12]. The third approach can be the time-based approach. Time of Arrival (ToA) [13][14][15][16] and Time Difference of Arrival (TDoA) [17] are the two types of time-based techniques. The time-based approaches use the time of the signal to reach the receiver. This time is then converted to the distance, and the localization is performed. The final and most accurate approach is the fingerprinting approach. This approach has two phases, the offline and the online phase. The measurements are taken in the offline phase and uploaded to the database. In the online phase, the location of the end nodes is predicted using machine learning algorithms by learning the data collected in the offline phase. Multiple machine-learning algorithms can be implemented to find the location of the end nodes. Depending upon the application, the classifiers [18][19][20] and the regression-based algorithms [21][22][23] are used. If it is enough to determine the region or an area where the end node is located, then the classification algorithms can be the easy catch. If there is a need to find the exact location of the end node, then the regression-based algorithms can be used to find the ground truth locations of the end nodes.

In this research article, we have implemented the fingerprinting approach to find the location of the end node. Firstly, the measurements are taken in the campus outdoor environment. Using those measurements, the location of the end node is predicted using support vector regression (SVR) to find the overall distance error. The performance of SVR is studied in the areas where the shadowing effect has its maximum presence.

From the technical problem evaluation perspective, the work is subdivided into the following sections. The literature review is discussed in Section II. In Section III, the research methodology is presented. Section IV presents the results. The conclusion is derived in the last section, along with the references.

II. Literature Review

Long Range Wide Area Network (LoRaWAN) technology is the key enabler of IoT technologies. It helped form a large-scale network connected to the internet at a very low cost because the range of LoRaWAN is high, requiring a minimal number of end devices and gateways to cover a large area. The technology has long-range communication with very little power consumption, thereby increasing the battery life of the end devices. Study [24] used LoRaWAN in the real environment in Thailand to present the experimental performance evaluation. The authors have found experimentally that in an outdoor rural environment, the LoRaWAN ranges up to 2 km and ranges 55-100 m in an indoor environment. It was pointed out that the range depends upon the properties of antennas, such as the antenna's height, gain, and directivity. Research [25] used the central business district of Melbourne, a high-density urban area, to present specific measurements to evaluate the performance of LoRaWAN. Their results show that within the radius of 200 m, only the communication is loss-free, while at around 600 m, the communication is a total loss. It isn't easy to have a precise measurement. Author in [26] explained with the results that environmental temperature highly affects communication. The authors showed that perfect communication could be converted to an almost useless one by increasing the environmental temperature. Therefore, it is important to consider the environment and the effect of the environment on LoRa signals to get good localization accuracy. The LoRaWAN technology offers an excellent option for Internet of Things (IoT) uses, such as advanced agriculture irrigation systems and intelligent urban development initiatives, among others. Thousands of end devices can be supported by a single gateway. Localization is significant for these LoRaWAN applications as the LoRaWAN network can have devices within the range of several thousand. Therefore, it is imperative to estimate each end device's location. An example of this can be multiple temperature sensors placed in various urban areas to measure temperature fluctuations. As the number of sensors can be thousands in this application; therefore, it is very tedious to program each of the sensors with their locations.

A natural solution to this problem is to equip GPS with every sensor. While this is a perfect solution, as GPS can have up to 10 m of accuracy, adding a GPS tracker to every sensor or device will increase the overall cost and power consumption [27]. Another problem with GPS is the lack of indoor coverage, as GPS signals can have so much signals losses when penetrating buildings etc. Therefore, it is very much important to find a solution for localization using LoRa. The in-depth studies on LoRaWAN can be found in [28] [29] [30] [31].

The study [32] calculated the positioning errors by constructing RSSI fingerprint data for LoRaWAN and SigFox using k-NN. The accuracy obtained by the authors was 398.4m. Their study used several gateways for measuring LoRa RSSI data.

Research [33] compared the fingerprinting and the range-based approaches. The authors concluded that the fingerprinting approach has less mean localization error than the range-based approach. The mean error using fingerprinting approach was 340 m, and 700 m using the range-based techniques. Similarly, [34] used k-NN, Extra Trees, and neural networks (NN) to find the location of the end node and had the mean error of 394 m, 379 m, and 357 m, respectively. Authors in [35] used the artificial neural network to find the end node's location and got a mean error of 381.8 m.

The study [36] compared linear regression methods, SVR, k-NN, weighted k-NN, and random forest, concluding that the random forest could perform with the minimum localization error of 340 m.

The research [37] used two layers to perform the localization. In the first layer, the authors used k-means clustering; in the second layer, the final position is estimated using the weighted kernel regression model. The authors were able to achieve a mean localization error of 346.03 m.

The literature provides valuable insights into LoRa localization. However, to the best of the author's knowledge, a significant gap still exists in the literature with regard to the performance of SVR, where the shadowing effect has its maximum presence.

III. Methodology

This section describes the hardware setup, the dataset, test point locations, and the methodology used.

A. *LoRaWAN Setup*

Fig. 1 shows the longitudes and the latitudes of the points where the RSSIs are measured. The gateway was placed on the rooftop of a lab with an elevation of 74 m above sea level, and the end node was moved to 14 random locations. The height of the end device was variable as it is challenging to make the elevation of the end device constant with different distances and areas. A total of 14 random locations were selected on the campus to find out the measured RSSI values. All the measurements were taken outdoors. No indoor measurements were taken. The minimum and the maximum distance used between the gateway and the end node are 17 m and 1330 m, respectively, to measure the RSSI. The distances between the End Node and gateways are calculated using Eq. (2).

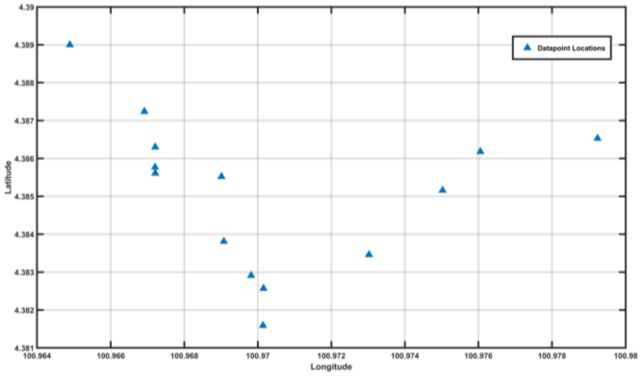


Fig. 1. Longitudes and latitudes of the datapoint location in an outdoor campus environment.

The Dragino LoRa shield served as the endpoint device for the experiments, and the RisingHF (RHF2S008) acted as the gateway. The end node was powered by a portable battery bank, while the gateway was powered through Power over Ethernet (PoE) and included an integrated GPS module, making it convenient for the experiments to determine Differential TDoA. The gateway possessed notable features, which are as under.

- The gateway supported 8 multi spreading factor uplink channels.
- The maximum output power of 27 dBm.
- -141 dBm of Receiver sensitivity.
- It supports ethernet connection, including Wi-Fi, GPRS, 3G, and 4G connections.
- Antenna gain is 3dBi.

The experiments utilized the online public network server called THE THINGS NETWORK as the network server. The LoRa Shield transmits the data to the gateway, which then passes the data, along with metadata such as SNR, RSSI, and timestamps, to THE THINGS NETWORK. The collected data is then uploaded to the computer for the application of Support Vector Regression (SVR) to predict the location of the end node.

B. Support Vector Regression

SVR is the supervised machine learning model that works similar to support vector machine. It finds the best-fit line for the predictions. The Support Vector Regression (SVR) approach differs from other regression models in that it aims to find the best line that falls within a specified range, known as the threshold value, instead of minimizing the difference between the actual and predicted values. This threshold value refers to the space between the hyperplane and the boundary line. However, SVR's computational time for fitting increases rapidly with the number of samples, making it challenging to handle datasets with over 10,000 data points [33].

There are a total of 21 features (RSSIs) for a single ground truth location to predict the location of the end node. We used the standardization in the preprocessing step on our data using z-score and then applied the SVR. The kernel scale used is 1.1. We have used the gaussian as the kernel function. The formula for the gaussian kernel function can be seen in Eq. (1) [38].

$$G(x_j, x_k) = exp(-||x_j - x_k||^2) \qquad (1)$$

Where x_j is the target variable, and x_k is the feature variable.

C. Distance Error

The distance error between the ground truth location and the predicted location is calculated using Eq. (2) by implying the predicted longitudes and the latitudes [39].

$$\boldsymbol{m = 2 * tan^{-1}\left(\frac{\sqrt{sin^2(\frac{l_2-l_1}{2})+cosl_1*cosl_2*sin^2\left(\frac{q_2-q_1}{2}\right)}}{\sqrt{1-\{sin^2(\frac{l_2-l_1}{2})+cosl_1*cosl_2*sin^2\left(\frac{q_2-q_1}{2}\right)\}}}\right)} \qquad (2)$$

Whereas $n=R*m$. The R is the earth's radius, n is the distance between two points on earth, l and q are the latitudes and longitudes, respectively.

IV. Results and Discussions

Fig. 2 shows the RSSI values at different distances. The results were taken by using a single gateway and a single node. There were 14 locations where the RSSIs were measured. Some locations were chosen to make the shadowing effect more significant. At each location, 21 readings were taken.

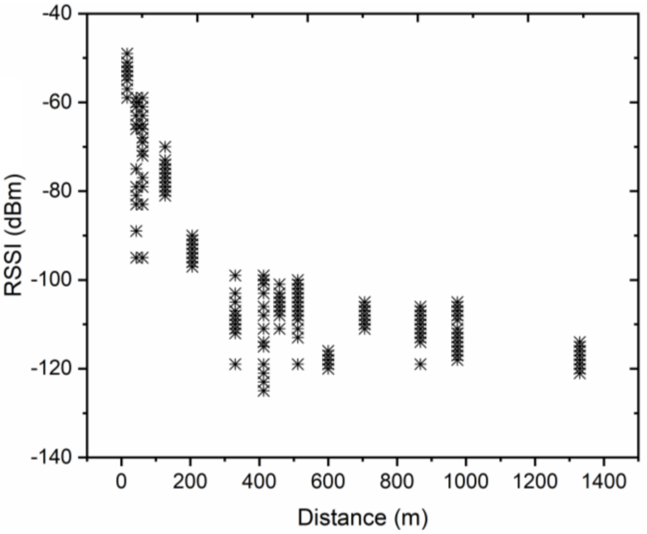


Fig. 2. RSSI values at different distances.

As shown in Fig. 2, the RSSI decreased with the increase in the distance, but there were some exceptions. The exceptions were the locations where the shadowing effect was more significant. This can be seen in Fig. 3 where the average RSSI plot was taken at different locations. As can be seen at the distance of 43m, 330m, 413m and 600m, the average RSSI decreased more because of shadowing, especially at 600m, which we measured behind the building. The RSSI decreased up to -118.4 dBm because of the shadowing effect. It is clear from the graphs that shadowing can be a bottleneck for localization using LoRa, thereby increasing distance error.

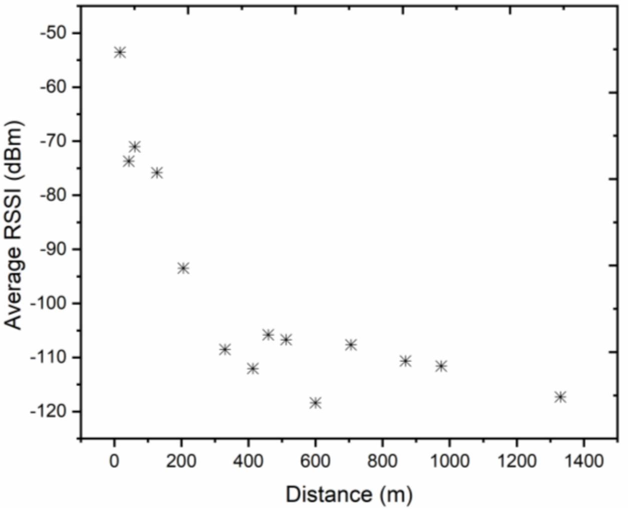


Fig. 3. Average RSSI values at different distances.

Fig. 4 shows the exponential non-linear least squares fitting on the average RSSIs. It is observed that the RSSI decreases with the increase in the distance, but after a certain distance, the decrease in RSSI becomes very minimal.

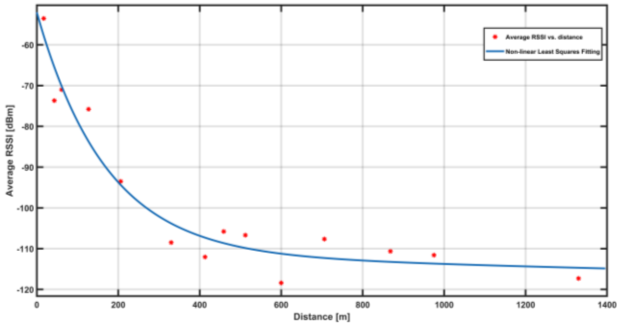


Fig. 4. Exponential non-linear least squares fitting on the average RSSIs.

Fig. 5 shows the SNR values at different distances. It can be observed that there were some locations where the SNR was in negative value because the noise power was more as compared to the signal.

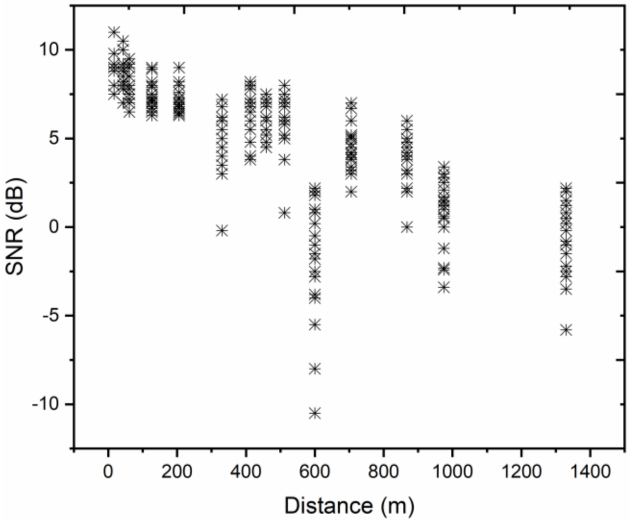


Fig. 5. SNR values at different distances.

Fig. 6 shows the average SNR at every testbed location. It is observed that the SNR measured has a negative value at 600 m (behind the building) and at a distance of 1330 m.

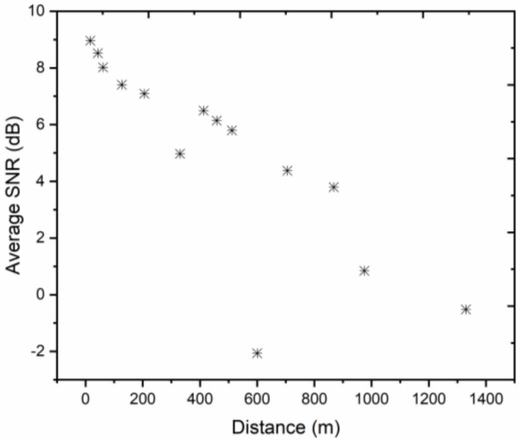


Fig. 6. Average SNR values at different distances.

A combined graph of Average SNR, Average RSSI and distance can be seen in Fig. 7. As can be seen that at a distance of 600 m, the RSSI and SNR are at their lowest value due to the shadowing effect. This is the datapoint directly taken behind the building, which clearly shows that the shadowing directly affects localization accuracy using SVR.

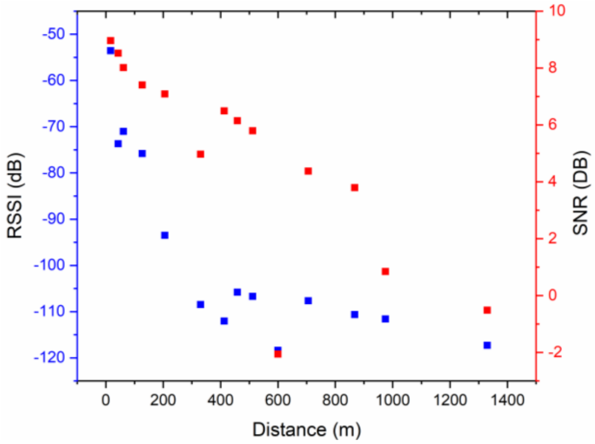


Fig. 7. Distance vs average RSSI and average SNR.

Table I shows the measured average RSSIs, average SNRs, longitude and latitudes of the actual locations, longitude and latitudes of the predicted locations, and the distance errors caused using the SVR. The table shows that the highest distance errors predicted by SVR were the locations where the shadowing effect was most significant. The least distance error predicted by the proposed method is 36.26, the mean error is 171.59 m, and the highest distance error is 755.54 m.

The limitations of the study can be attributed to the environmental factors on the signal strength, which can affect the accuracy of the LoRa localization. Some limitations include signal strength variability due to environmental factors such as obstacles, interference, and atmospheric conditions. The complexity of environmental modeling, such as terrain and buildings, affects the propagation of LoRa signals. Accurately modeling these environmental factors can be complex and require detailed knowledge of the local environment.

One use case scenario where the localization method can still be useful even with a high error is in wildlife tracking. For example, researchers tracking the movement of large animals such as elephants or giraffes in a wildlife reserve can benefit from using localization methods to get a general idea of the animal's location, even with a high error margin. Even if the location error is high, it can still provide valuable information about the animal's general movements, such as where they are likely to feed, rest, or migrate. Additionally, the data collected over time can help researchers identify patterns, make predictions about the animal's behavior, and inform conservation efforts. By tracking the movements of wildlife, researchers can gain insights into their behaviors and habitats and use that knowledge to protect them better.

TABLE I. DISTANCE, AVERAGE RSSI, DISTANCE ERROR AND AVERAGE DISTANCE ERROR IN METERS

Distances between GW and End Nodes (m)	**Average RSSI (dBm)**	**Average SNR (dB)**	**Actual Locations**		**Predicted Locations**		**Distance Error (m)**	**Average Distance Error (m)**
			Longitude	**Latitude**	**Longitude**	**Latitude**		
17	-53.55	8.96	100.96721	4.38630	100.96931	4.38598	236.34	171.59
43	-73.70	8.52	100.96720	4.38577	100.96765	4.38563	53.00	
61	-71.00	8.01	100.96721	4.38561	100.96929	4.38564	230.81	
127	-75.80	7.40	100.96691	4.38724	100.96674	4.38692	39.10	
206	-93.50	7.09	100.96901	4.38552	100.96932	4.38562	36.26	
330	-108.50	4.97	100.96908	4.38381	100.96983	4.38408	89.42	
413	-112.05	6.49	100.96489	4.38900	100.96539	4.38763	161.65	
459	-105.80	6.14	100.96982	4.38291	100.97094	4.38318	127.96	
512	-106.70	5.79	100.97015	4.38257	100.97064	4.38309	80.11	
600	-118.40	-2.06	100.97014	4.38159	100.97058	4.38526	412.46	
706	-107.65	4.37	100.97302	4.38346	100.97291	4.38380	39.97	
868	-110.65	3.79	100.97502	4.38516	100.97448	4.38530	61.57	
975	-111.60	0.84	100.97605	4.38618	100.97536	4.38603	78.06	
1330	-117.30	-0.51	100.97923	4.38653	100.97241	4.38623	755.54	

V. CONCLUSION AND FUTURE WORK

In this research article, the experiments were conducted on a university campus to find the location of the end node. A localization algorithm using Support Vector Regression (SVR) has been implemented on a LoRaWAN architecture. The results show that using the RSSI as features and the SVR as a regression algorithm, the end node can be located with an average distance error of 171.59 meters and a minimum error of 36.26 meters. This shows that the low-powered LoRaWAN can be used for localization in applications where high localization accuracy is not needed. This work can be extended to find the effect on localization accuracy by increasing the number of gateways, the dataset, and the inclusion of other fingerprints like SNR and time fingerprints. Additionally, different environments can be included in the experiments, like indoor areas and finding out the localization distance errors in the combined space of indoors and outdoors.

ACKNOWLEDGMENT

The author of this research article acknowledges the financial support from the Universiti Teknologi PETRONAS under full-time PhD sponsorship for Graduate Assistantship (GA) program.

REFERENCES

[1] K Schwab, The fourth industrial revolution. Crown Publishing Group, New York, USA, 2017. https://books.google.com

[2] E. J. Oughton, et al., "Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6," Telecomm Policy, vol. 45, no. 5, Jun. 2021, doi: 10.1016/J.TELPOL.2021.102127.

[3] F. Montori, L. Bedogni, M. di Felice, and L. Bononi, "Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues," Pervasive Mob Comput, vol. 50, pp. 56–81, Oct. 2018, doi: 10.1016/J.PMCJ.2018.08.002.

[4] O. Dieng, C. Pham, and O. Thiare, "Outdoor Localization and Distance Estimation Based on Dynamic RSSI Measurements in LoRa Networks: Application to Cattle Rustling Prevention," International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 2019-October, Oct. 2019, doi: 10.1109/WIMOB.2019.8923542.

[5] W. T. Sung, S. J. Hsiao, S. Y. Wang, and J. H. Chou, "LoRa-based internet of things secure localization system and application," Conf Proc IEEE Int Conf Syst Man Cybern, vol. 2019-October, pp. 1672–1677, Oct. 2019, doi: 10.1109/SMC.2019.8913875.

[6] C. Bouras, A. Gkamas, and S. A. K. Salgado, "Energy efficient mechanism for LoRa networks," Internet of Things (Netherlands), vol. 13, Mar. 2021, doi: 10.1016/j.iot.2021.100360.

[7] L. Khalil, "LoRa-positioning in Malmö compared with GPS possibilities, power consumption & accuracy LoRa-positionering i Malmö jämfört med GPS möjligheter, strömförbrukning & noggrannhet," 2018.

[8] A. Vazquez-Rodas, F. Astudillo-Salinas, C. Sanchez, B. Arpi, and L. I. Minchala, "Experimental evaluation of RSSI-based positioning system with low-cost LoRa devices," Ad Hoc Networks, vol. 105, Aug. 2020, doi: 10.1016/j.adhoc.2020.102168.

[9] S. A. Magsi, et al., "Experimental evaluation of Trilateration-Based outdoor localization with LoRaWAN," Computers, Materials & Continua, 2023, 75(1): 845–862. doi: 10.32604/cmc.2023.033636.

[10] M Naseri, et al., "Machine Learning-Based Angle of Arrival Estimation for Ultra-Wide Band Radios," IEEE Communications Letters, 2022, 26(6): 1273–1277. doi: 10.1109/LCOMM.2022.3167020.

[11] A. C. Eska, "Determination of MS Location through Building Using AoA Method of Frequency 47 GHz," 2017.

[12] R. Peng and M. L. Sichitiu, "Angle of Arrival Localization for Wireless Sensor Networks," 2006.

[13] S. Wu, S. Zhang, and D. Huang, "A TOA-Based Localization Algorithm with Simultaneous NLOS Mitigation and Synchronization Error Elimination," IEEE Sens Lett, vol. 3, no. 3, Mar. 2019, doi: 10.1109/LSENS.2019.2897924.

[14] L. Xu, J. He, P. Wang, K. Pahlavan, H. Ning, and Q. Wang, "Toward emergency indoor localization: Maximum correntropy criterion based direction estimation algorithm for mobile TOA rotation anchor," IEEE Access, vol. 6, pp. 35867–35878, Jun. 2018, doi: 10.1109/ACCESS.2018.2850967.

[15] S. Pradhan, S. Shin, G.-R. Kwon, J.-Y. Pyun, and S.-S. Hwang, "The Advanced TOA Trilateration Algorithms with Performance Analysis," in IEEE Conference on Signals, Systems and Computers, CA, USA, 2016, pp. 923–928.

[16] K. W. Cheung, H. C. So, W. K. Ma, and Y. T. Chan, "Least Squares Algorithms for Time-of-Arrival-Based Mobile Location," IEEE Transactions on Signal Processing, vol. 52, no. 4, pp. 1121–1128, Apr. 2004, doi: 10.1109/TSP.2004.823465.

[17] J. Pospisil, R. Fujdiak, and K. Mikhaylov, "Investigation of the performance of tdoa-based localization over lorawan in theory and practice," Sensors (Switzerland), 2020, 20(19): 1–22. doi: 10.3390/s20195464.

[18] W. Farjow, A. Chehri, M. Hussein, and X. Fernando, "Support Vector Machines for indoor sensor localization," in 2011 IEEE Wireless Communications and Networking Conference, WCNC 2011, 2011, pp. 779–783. doi: 10.1109/WCNC.2011.5779231.

[19] C. Zhou, J. Liu, M. Sheng, Y. Zheng, and J. Li, "Exploiting Fingerprint Correlation for Fingerprint-Based Indoor Localization: A Deep Learning Based Approach," IEEE Trans Veh Technol, vol. 70, no. 6, pp. 5762–5774, Jun. 2021, doi: 10.1109/TVT.2021.3075539.

[20] D.-H. Kim, A. Farhad, and J.-Y. Pyun, "UWB Positioning System Based on LSTM Classification with Mitigated NLOS Effects," IEEE Internet Things J, pp. 1–1, 2022, doi: 10.1109/JIOT.2022.3209735.

[21] G. M. Mendoza-Silva, A. C. Costa, J. Torres-Sospedra, M. Painho, and J. Huerta, "Environment-Aware Regression for Indoor Localization Based on Wi-Fi Fingerprinting," IEEE Sens J, vol. 22, no. 6, pp. 4978–4988, Mar. 2022, doi: 10.1109/JSEN.2021.3073878.

[22] M. Anjum, M. Abdullah Khan, S. A. Hassan, H. Jung, and K. Dev, "Analysis of time-weighted LoRa-based positioning using machine learning," Comput Commun, vol. 193, pp. 266–278, Sep. 2022, doi: 10.1016/j.comcom.2022.07.010.

[23] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, H. K. Qureshi, and M. Gidlund, "RSSI Fingerprinting-Based Localization Using Machine Learning in LoRa Networks," IEEE Internet of Things Magazine, vol. 3, no. 4, pp. 53–59, Jan. 2021, doi: 10.1109/iotm.0001.2000019.

[24] N. Vatcharatiansakul, P. Tuwanut., and C. Pornavalai, "Experimental performance evaluation of LoRaWAN: A case study in Bangkok," in14th International Joint Conference on Computer Science and Software Engineering (JCSSE), NakhonSiThammarat, Thailand, 2017, pp. 1–4.

[25] P.j. Radcliffe, K. G. Chavez, P. Beckett, J. Spangaro, and C. Jakob, "Usability of LoRaWAN technology in a central business district," in IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 2017, pp. 1–5.

[26] C. A. Boano, M. Cattani, and K. Römer, "Impact of temperature variations on the reliability of lora," in 7th Int. Conf. Sensor Netw., Austria, 2018, pp. 39–50.

[27] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," IEEE Signal Processing magazine, 2005, 22(4): 24–40.

[28] F. Adelantado, et al., "Understanding the limits of LoRaWAN," IEEE Communications magazine, 2017, 55(9): 34–40.

[29] M. Loriot, A. Aljer, and I. Shahrour, "Analysis of the use of LoRaWan technology in a large-scale smart city demonstrator," in Sensors Networks Smart and Emerging Technologies (SENSET), Beiriut, Lebanon, 2017, pp. 1–4.

[30] R. Yasmin, J. Petäjäjärvi, K. Mikhaylov, and A. Pouttu, "Large and dense lorawan deployment to monitor real estate conditions and utilization rate," in IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 2018, pp. 1–6.

[31] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," Sensors, 2016, 16(9), 1466.

[32] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, “Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas,” Data, 2018, 3(2). doi: 10.5281/zenodo.1193562.

[33] T. Janssen, R. Berkvens, and M. Weyn, "Benchmarking RSS-based localization algorithms with LoRaWAN," Internet of Things (Netherlands), 2020, 11. doi: 10.1016/j.iot.2020.100235.

[34] G. G. Anagnostopoulos, and A. Kalousis, "A Reproducible Comparison of RSSI Fingerprinting Localization Methods Using LoRaWAN," in 16th Workshop on Positioning, Navigation and Communications (WPNC), Bremen, Germany, 2019. doi: 10.1109/WPNC47567.2019.8970177.

[35] T. A. Nguyen, "LoRa Localisation in Cities with Neural Networks" Delft University of Technology: Delft, The Netherlands, 2019. https://repository.tudelft.nl.

[36] F. Lemic, et al., "Regression-Based Estimation of Individual Errors in Fingerprinting Localization," IEEE Access, 2019, 7, 33652–33664. doi: 10.1109/ACCESS.2019.2903880.

[37] Y. Li, J. Barthelemy, S. Sun, P. Perez, and B. Moran, "Urban vehicle localization in public LoRaWan network," IEEE Internet Things J, 2021, 1. doi: 10.1109/jiot.2021.3121778.

[38] The Mathworks, Inc. "Understanding support vector machine regression," Accessed: November 07, 2022. [Online]. Available: https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html.

[39] E. Winarno, W. Hadikurniawati, and R. N. Rosso, "Location based service for presence system using haversine method," in Proceedings - 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT, ICITech, Salatiga, Indonesia, 2018. doi: 10.1109/INNOCIT.2017.8319153.

Text-based Sarcasm Detection on Social Networks: A Systematic Review

Amal Alqahtani[1], Lubna Alhenaki[2], Abeer Alsheddi[3]
Computer Science Department, King Khalid University, Abha, Saudi Arabia[1]
Computer Science Department, Majmaah University, Al-Majmaah, Saudi Arabia[2]
Computer Science Department, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia[3]
King Saud University, Riyadh, Saudi Arabia[1, 3]

***Abstract*—Sarcasm is a sophisticated phenomenon used for conveying a meaning that differs from what is being said, and it is usually used to express displeasure or ridicule others. Sentiment analysis is a process of uncovering the subjective information from a text. Detecting figurative language such as irony or sarcasm, is a focused challenging research field of sentiment analysis. Detecting and understanding the use of sarcasm in social networks could provide businesses and politicians with significant insight, since it reflects people's opinions about certain topics, news, and products. This has especially become relevant recently because sarcastic texts have been trending on social networks and are being posted by millions of active users. As a result of this situation, there is now an increasing amount of research on the detection of sarcasm in social network posts. Many works have been published on sarcasm detection, and they include a wide variety of techniques based on rules, lexicons, traditional machine learning, deep learning, and transformers. However, sarcasm detection is a challenging task due to the ambiguity and non-straightforward nature of sarcastic text. In addition, very few reviews have been conducted on the research in this area. Therefore, this systematic review mainly aims at exploring the newly published sarcasm detection articles on social networks in the years between 2019 and 2022. Several databases were extensively searched, and 30 articles that met the criteria were included. The selected articles were reviewed based on their approaches, datasets, and evaluation metrics. The findings emphasized that deep learning is the most commonly used technique for sarcasm detection in recent literature, and Twitter and F-measure are the most used source and performance metric, respectively. Finally, this article presents a brief discussion regarding the challenges in sarcasm detection and future research directions.**

Keywords*—*Sentiment analysis; figurative language; sarcasm detection; irony; machine learning; deep learning; transformer

I. Introduction

Over the last few years, natural language processing (NLP) has been one of the most active areas of artificial intelligence (AI) research. Researchers in this area have made considerable effort to enable machines to mimic the human ability for language, and the results have often been ground-breaking. For example, sentiment analysis, also known as opinion mining, is an NLP task that involves identifying the subjectivity and sentiments present in opinions [1]. Social networks, such as Twitter and Facebook, are gaining increasing popularity and have millions of active users. In particular, Twitter is one of the most popular social networks that attracts millions of users [2]. In addition, text is considered as the most commonly used form of communication, with social network posts varying from short-text data, such as tweets, to long-text posts such as debates.

Sarcasm can be defined as saying or writing the opposite of what is intended. As a result, sarcasm generates ambiguous and non-straightforward data. For instance, "I love to go to the dentist!" is an obvious example of the use of sarcasm for expressing negative feelings. Overall, it is occasionally hard to efficiently recognize sarcasm due to the contradiction between the implicit and explicit meaning [3]. Moreover, textual sarcasm is challenging due to the lack of tone and facial expressions, and this makes it hard for even human beings to detect sarcasm [4]. Therefore, textual sarcasm is a vague task that needs to be studied carefully. A well-designed NLP model for text-based sarcasm detection is, thus, crucial.

Over the past years, a few reviews about sarcasm detection in social networks have been published, but most of them focused mainly on the implementation phase, for example, [5],[6] and [7]. However, some of the previous research did not cover all the approaches used for sarcasm detection. For example, the authors in [5] reviewed and analyzed machine learning-based sarcasm detection studies and found that support vector machine (SVM) is the most frequently utilized classification algorithm for sarcasm detection. However, there are many other techniques in use that need to be studied. The researchers in [7] reviewed the rule-based, statistical-based, and deep learning (DL) approaches for sarcasm detection but did not consider other popular approaches such as transformers, while the researchers in [6] only presented a technical review of sarcasm detection algorithms and reported the mostly frequently used algorithms for sarcasm identification.

Based on the gaps in the literature discussed above, the main aim of this article is to conduct a systematic literature review (SLR) that focuses on identifying and analyzing text-based sarcasm detection articles on social networks based on their development approaches, evaluation metrics, and datasets. Moreover, this article presents an overview of the main sarcasm detection challenges and future possible improvements. To achieve these objectives, the following four research questions will be answered:

- RQ1: What are the main approaches used for the development of automatic sarcasm detection models?

- RQ2: What are the most commonly used metrics to evaluate the performance of sarcasm detection models?
- RQ3: What datasets are most commonly used for detecting sarcasm on social networks?
- RQ4: What are the main challenges in sarcasm detection?

The remainder of this article is organized as follows. Section II provides the problem statement, and Section III describes the methodology used in this SLR. The approaches, metrics, and datasets of the reviewed articles are provided in Sections IV, V, and VI, respectively. Section VII discusses the findings, research problems, and future research directions. Finally, the conclusion is provided in Section VIII.

II. Problem Statement

Over the past decade, the increase in the number of social network users has caused researchers to deeply investigate and analyze data on social networks. Sarcasm detection is one of the most challenging tasks and is a hot topic in the NLP field. Non-straightforward sarcastic data may reflect positive or negative sentiments or both polarities. In fact, it is difficult to detect sarcasm because sarcastic text is often obscure and ambiguous. In other words, there is little agreement on the actual intention behind indirect sarcastic sentences even by humans, and this makes it even harder to accomplish such tasks with AI technology. Most of the text-based sarcasm cannot be interpreted literally since the actual purpose of the sarcastic text might be the opposite of the apparent meaning of the text. Moreover, the lack of body language and voice tone in text-based sarcasm make it difficult to understand sarcasm in text. Another challenge to sarcasm detection is that the context of sarcasm is strongly dependent on cultures, personalities, and languages.

Sarcasm detection is important for tracking people’s opinion and satisfaction in relation to products. Therefore, sarcasm detection is an essential task for decision making by businesses. Social networks, by nature, are rich in sarcastic texts, and this further increases the need for extensive analysis and study. However, applying basic sentiment techniques such as rule-based techniques, with sarcastic text is not sufficient. Therefore, there is a strong need for a well-designed model specifically oriented towards sarcasm detection tasks. The availability of recent review in sarcasm detection field would pave the way for a new novel solution. Therefore, it is crucial to conduct a review that covers the most recent techniques as well as the state-of the art techniques.

Recently, several works have been published on sarcasm detection with machine-learning (ML), DL, and transformer techniques. However, a limited number of the reviews so far have conducted in-depth investigations into sarcasm detection. Therefore, the present SLR comprehensively covers recent articles on text-based sarcasm detection in social networks that were published between 2019 and 2022. In addition, the reviews published so far, that is [8], [9], [10], [11] and [7] have several limitations. For instance, the study in [8] used a different database and selection criteria compared to this study, and the studies in [9] and [10] differ with regard to their research questions. Further, the challenges involved in the development of an effective model for sarcasm detection are not highlighted in [11]. The researchers in [7] did not provide sufficiently detailed characteristics and findings regarding the recent sarcasm datasets and metrics. To sum up, this SLR was conducted with the aim of filling in the highlighted gaps in the previous reviews, as described above. With this survey, our aim is to identify and analyze text-based sarcasm detection articles on social networks based on their development approaches, evaluation metrices, and datasets.

III. Survey Methodology

This SLR uses the Kitchenham guidelines for reviewing articles on sarcasm detection [12]. According to these guidelines, the three stages of a review are planning, conducting, and reporting the review. The following subsections provide the details of these three stages. First, Section A presents the planning stage, including the goals and research questions, database identification and search procedure, and inclusion and exclusion criteria. Second, article selection and quality assessment. Third, from Section IV to Section VI the third stage is reported.

A. Planning

1) Goals and research questions: The primary purpose of this SLR is to identify and analyze articles on the state of the art of sarcasm detection tools based on their development approaches, evaluation metrics, most commonly used datasets, and the major challenges to sarcasm detection identified. To achieve these objectives, the following research questions are investigated:

- RQ1: What are the main approaches used for automatic sarcasm detection models?
- RQ2: What are the commonly used metrics to evaluate the performance of sarcasm detection models?
- RQ3: What datasets are most commonly used for detecting sarcasm on social networks?
- RQ4: What are the main challenges in sarcasm detection?

2) Databases identification and search procedure: Four scientific databases, namely, IEEE, Springer, ScienceDirect, and ACM, were used to search and identify relevant research articles. The search was conducted using nine keywords based on specific selection criteria, which will be described in Section 3. The keywords were selected based on those mentioned in [13],[8] and [9]. Table I presents the number of selected articles based on the keywords and database names.

3) Inclusion and exclusion criteria: The inclusion and exclusion criteria for selecting the most relevant articles were based on the objectives of this SLR. The inclusion criteria were as follows:

a) Articles published in the English language 2. Articles published from 2019 to 2022.

b) Full journal articles.

c) Articles published in the field of computer science.

TABLE I. NUMBER OF SELECTED ARTICLES BASED ON THE KEYWORDS FOR EACH OF THE FOUR DATABASES

Keyword	IEEE	Springer	ScienceDirect	ACM
Sarcasm AND Detection AND Sentiment analysis	11	9	150	83
Sarcasm AND Detection AND Artificial intelligence	12	8	66	48
Sarcasm AND Detection AND machine learning	7	9	155	82
Sarcasm AND Detection AND Deep learning	9	9	139	82
Sarcasm AND Recognition AND Sentiment analysis	2	9	87	83
Sarcasm AND Recognition AND Artificial intelligence	4	8	44	47
Sarcasm AND Recognition AND machine learning	1	9	85	81
Sarcasm AND Recognition AND Deep learning	1	9	84	81
Irony AND Detection AND Sentiment analysis	1	1	92	56
Irony AND Detection AND Artificial intelligence	0	1	47	37
Irony AND Detection AND machine learning	0	1	100	54
Irony AND Detection AND Deep learning	0	1	92	56
Irony AND Recognition AND Sentiment analysis	0	2	62	0
Irony AND Recognition AND Artificial intelligence	2	1	37	41
Irony AND Recognition AND machine learning	2	2	54	0
Irony AND Recognition AND Deep learning	2	0	47	0
Figurative language AND Detection AND Sentiment analysis	4	4	37	12
Figurative language AND Detection AND Artificial intelligence	3	0	16	11
Figurative language AND Detection AND machine learning	1	1	45	12
Figurative language AND Detection AND Deep learning	3	3	44	12
Figurative language AND Recognition AND Sentiment analysis	0	4	17	13
Figurative language AND Recognition AND Artificial intelligence	1	0	14	52
Figurative language AND Recognition AND machine learning	0	2	30	17
Figurative language AND Recognition AND Deep learning	1	3	30	17
Total	67	96	1574	977

A large number of articles met the inclusion criteria, and these were filtered using the following three exclusion criteria.

- Titles and abstracts that were irrelevant to sarcasm detection.
- Duplication.
- Inability of the articles to address the research questions.

As the number of articles retrieved was too large to process manually, it is assumed that the retrieved articles in a database search engine are arranged in accordance with the keywords. According to the first exclusion criterion, articles with titles and abstracts that were not related to sarcasm detection were excluded. Next, duplicate articles that appear in more than one of the databases were excluded. The last criterion relates to whether the articles could address the research questions and involves quality assessment of the candidate articles, as discussed in the following subsection.

4) Article selection: The initial search in the databases returned about 2726 articles. Table I details the number of articles returned for each possible keyword query in all four databases. In general, the maximum number of articles (1574) was retrieved from ScienceDirect database; this is probably due to differences in the content of the databases, interests, and domains. Moreover, the highest number of articles was retrieved with the query “Sarcasm AND Detection AND Machine learning”.

For screening the retrieved articles, the inclusion and exclusion criteria described in the previous subsection are applied. Based on these criteria, 2634 irrelevant articles were excluded, and 92 relevant articles were considered. Following this, 47 duplicated articles were further excluded, and the remaining 45 articles were considered for deeper investigation. Finally, 15 articles that did not address the research questions were excluded, and this left us with 30 articles. Fig. 1 illustrates the article selection process.

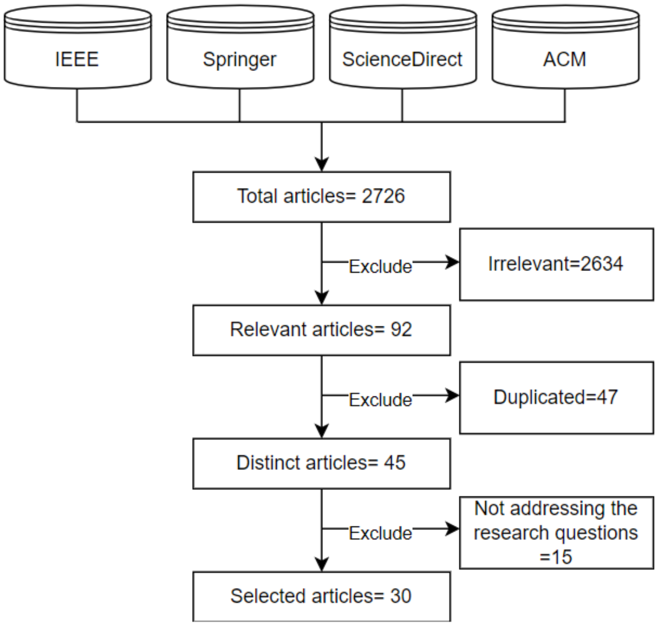


Fig. 1. Flowchart depicting the article selection process.

5) Quality assessment: This section describes quality assessment of the articles based on the method described in [14]. The articles were assessed using the following 10 questions, and articles for which the response was "yes" for at least seven questions were selected.

- Are the article objectives clearly defined?
- Does the article provide a brief description of the previous sarcasm detection approaches?
- Are the evaluation metrices explained clearly?
- Is the article structure designed appropriately?
- Are the data collection processes explained in detail?
- Are the approach, formulation, and analysis described adequately?
- Does the article list the used dataset?
- Is the article understandable and well-written?
- Does the article utilize a well-designed methodology?
- Does the article present and interpret the results clearly?

IV. Sarcasm Detection Approaches and Techniques

There are many studies on NLP methods for sarcasm detection. Recent articles in the field of text-based sarcasm detection on different social networking platforms and online media is surveyed and discussed in this section, but it is not meant to be exhaustive. Sarcasm detection approaches can be categorized based on the classification technique into rule-based, lexicon-based, traditional ML-based, DL-based, and transformer-based approaches. Fig. 2 presents the general structure of sarcasm detection approaches along with their common techniques in the selected articles.

The related works are categorized into five subsections based on the approaches they have explored: Section A focuses on the rule-based approach; Section B, the lexicon-based approach; Section C, traditional ML-based approaches; Section D, DL-based approaches; Section E, the transformer-based approaches. Table II presents a detailed comparison of these works. Overall, traditional ML, DL, and transformer-based approaches are becoming popular in the field of NLP, especially in the area of sarcasm detection. Therefore, in this SLR, studies that focus on these three approaches will be studied in detail.

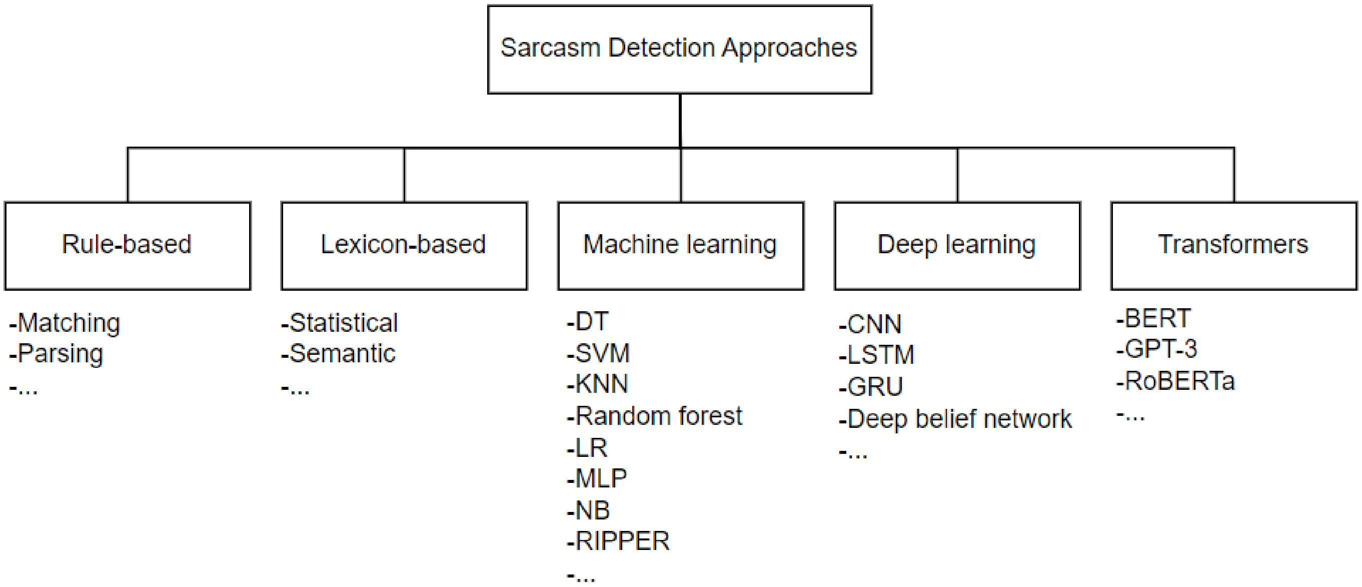


Fig. 2. General structure of sarcasm detection approaches.

TABLE II. SUMMARY OF THE REVIEWED ARTICLES

No.	Article	Model	Year
1	[33]	Combination of a machine learning classifier	2021
2	[31]	Combination of a machine learning classifier	2020
3	[34]	Combination of a machine learning classifier	2021
4	[36]	Combination of a machine learning classifier	2020
5	[32]	Combination of a machine learning classifier	2022
6	[38]	SVM classifier	2022
7	[40]	Bi-LSTM	2019
8	[41]	Att-BiLSTM and convNet deep learning model	2019
9	[42]	MHA-BiLSTM (Multi-Head Attention-based Bidirectional Long Short-term Memory)	2020
10	[43]	CNN	2020
11	[44]	MMNSS (Multi-level Memory Network based on sentiment semantics)	2020
12	[45]	Deep learning approach that consists of an input, embedding, convolutional, Bi-directional Gated Recurrent Unit (BiGRU), and two attention layers	2022
13	[46]	Term-weighted word embedding combined with trigram and 3-layer LSTM	2021
14	[47]	CNN + attention-based BiLSTM	2022
15	[48]	Pre-trained (BERT) for word embeddings + GRU	2021
16	[49]	Multi-task Bi-GRU and attention-based CNN	2021
17	[50]	Deep belief network	2022
18	[51]	HA-LSTM (hybrid attention-based Long Short-Term Memory)	2021
19	[52]	Combination of a machine learning classifier and CNN	2021
20	[53]	BERT, CNN, and LSTM	2022
21	[10]	AWD-LSTM (Averaged Stochastic Gradient Descent Weight-Dropped LSTM)	2022
22	[54]	Attention-based BiGRU	2022
23	[55]	Combination of a machine learning classifier and LSTM	2021
24	[59]	BERT, BiLSTM, and NetXtVLAD	2020
25	[60]	BERT	2021
26	[63]	Bi-LSTM, BERT, and GloVe	2021
27	[66]	BERT	2022
28	[67]	COMET Model	2021
29	[70]	RCNN-RoBERTa	2019
30	[71]	Encoder model called LMTweets with multiple techniques	2021

A. Rule-based Approach

This approach comprises a set of predefined human-made rules that act as indicators of sarcasm. Different researchers have proposed different approaches for making the rules such as parsing and matching. For example, some authors used hashtags as a key indicator of sarcasm. That is, they assumed that if tweets contain specific hashtags and do not fit in with the rest of the tweets, then that statement is sarcasm [15]. Another author combined two rule-based approaches: the first one is used for developing and recognizing the parse tree, and the other approach captures hyperboles features by using interjection and intensifiers together [16]. A third rule-based approach is “simile,” which involves comparing two things directly. One of the studies that utilized this approach for sarcasm detection was described in [17].

B. Lexicon-based Approach

Lexicon-based approaches rely on a predefined collection of words, referred to as a lexicon, with each of the words assigned to a particular polarity category indicating its nature, namely, positive, natural, negative, which are represented by the numerical values -1, 0, and +1, respectively. The lexicon can be weighted or unweighted, such that the words which induce higher positivity or negativity are given a higher probability [1]. In this sarcasm detection process, a bags-of-lexicon which comprises a positive sentiment, a negative sentiment, a positive context, and a negative context is created. A text is divided into tokens of a single word, and the score of each token is obtained using the lexicon. The overall score of the text is determined by adding the individual scores and calculating the average, which is used to determine the sentiment of the text [18]. Sarcasm is detected when a positive context comprises a negative sentiment or a negative context comprise a positive sentiment [16]. An advantage of the lexicon-based approach is that it is suitable at both the sentence and feature level. Moreover, it can be considered as an unsupervised approach because it does not include a training process. However, a major limitation is that it is domain dependent, as the same word would have different meanings according to its context. For example, the word “small” in the statement “this camera is extremely small” could imply a positive sentiment, whereas the use of “small” in “the TV screen is too small” implies a negative sentiment. This could be overcome by constructing a domain-specific lexicon or adapting the current lexicons [19]. In addition, lexicon-based approaches can be divided into the corpus-based approach and the dictionary-based approach.

1) Corpus-based approaches: The corpus-based approach starts with a pre-defined list of polar words with their orientation; their syntactic and co-occurrence pattern is then investigated to obtain other polar words and their corresponding orientation to obtain a bigger “corpus”. This approach was first proposed in [20]: a list of adjectives (polar words) with their orientation were pre-defined, and new adjectives and orientations were added using linguistic constraints and rules. For example, in the sentence “the question is simple and easy,” there is a connective word “AND” which indicates that both adjectives have the same orientation; in contrast, the connective word “OR” indicates that the adjectives have opposite orientations. This approach is known as “sentiment consistency”.

There are two approaches to determining the orientation of polar words, namely, the statistical approach and the semantic approach [18]. The statistical approach relies on the notion that words with similar orientation are likely to appear together frequently. Hence, the new unknown word can be assigned a certain orientation based on its frequency and co-occurrence with other words for which the orientation is known [21]. Some studies on the statistical approach have been published, such as [22] and [23]. The semantic approach, on the other hand, exploits the sentiment dictionary to discover synonyms and antonyms in order to construct a lexicon that can be used to assign the same orientation to words that are semantically similar [24]. Some studies have utilized the semantic approach to build the lexicon, such as [25] and [26]. In addition, a hybrid method can be used to take advantage of both approaches, as described in the work of Zhang [27].

2) Dictionary-based approaches: The dictionary-based approach is roughly based on the idea that synonymous words have the same orientation, and antonyms have the opposite orientation. Therefore, an initial well-known dictionary, such as Thesauri, is constructed with a pre-defined lists of polar words and their orientation. Then, this dictionary is expanded manually based on synonyms and antonyms of the existing words by adding new words and their orientation iteratively until no more words can be added [28]. Finally, manual evaluation and correction can be performed to ensure the validity of the dictionary. This is known as the bootstrapping technique. A popular recently developed dictionary is SentiWordNet 3.0, which uses the automatic annotation of Synsets of WordNet 3 [29]. In addition, Park and Kim in [30] proposed a rule-based method to label the words in advertisements based on three online dictionaries.

C. Traditional ML-based Approaches

Since the earlier years, many studies on text sarcasm detection utilized supervised ML classifiers. Based on the surveyed studies, SVM is one of the most popular classifiers, as evident in [31], [32] and [33].

In 2020, researchers in [31] proposed a sarcasm type detection approach that utilized the multi-rule based ensemble feature selection model. The main aim of this study was to determine the level of hurt that is expressed in sarcasm. Four classes of sarcasm type were determined, including rude, raging, polite, and deadpan. This study used ensemble learning to identify the optimal feature set among all the features and to classify a tweet as sarcastic or not. Following this, the type of sarcasm was determined by using a rule-based approach. This experiment was conducted by using tweets obtained through the Twitter Application Programming Interface (API) Tweepy and Twython. A study conducted in 2021 [33] developed three kinds of ensemble classification algorithms for detecting sarcasm with the Principal Component Analysis (PCA) algorithm. The ensemble classification algorithm is a combination of SVM, KNN, decision tree, logistic regression, and Multi-layer Perceptron (MLP). The three models were tested on five datasets of different sizes from the Twitter streaming API.

Another related study [34] used different ML techniques, such as SVM and logistic regression, for classification. The main contribution was combining the features extracted from a Convolutional Neural Network (CNN) architecture with contextual handcrafted features to obtain the most optimal features. The experiments were conducted on a Twitter dataset created by the researchers and shared publicly [35]. One of the studies that utilized the supervised ML classifier approach with BERT and GloVe embeddings for sarcasm identification [36] also used a Twitter dataset for evaluation. A related study [32] investigated tweets with a negative mood and hyperboles to detect sarcasm. Several ML algorithms, such as SVM, random forest (RF), and RF with bagging, were utilized to analyze five hyperbole features, namely, interjection, intensifier, capital letter, punctuation mark, and elongated word. This study was conducted on tweets collected using the Twitter streaming API [37].

In 2022, the researchers in [38] proposed an intelligent ML-based sarcasm detection and classification (IMLB-SDC) technique in which an SVM classifier is used for sarcasm identification on social networks. The proposed model consists of different stages, namely, preprocessing, feature engineering, feature selection and classification, and parameter tuning.

D. DL-based Approaches

DL is gaining more attention in the sarcasm detection process, since it can be used to obtain better results from unstructured data. It has the ability to learn from a given text in order to either extract automated features or perform sarcasm classification. Based on our investigations, most sarcasm detection articles combine several DL techniques in a model. The most frequently used DL approaches are CNN, artificial neural network, and long short-term memory (LSTM). These are described below.

CNN is a version of the feed forward neural network with multiple hidden layers. It first emerged in computer vision applications, and since then, it has been widely used recently in NLP applications. The network comprises an input layer, hidden layers that consist of many convolution layers, pooling layers, normalization layers, a fully connected layer, and an output layer. The generic workflow of CNN in sarcasm detection is as follows: The convolution layer extracts the features from the input text (word embedding); the pooling layer reduces the size of the feature by removing the noise and un-needed details; the output of the previous layer is plugged to the normalization layer to normalize the input for the current layer in order to aid convergence; finally, a fully connected network is created and used for classification [18]. However, these steps are not identical for all studies. According to our investigations, most studies combined CNN with other DL algorithms such as recurrent neural network (RNN).

RNN is designed for sequence data and has the ability to remember the needed information. Therefore, it has been widely applied in sentiment analysis and sarcasm detection. The output of such networks depends on all previous computations. In other words, to predict the class of a specific word, the model may use the class of previous words and their relations. However, one of the most serious problems with this technique is gradient vanishing. To tackle this problem, Hochreiter and Schmidhuber [39] introduced LSTM and utilized it for sarcasm classification. Later, a new bidirectional version of LSTM (Bi-LSTM) was introduced. Bi-LSTM has the ability to learn from the relationships between the polar words and classify them without relying on an external lexicon. Such an approach has been found to produce better results in many studies. Another important feature is the attention layer [40], which gives the model the ability to focus on words that contribute more to sarcasm classification.

In [40], the researchers developed an attention-based Bi-LSTM model based on features learned by external pre-defined sentiment lexica, thus eliminating the need for the traditional feature vector and increasing the ability of the model to detect incongruity in sarcastic sentences. The researchers in [41] designed a hybrid system that coupled a soft attention-based Bi-LSTM with a CNN. The attention layer generates a feature vector according to which higher weights are assigned to words that are closely related to the sentence semantics. Consequently, this feature vector with pragmatic features is input in the CNN to generate the final classification. The study aimed to improve the performance in terms of accuracy, recall precision and F-measure. Another study [42] developed an attention-based Bi-LSTM model for sarcasm classification. In this model, the multi-head attention layer consists of five heads. The multiple heads allow the attention layer to move among several disjointed information spaces that reflect different representations. They used SVM for handcrafted feature extraction to be used as input for the proposed model. Another work in [43] utilized an attention-based Bi-LSTM for sarcasm classification. For better word embedding, a question answering network was designed based on five different layers, each of which provides different representations. In [44], an improved attention-based multilevel LSTM model was developed to exploit sentiment semantics in sarcasm detection. The semantic is extracted using the first-level attention-based LSTM network. Then, the sentiment semantic features obtained from the first level are used as the input for the second level. In the second level, the polarity between the sentiment semantic features and all the words in the sentence is captured to detect sarcasm by combining the LSTM and CNN networks. Later, a more complex framework was proposed in [45], in which the researchers proposed a Self-Deprecating Sarcasm (SDS) framework that incorporates GloVe embedding, CNN to extract features, bidirectional gated recurrent unit (BiGRU) to extract context information that would be useful for SDS classification, and two attention layers to assign higher weights to SDS-identified sarcastic words.

Another effective sarcasm identification system was engineered in [46] using the Bi-LSTM framework based on two main phases. In the first phase, weighted word embedding was combined with the trigram model for better word representation. In the second phase, the first phase output was inserted into a Bi-LSTM network. A novel approach was suggested in [47], in which sarcasm detection involved the sentiment of the reply to the sarcasm and the user’s expression habit. In this approach, a dual-channel CNN was utilized for sarcasm detection and sentiment analysis of the reply. Moreover, attention-based LSTM was exploited to identify the user’s expression habit. In a subsequent study [48], the researchers proposed a multi-head self-attention-based GRU model to detect sarcasm while considering automatic, lexical, contextual, and handcrafted features. Feature embedding was performed by a pretrained model and was enhanced using the multi-head self-attention layers to identify keywords that contribute more to classification. In [49], the researchers proposed a novel multi-task system for joint sarcasm and sentiment analysis. The local features are obtained using BiGRU, and the global features are obtained by attention-based CNN. In [50], the researchers proposed a novel feature selection approach with deep belief for detection of cyberbullying on social networks. Additionally, the Salp Swarm Algorithm was exploited to tune the network parameter for better classification accuracy. In a subsequent study [51], an attention-based LSTM sarcasm detection model was proposed to combine both hand-crafted features that are usually extracted from classical ML algorithm, such as verbs, nouns, and adjectives, with automatic features that are extracted by DL approaches. That is, the attention layer is utilized to assign weights to the words according to their level of contribution to sarcasm detection. Moreover, 16 different textual classical features are extracted and combined with the automatic features generated from the attention layer. The main contribution in this study was the proposed feature engineering approach.

To capture the variation in the performance of different classification techniques , the researchers in [52] applied five different ML algorithms, namely, Naïve Bayes, KNN, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), C4.5 Decision Tree, and SVM. Moreover, a CNN network was implemented. Additionally, different pre-processing methods were applied with the classifier to obtain the best results. In fact, a pre-trained model can be used for data preprocessing, as described in the approach in [53]. The BERT model is used for data preprocessing by converting the text into distinct tokens, and the tokens are further processed by four CNN layers. The output of this process is plugged into the LSTM layer for classification. In [10], the researchers proposed a system that combines the classical ML approach to extract different text patterns with sarcasm detection using the LSTM classifier. The basic pre-processing steps are performed on the original text before the classification. Further, in [54], the researchers proposed a new attention-based BiGRU for detecting sarcasm in which hyper parameter tuning is performed using an artificial flora algorithm and embedding is performed by the GloVe model.

Very few works have utilized an ensemble of ML and DL approaches. One such study [55] proposed the use of a DL model in combination with an ML classifier to extract the target of sarcasm from the text. The researchers started by using an ensemble of classifiers consisting of RF, SVM, and logistic regression to classify sarcastic sentences and determine whether they contain a target. On the other hand, an LSTM is used to extract the target using aspect-based sentiment analysis.

E. *Transformer-based Approaches*

The sequence-to-sequence (seq2seq) model is used for many purposes, one of which is language translation, for example, translating Chinese into English [56]. One of the main disadvantages of the Seq2Seq model is that it cannot be applied to long sentences or perform parallelization. The main solution for this limitation was proposed in December 2017 in an article titled “Attention Is All You Need,” which described a model called the “original transformer model” that laid the basis for transformer-based approaches [57]. In the field of NLP, a transformer can be described as a novel architecture that can solve Seq2Seq tasks while handling long-range dependencies. In addition, transformer models are trained on large-scale corpora to learn universal language representations, so the need to train a new model from scratch is eliminated [58].

Most recent studies are based on transformer models that exhibited strong performance in sarcasm detection [59], [60]. These architecture models are frequently based on transformer models such as Bidirectional Encoder Representations from Transformers (BERT) and OpenAI Generative Pre-Training-3 Model (GPT-3) [61], [62]. Recently, many researchers have been focusing on transformer models: for example, in 2021, the authors in [63] developed a context-based feature technique to detect sarcasm based on the DL model, BERT model, and conventional ML model. Two Twitter benchmark datasets, one provided by Riloff and one by Ghosh and Veale, were utilized [64], [65]; in addition, the Internet Argument Corpus (IAC-v2) benchmark was also applied. A related study [60] proposed an enhancement to BERT in order to improve its ability to handle the volume, velocity, and veracity of data.

Similarly, in 2022, the researchers in [66] introduced an enhancement to the BERT model by fine-tuning it to related intermediate tasks before applying it to the target task. The authors in [67] applied the pre-trained COMET model to generate relevant commonsense knowledge. The experiment was conducted on three datasets, including Ghosh and Ptácek from Twitter and SARC-Pol from Reddit [35], [65], and [68]. The researchers in [59] proposed a model called Contextual Response Augmentation (CRA) which uses of BERT, BiLSTM, and NetXtVLAD. The dataset consisted of Twitter and Reddit posts. To evaluate the proposed model, the IAC-V12 and AC-V23 datasets [69] and two datasets collected by Riloff et al. [64] and Ptáček et al. [35] were used. Furthermore, two datasets from Reddit [68] were utilized.

Another study in [70] developed an RCNN-RoBERTa model to tackle figurative language in social networks. This model consists of a pre-trained RoBERTa model combined with a recurrent CNN. The Semantic Evaluation Workshop Task 3 (SemEval-2018) dataset was used to measure the performance of the proposed model. Another researcher [71]proposed an encoder model called LMTweets, which is an ensemble of multiple types of techniques. Five classical classifiers, six DL algorithms, and transformer models were utilized for classification in this model. The experiments were conducted on three datasets, namely, Twitter SemEval-2018-Task, Self-Annotated Reddit Corpus (SARC), and Riloff Sarcastic Dataset [72], [64], [68].

V. Evaluation Metrics

One of the most significant aspects of most articles on models for sarcasm detection is performance evaluation, because the results provide an indication of the significance of a study. In this section, the common evaluation metrics used to assess sarcasm detection in the selected articles will be discussed. Confusion matrix is used for analyzing the performance of a binary-class model by depicting the relationship between the actual class and the predicted class. In this matrix, each row contains information about an actual class, while each column contains information about a predicted class. Accordingly, the confusion matrix aims to analyze how well a classification can recognize instances of different classes. Table III illustrates the confusion matrix [73].

TABLE III. Confusion Matrix

Class	Predicted Positive	Predicted Negative
Positive	TP	FN
Negative	FP	TN

In the sarcasm detection problem, true positives (TPs) are considered as sarcastic tweets that are correctly classified as sarcastic text, and true negatives (TNs) are tweets which are not sarcastic that are correctly classified as not being sarcastic (i.e., these refer to correct decisions, which are represented by the diagonal in the confusion matrix). In contrast, false positives (FPs) are instances which are not sarcastic that are misclassified as sarcastic text, and false negatives (FNs) are sarcastic tweets which are misclassified as text that is not sarcastic. The following subsection describes the most common and significant metrics for evaluation with the confusion matrix.

A. *Accuracy*

Accuracy is a common external measurement that reflects the percentage of the total number of tweets that are correctly classified as sarcastic or not sarcastic. It is calculated using the following equation, in which the denominator represents the total number of sarcastic tweets.

$$\text{Accuracy}=\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}\times 100 \qquad (1)$$

TABLE IV. SUMMARY OF THE EVALUATION METRICS AND PERFORMANCE OF THE REVIEWED ARTICLES

No.	Articles	Accuracy	F1-Score	Precision	Recall	AUC
1	[33]	99.1	89.0	90.0	89.0	-
2	[31]	92.7	95.5	93.0	98.3	-
3	[34]	94.0	94.0	95.0	94.0	-
4	[36]	-	69.0	-	-	-
5	[32]	77.3	69.0	69.0	69.0	-
6	[38]	-	94.9	94.7	95.2	-
7	[40]	95.3	99.0	-	-	-
8	[41]	97.9	93.5	92.1	96.8	-
9	[42]	-	77.4	72.6	83.0	-
10	[43]	-	70.8	68.9	72.8	-
11	[44]	-	87.1	85.7	89.2	-
12	[45]	93.0	94.0	92.0	98.0	-
13	[46]	95.3	-	-	-	-
14	[47]	73.0	76.0	-	-	-
15	[48]	-	98.7	97.9	99.6	99.6
16	[49]	92.2	-	91.6	92.0	-
17	[50]	99.0	94.0	-	-	-
18	[51]	-	99.0	99.0	99.0	-
19	[52]	-	66.0	-	-	-
20	[53]	99.6	99.5	99.3	99.8	-
21	[10]	-	82.3	89.3	76.4	72.2
22	[54]	96.8	97.0	97.2	97.2	-
23	[55]	21.7	54.9	-	-	-
24	[59]	-	93.1	93.2	93.6	-
25	[60]	70.6	70.5	68.7	72.5	-
26	[63]	99.0	99.0	98.0	99.5	-
27	[66]	-	97.4	-	-	-
28	[67]	-	85.4	85.7	86.1	-
29	[70]	91.0	90.0	90.0	90.0	94.0
30	[71]	75.0	74.0	73.0	85.0	76.0

B. F1-Score

F1-score is a combination of precision and recall measures, which are the most frequently used metrics. Indeed, to calculate F1-score, precision and recall need to be calculated using the equations (2) and (3).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100 \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100 \quad (3)$$

As mentioned before, F1-score is calculated as a harmonic mean of precision and recall, as demonstrated in the equation below.

$$F(i,j) = \frac{(2 \times Precision \times Recall)}{Precision + Recall} \times 100 \quad (4)$$

In general, F1-score values are within the interval [0, 1]; therefore, the higher the F1-score value, the better is the classification. Table IV presents a summary and comparison of the evaluation metrics used in sarcasm detection in the selected articles. The table shows that more than four types of evaluation metrics have been applied to evaluate sarcasm detection. From the table, it can be observed that the most common measures are F1-score, precision, and recall, and they are followed by accuracy. The results in this table are based on the highest results reported by studies that used multiple algorithms or multiple datasets.

VI. DATASET COLLECTION

Dataset collection is a crucial step in the sarcasm classification process that can affect the entire procedure. Building and annotating datasets for sarcasm detection is a challenging task even for human annotators, since the sarcastic text could be implicit, ambiguous, and hard to identify [74], [75]. It is normal for disagreements between annotators regarding the classification of a single text as sarcastic or not, so the task is even harder for an AI program. This section describes the datasets that were used in the reviewed literature. Noticeably, some articles utilized datasets that were used in the reviewed literature. Also, some articles utilized datasets from multiple sources, including social networks, news headlines, sarcastic reviews on online shops, books snippets, and forums, to stress on the generalization of their systems. For instance, the researchers in [55] utilized three different datasets, including book snippets, tweets, and Reddit comments. However, other articles relied on a single source for the dataset, for example, social network posts [52].

Social network posts have limited length; for example, Twitter limits tweets to 280 characters. This makes it simpler to obtain annotated text based on hashtags and API. The monthly number of active users on Twitter is about 330 million, which makes it a rich source of sarcastic tweets [11]. Therefore, most of the reviewed studies rely on Twitter as a source for their datasets, and few articles used datasets from Reddit and other sources. Twitter-based datasets can be built automatically using the Twitter streaming API while searching for a specific hashtag, such as "#Irony" and "#sarcasm" [32]. In this case, the annotation process is guided by the hashtag itself. In addition, it is already accurate to some extent, since the author clearly declares the sarcasm in the tweet. Another process for collecting datasets is manual self-annotation. For instance, in [60], the annotation process was undertaken by three linguistic annotators, each of whom worked on a subset of the dataset, and in [32], four expert annotators participated in the dataset annotation. In [52], the annotation was manually performed by three students. To ensure the reliability of the self-annotation, an evaluation step can be performed later by a third annotator [60].

The number of the collected instances of sarcasm in the considered datasets varied from 1264 to 1055277 [49], [76]. Generally, the higher the number of tweets in the dataset, the higher is the effectiveness of the proposed models. Some studies used a public dataset, such as [44], while other articles collected data on their own, such as [10]. When a public dataset is used for evaluation, it allows for fairer and more meaningful comparison with other works that use the same dataset.

The number of sarcastic and non-sarcastic samples in the dataset obviously affect the performance of the detection model. An imbalanced dataset may skew the performance of the classification model. In general, the models developed using imbalanced datasets are likely to achieve greater accuracy than other models with conflicting F1-score values [77]. For example, the Riloff dataset [64] creates a bias toward non-sarcastic tweets as it consists of 1648 non-sarcastic tweets and 308 sarcastic tweets. A detailed description of the datasets in the reviewed articles is presented in Table V.

TABLE V. DESCRIPTION OF THE DATASETS USED IN THE REVIEWED ARTICLES

No.	Article	Dataset source	Accessibility	#Instances	Annotation	#Sources	Balance
1	[33]	Twitter	Private	NA	Hashtag	Single	NA
2	[31]	Twitter	Private	76,799	Both	Single	NA
3	[34]	Twitter	Public	780,000	Hashtag	Single	No
4	[36]	Twitter	Private	5000	NA	Single	Yes
5	[32]	Twitter	Private	6600	Both	Single	Yes
6	[38]	Others	Public	28501	Self-annotated	Single	Yes
7	[40]	7 datasets from Twitter	Public	12162	Both	Multiple	4 yes, 3 no
8	[41]	Twitter1, Twitter2	Public/Private	55961	Self-annotated	Multiple	Yes, no
9	[42]	Reddit	Public	6534	Self-annotated	Single	Yes
10	[43]	Others	Public	4692	Self-annotated	Single	Yes
11	[44]	Others, others, Twitter	Public	55795	NA	Multiple	NA
12	[45]	7 datasets from Twitter	Public	134407	Both	Multiple	Yes
13	[46]	Twitter, others, others	Public	40000	Self-annotated	Multiple	No, yes
14	[47]	Reddit, Twitter	Public	45301	Both	Multiple	NA
15	[48]	Twitter1, others, Twitter2, Reddit, others	Public	309566	Both	Multiple	4 yes, 1 no
16	[49]	Others, Twitter	Public/Private	1264	Hashtag	Multiple	No, yes
17	[50]	NA	Private	NA	NA	Single	NA
18	[51]	Twitter1, Twitter2, others	Public	83596	Both	Multiple	2 yes, 1 no
19	[52]	Twitter	Public	4618	Self-annotated	Single	Yes
20	[53]	Others, others	Public	55328	Self-annotated	Multiple	Yes
21	[10]	Tweets, Reddit, Others	Private	20000	Self-annotated	Multiple	Yes
22	[54]	Others	Public	28,501	Self-annotated	Single	Yes
23	[55]	Twitter, Reddit, others	Public	1680	Self-annotated	Multiple	NA
24	[59]	Twitter, Reddit	Private	13000	NA	Multiple	NA
25	[60]	Twitter	Public	3000	Self-annotated	Single	NA
26	[63]	Twitter1, Twitter2, others	Public	58436	Hashtag	Multiple	No, yes, yes
27	[66]	Others, Reddit, Twitter	Public	1018291	Self-annotated	Multiple	Yes, yes, no
28	[67]	Twitter1, Twitter2, Reddit	Public	65551	Self-annotated	Multiple	NA
29	[70]	Twitter, Twitter, Reddit, Twitter	Public	NA	NA	Multiple	NA
30	[71]	Twitter, Reddit	Public	47115	Self-annotated	Multiple	No

VII. Discussion

This SLR analyzed 30 articles that were able to address the four research questions. This section discusses the findings of the review, highlights the challenges, and provides future research directions that can help in the development of more accurate and efficient sarcasm detection tools.

A. Findings

In several domains, NLP is an increasingly important topic with regard to AI and its applications. The research community is paying close attention to the sarcasm detection approaches, datasets and metrics. This subsection focuses on several observations from examination of different aspects of sarcasm detection.

1) Approaches: In general, it is impossible to compare the different approaches objectively due to several variations in the dataset sources and task requirements. One of the most interesting findings, as shown in Fig. 3, is that more than half of the reviewed articles used DL as a classification method for sarcasm detection, and there was a noticeable upward trend in the application of DL techniques for solving several NLP problems. In fact, DL has proved its superiority in sentiment analysis, in general, and in sarcasm detection in particular. One possible reason for this is that the automated feature extraction aspect is more effective and gives better insights about the target text than handcrafted features used in other classical sarcasm detection techniques. There are, however, other possible explanations. For instance, with regard to model performance, it is found that the best accuracy for the reviewed articles was obtained with DL models. Moreover, specific DL techniques, such as RNN, are particularly designed for sequence input data, and this fits the requirements of sarcasm detection tasks.

In addition, as depicted in Fig. 3, an interesting observation was that most articles used hybrid approaches in order to exploit the advantages of more than one approaches. The hybrid approach is extremely important in the development of sarcasm detection tools, as demonstrated in several articles in Section IV. Moreover, classical ML algorithms were utilized by 16% of the researchers. In contrast, only a few of the reviewed articles utilized transformer-based approaches. This is probably because transformers are a relatively new invention for application to sarcasm detection models. However, the rapid improvement in computational resources and increase in the available datasets have led to an increase in the application of transformer-based approaches in recent times.

Fig. 4 depicts the frequency at which various sarcasm detection techniques were used in the reviewed articles in this SLR. Among the classical ML approaches, the most commonly used classifier is SVM. Moreover, for DL approaches, the most commonly used technique is Bi-LSTM, and for transformers, the most applied technique is BERT. To sum up, the most frequently utilized sarcasm detection approach is DL. Moreover, the transformer approach appears to be an emerging promising solution with comparative performance to currently popular techniques and it warrants further investigation.

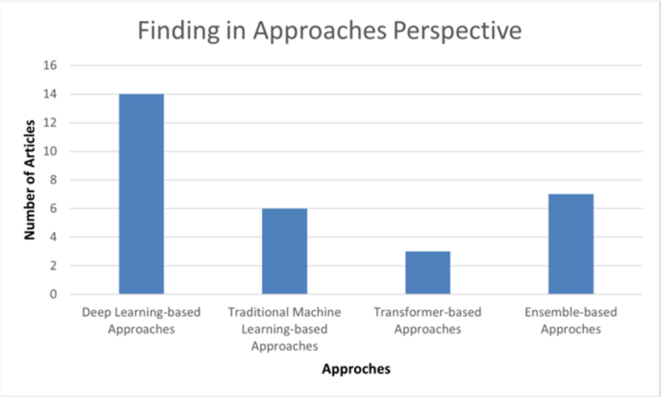


Fig. 3. Trends in the sarcasm detection approaches used by the reviewed articles.

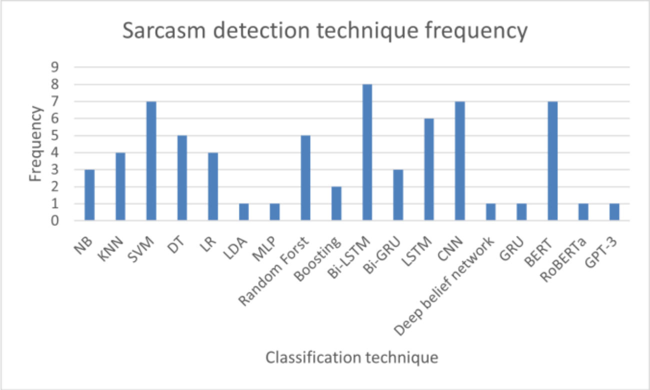


Fig. 4. Classification techniques used for sarcasm detection in the reviewed articles.

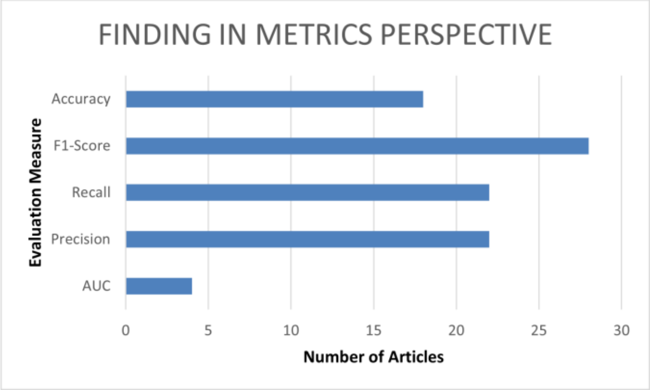


Fig. 5. Frequency of the use of various metrics for evaluation of sarcasm detection models in the included articles.

2) Metrics: As discussed in Section V, researchers used precision, accuracy, recall, F1-score, and AUC as evaluation metrics. As shown in Fig. 5, one of the most significant findings from this SLR is that the majority of researchers utilized F-score, followed by precision and recall. Furthermore, the most obvious finding to emerge from the analysis is that 10% of the reviewed articles used AUC as the evaluation metric. In addition, from the data in Fig. 5, it is apparent that accuracy was used as a metric by 63% of the researchers.

Overall, none of the evaluation metrics fit all sarcasm detection problems due to differences in the characteristics of datasets and approaches used. It is not surprising that F1-score was the most frequently used metric (90% of the researchers used this metric). This is probably because the F-score can balance the precision and recall of the positive class. Moreover, the F1-score could be more suitable than other measures when the target classes are unevenly distributed. Another interesting observation was the correlation between accuracy and dataset balance in the reviewed articles, since the vast majority of datasets were balanced datasets. This may explain why the use of accuracy as an evaluation metric was as high as 63% in the reviewed articles. AUC was the least frequently used metric; this is probably because AUC is based only on the thresholds of the true positive rate and the false positive rate. This is in contrast to the F1-socre, which takes into account the overall recall and precision values. In general, 87% of the observed studies used more than two metrics, and this makes the evaluation framework more robust.

3) Datasets: The dataset sources, number of datasets, dataset accessibility, number of instances, annotation methods, and dataset types of the included articles are discussed here.

An essential factor that affects the sarcasm detection process is the source of the dataset, as shown in Fig. 6. The findings showed that 34% of the analyzed articles used Twitter as a unique source of datasets. One possible reason for this is the huge number of Twitter users, which is 330 million monthly active users [11]. Moreover, Twitter provides concise text that can be automatically annotated by hashtags, and this facilitates dataset building. However, no single public dataset was used across all the reviewed articles.

The most obvious finding to emerge from the analysis is that 50% of the reviewed articles rely on heterogeneous dataset sources. This result may be explained by the different advantages offered by different sources. For instance, Twitter provides short texts while Facebook provides longer texts. Therefore, considering different sources for model building is expected to produce a more comprehensive classification model. Fig. 7 supports this notion, as it shows that 63% of the reviewed studies used multiple datasets rather than a single dataset.

Another important finding that strongly supports the transparency of the evaluation framework is that 71% of the considered articles used public datasets, 23% used private datasets, and 6% used both private and public datasets, see Fig. 8. This enabled the researchers to conduct a fair comparison of the proposed work with others conducted with the same public dataset. Additionally, 73% of the reviewed articles used less than 100,000 instances to build their classification model, while only 17% used more than 100,000 instances, as shown in Fig. 9. A possible explanation for this is that sarcasm detection tasks do not require a huge dataset to differentiate between sarcastic and non-sarcastic text. This is supported by the finding that good performance was observed for most datasets containing less than 100,000 instances. Moreover, the computation overhead is a serious concern when it comes to building a classification model.

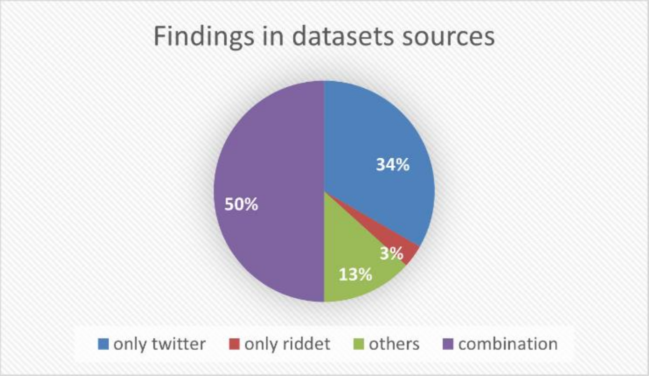


Fig. 6. Dataset sources of the included articles.

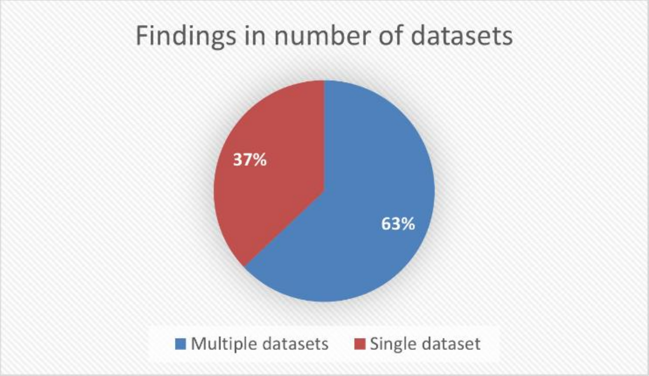


Fig. 7. Use of single or multiple datasets in the reviewed articles.

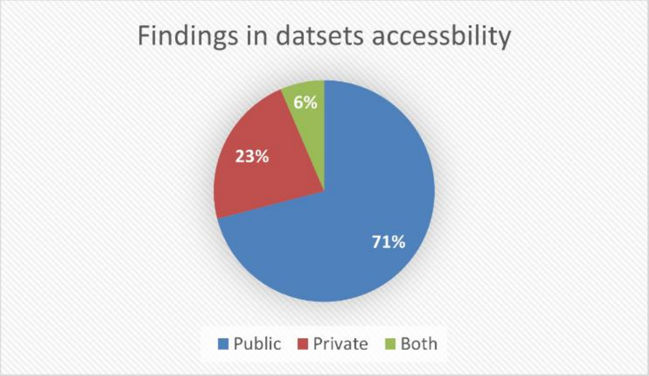


Fig. 8. Accessibility of datasets used in the reviewed articles.

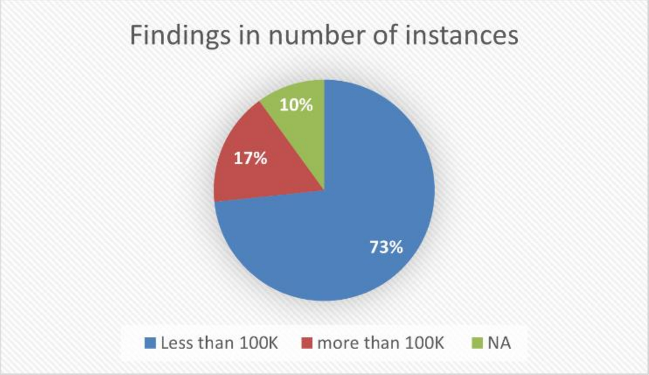


Fig. 9. Number of instances evaluated in the reviewed articles.

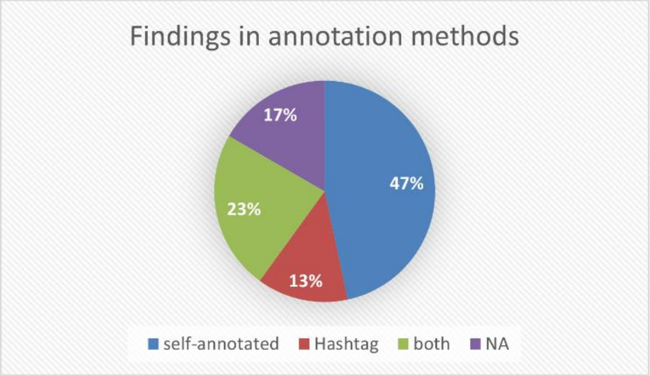


Fig. 10. Annotation methods used in the reviewed articles.

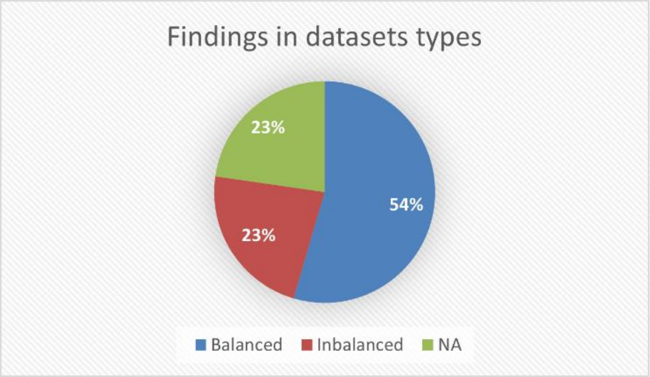


Fig. 11. Distribution of dataset types in the reviewed articles.

Another important issue related to datasets is the annotation method. As expected, 47% of the analyzed articles used self-annotated datasets, illustrated in Fig. 10. Self-annotated datasets are precise because the text is analyzed and annotated by experts and reviewed by another group of experts. However, self-annotation requires a tremendous amount of time [13]. Therefore, tweets could be annotated automatically based on the hashtag included in the tweet; this is a simple and time-conserving approach for annotations that has an acceptable level of correctness. However, only 13% of the considered articles used hashtag-based annotation, and 23% used both the self-annotation and hashtag annotation methods.

Another relevant finding was that 54% of the used datasets were balanced datasets in which the number of sarcastic and non-sarcastic instances was similar, as shown in Fig. 11. This is probably because the nature of the dataset highly influences the model prediction metrics, particularly accuracy and F-measure. These findings reflect those of Eke et al. [13], who also found that an imbalanced dataset can increase the accuracy of the model.

B. *Open Research Questions*

This subsection discusses the common issues and main challenges in the development of sarcasm detection tools for social networks, based on the findings from the reviewed articles.

1) Language used in the social network: The language used in social networks is not only restricted with regard to grammar, but also restricted to words that are not often included in dictionaries. This might pose an additional challenge in the recognition of sarcasm on Twitter and Reddit because of typos, non-vocabulary language, and non-grammatical context. As multilingual text has recently grabbed the attention of researchers, training models in more than one language might be more efficient.

2) Dataset: One of the biggest challenges in training models is the skewness of data. This problem arises when the number of instances in one class, such as sarcastic text, is greater than that in the other class, that is, non-sarcastic text. Furthermore, the quality of the dataset is another challenge. The use of a mixed dataset that uses slang and informal language makes it more difficult to train the classification model, especially if the dataset does not contain hashtags. In such a scenario, creating standard datasets is a solution that may solve the mentioned problems.

3) Text-based sarcasm detection: In speech, sarcasm detection includes features such as eye contact and body language, which help in the recognition of sarcasm. However, text data lack such features. Therefore, it is difficult and takes considerably more effort to identify sarcasm in text.

4) Variable context length: According to the reviewed articles, finding the optimal length of conversational context is a challenge. The Twitter dataset is the most commonly used domain for sarcasm detection, but the short text can be noisy and may not have any relevant features. Therefore, detecting sarcasm from short text is difficult. Overall, the researchers' task is still challenging due to the variability in context length.

5) Emoticons and special characters: In the last decade, the use of emoticons and special characters in social networks has increased. Most people prefer to express their feeling through emojis and emoticons, especially in applications that have restrictions on the number of characters such as Twitter. This increases the likelihood of ambiguity and makes sarcasm detection more difficult. Therefore, researchers should take into account the importance of these features, as they may change the overall sentiment of the sentence.

6) Data annotation: The manual annotation method is a major challenge. The main problem is distinguishing between perceived and intended sarcasm. Most datasets built through manual annotation may, therefore, be limited by differences in the perception of the annotator and the intention of the author of the utterance. As the labeling is based on the perceived sarcasm, this may lead to false positives and false negatives. A solution for this was proposed in [78], according to which the annotator and author of the utterances should be the same individual. Moreover, manual annotation requires a lot of time and the recruitment of domain experts.

7) Lack of real-time sarcasm detection: With the increase in the volume of generated data on social networks, sarcasm detection in real time is a challenging but significant task. Despite this, none of the reviewed articles included real-time data analysis.

Overall, there are still several challenges and open problems in sarcasm detection that need to be worked on. The following subsection provides future research directions.

C. Future Research Directions

This section describes the possible research directions based on our analysis of the 30 articles.

1) Considering more languages: The majority of the recent sarcasm detection works focus on English and ignore other languages. To this end, one possible future direction is to consider multiple language models that have the ability to perform all sarcasm detection sub-tasks for multiple languages.

2) Application of transformers and DL models: While considerably more work will need to be done on transformer-based, DL-based, and hybrid systems, their performance is superior to that of ML and classical NLP techniques. Moreover, the amount of work on transformer-based approaches is still limited, and therefore, there is scope for the development of more transformer-based sarcasm detection models.

3) Tweet correctness techniques: The findings in the datasets indicates that Twitter is the most frequently used source of data for sarcasm detection model evaluation in the reviewed articles. However, tweets are likely to have many typos, which may negatively influence model performance. One possible future direction is to use an automatic technique for typo correction in the early stage of development of sarcasm detection systems.

4) Exploring other social network sources: Twitter and Reddit were the only dataset sources in the reviewed articles. While they are both good sources of data, the addition of more social networks sources would provide a more comprehensive model. Therefore, further work in this domain should focus more on other social networks sources such as Facebook and Instagram.

5) Multi-culture datasets: Sarcasm by its nature differs across cultures. In fact, there could be cultural differences even between people who speak the same language. Therefore, further research could focus on the relationships between culture and sarcasm and the detection of sarcasm in multi-culture datasets.

6) Building multimodal sarcasm detection models: Most of the recent work on sarcasm detection focuses only on text-based datasets. However, considering multimodal models is a good idea for exploring new methods to solve such problems.

7) Use of emojis and emotions: Sarcastic text on social network often contains emojis that are used to express a specific emotion, due to the limitations on the length of posts on some platforms. Therefore, more research is required on new ideas for dealing with data that can improve the performance of such classification models.

VIII. Conclusions

Recently, sarcasm detection, especially in social networks, has grabbed the attention of many researchers. This SLR covers articles on sarcasm detection to answer four research questions. The review of the selected studies provides an analysis of the current approaches, metrics and datasets used to evaluate their models, as well as the challenges facing the development of sarcasm detection applications. In this SLRs, 30 articles published between 2019 and 2022 obtained from four well-known digital databases in Computer Science were analyzed based on their approaches, datasets, and evaluation metrics. Moreover, challenges and open research problems that still prevail in sarcasm detection are discussed. The findings show that the DL approach is most widely utilized, and it is followed by hybrid approaches. Furthermore, Twitter is the most commonly utilized source for datasets, and most researchers used public heterogeneous datasets. With regard to the features of the datasets, most studies used balanced datasets, and there is no consensus among researchers about whether standard, publicly available datasets are suitable for sarcasm detection in social networks. With regard to performance metrics, precision, recall, accuracy, and F1-score were most frequently used in the selected articles, and the majority of the articles used F1-score. Finally, several recommendations, including considering more languages, building multimodal sarcasm detection models and tweet correctness techniques have been suggested to improve the efficiency and performance of sarcasm detection tools.

References

[1] B. Liu and L. Zhang, “A Survey of Opinion Mining and Sentiment Analysis,” in Mining Text Data, C. C. Aggarwal and C. Zhai, Eds. Boston, MA: Springer US, 2012, pp. 415–463. doi: 10.1007/978-1-4614-3223-4_13.

[2] “Biggest social media platforms 2022,” Statista. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ (accessed Oct. 27, 2022).

[3] B. Yee Liau and P. Pei Tan, “Gaining customer knowledge in low cost airlines through text mining,” Ind. Manag. Data Syst., vol. 114, no. 9, pp. 1344–1359, Jan. 2014, doi: 10.1108/IMDS-07-2014-0225.

[4] S. K. Bharti, B. Vachha, R. K. Pradhan, K. S. Babu, and S. K. Jena, “Sarcastic sentiment detection in tweets streamed in real time: a big data approach,” Digit. Commun. Netw., vol. 2, no. 3, pp. 108–121, Aug. 2016, doi: 10.1016/j.dcan.2016.06.002.

[5] S. G. Wicana, T. Y. Ibisoglu, and U. Yavanoglu, “A Review on Sarcasm Detection from Machine-Learning Perspective,” in 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 2017, pp. 469–476. doi: 10.1109/ICSC.2017.74.

[6] U. Yavanoglu, T. Y. Ibisoglu, and S. G. Wıcana, “Sarcasm Detection Algorithms,” Int. J. Semantic Comput., vol. 12, no. 03, pp. 457–478, Sep. 2018, doi: 10.1142/S1793351X18300017.

[7] Y. Kumar and N. Goel, “AI-Based Learning Techniques for Sarcasm Detection of Social Media Tweets: State-of-the-Art Survey,” SN Comput. Sci., vol. 1, no. 6, p. 318, Nov. 2020, doi: 10.1007/s42979-020-00336-3.

[8] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, “Sarcasm detection using machine learning algorithms in Twitter: A systematic review,” Int. J. Mark. Res., vol. 62, no. 5, pp. 578–598, Sep. 2020, doi: 10.1177/1470785320921779.

[9] F. B. Kader, N. H. Nujat, T. B. Sogir, M. Kabir, H. Mahmud, and K. Hasan, “Computational Sarcasm Analysis on Social Media: A Systematic Review.” arXiv, Sep. 20, 2022. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/2209.06170.

[10] M. Bouazizi and T. Ohtsuki, “Sarcasm Over Time and Across Platforms: Does the Way We Express Sarcasm Change?,” IEEE Access, vol. 10, pp. 55958–55987, 2022, doi: 10.1109/ACCESS.2022.3174862.

[11] A.-C. Băroiu and Ștefan Trăușan-Matu, “Automatic Sarcasm Detection: Systematic Literature Review,” Information, vol. 13, no. 8, p. 399, Aug. 2022, doi: 10.3390/info13080399.

[12] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” vol. 2, Jan. 2007.

[13] C. I. Eke, A. A. Norman, Liyana Shuib, and H. F. Nweke, “Sarcasm identification in textual data: systematic review, research challenges and open directions,” Artif. Intell. Rev., vol. 53, no. 6, pp. 4215–4258, Aug. 2020, doi: 10.1007/s10462-019-09791-8.

[14] N. Genc-Nayebi and A. Abran, “A systematic literature review: Opinion mining studies from mobile app store user reviews,” J. Syst. Softw., vol. 125, pp. 207–219, Mar. 2017, doi: 10.1016/j.jss.2016.11.027.

[15] D. G. Maynard and M. A. Greenwood, “Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis,” in Lrec 2014 proceedings, 2014.

[16] S. K. Bharti, K. S. Babu, and S. K. Jena, “Parsing-based sarcasm sentiment recognition in Twitter data,” in 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Aug. 2015, pp. 1373–1380. doi: 10.1145/2808797.2808910.

[17] T. Veale and Y. Hao, “Detecting Ironic Intent in Creative Comparisons,” presented at the ECAI 2010, 2010, pp. 765–770.

[18] M. Birjali, M. Kasri, and A. Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” Knowl.-Based Syst., vol. 226, p. 107134, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.

[19] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” Artif. Intell. Rev., vol. 55, no. 7, pp. 5731–5780, Oct. 2022, doi: 10.1007/s10462-022-10144-1.

[20] V. Hatzivassiloglou and K. R. McKeown, “Predicting the Semantic Orientation of Adjectives,” in 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, Jul. 1997, pp. 174–181. doi: 10.3115/976909.979640.

[21] P. D. Turney and M. L. Littman, “Measuring praise and criticism: Inference of semantic orientation from association,” ACM Trans. Inf. Syst., vol. 21, no. 4, pp. 315–346, Oct. 2003, doi: 10.1145/944012.944013.

[22] H. Han, J. Zhang, J. Yang, Y. Shen, and Y. Zhang, “Generate domain-specific sentiment lexicon for review sentiment analysis,” Multimed. Tools Appl., vol. 77, no. 16, pp. 21265–21280, Aug. 2018, doi: 10.1007/s11042-017-5529-5.

[23] Q. Cao, W. Duan, and Q. Gan, “Exploring determinants of voting for the ‘helpfulness’ of online user reviews: A text mining approach,” Decis. Support Syst., vol. 50, no. 2, pp. 511–521, Jan. 2011, doi: 10.1016/j.dss.2010.11.009.

[24] O. Araque, G. Zhu, and C. A. Iglesias, “A semantic similarity-based perspective of affect lexicons for sentiment analysis,” Knowl.-Based Syst., vol. 165, pp. 346–359, Feb. 2019, doi: 10.1016/j.knosys.2018.12.005.

[25] J. Bhaskar, K. Sruthi, and P. Nedungadi, “Hybrid Approach for Emotion Classification of Audio Conversation Based on Text and Speech Mining,” Procedia Comput. Sci., vol. 46, pp. 635–643, Jan. 2015, doi: 10.1016/j.procs.2015.02.112.

[26] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, “A semantic matching energy function for learning with multi-relational data,” Mach. Learn., vol. 94, no. 2, pp. 233–259, Feb. 2014, doi: 10.1007/s10994-013-5363-6.

[27] W. Zhang, H. Xu, and W. Wan, “Weakness Finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis,” Expert Syst. Appl., vol. 39, no. 11, pp. 10283–10291, Sep. 2012, doi: 10.1016/j.eswa.2012.02.166.

[28] N. N. Yusof, A. Mohamed, and S. Abdul-Rahman, “Reviewing Classification Approaches in Sentiment Analysis,” in Soft Computing in Data Science, Singapore, 2015, pp. 43–53. doi: 10.1007/978-981-287-936-3_5.

[29] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” in Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10), Valletta, Malta, May 2010. Accessed: Oct. 14, 2022. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

[30] S. Park and Y. Kim, “Building thesaurus lexicon using dictionary-based approach for sentiment classification,” in 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), Jun. 2016, pp. 39–44. doi: 10.1109/SERA.2016.7516126.

[31] K. Sundararajan and A. Palanisamy, “Multi-Rule Based Ensemble Feature Selection Model for Sarcasm Type Detection in Twitter,” Comput. Intell. Neurosci., vol. 2020, pp. 1–17, Jan. 2020, doi: 10.1155/2020/2860479.

[32] V. Govindan and V. Balakrishnan, “A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection,” J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 8, pp. 5110–5120, Sep. 2022, doi: 10.1016/j.jksuci.2022.01.008.

[33] J. Godara, I. Batra, R. Aron, and M. Shabaz, “Ensemble Classification Approach for Sarcasm Detection,” Behav. Neurol., vol. 2021, pp. 1–13, Nov. 2021, doi: 10.1155/2021/9731519.

[34] M. S. Razali, A. A. Halin, L. Ye, S. Doraisamy, and N. M. Norowi, “Sarcasm Detection Using Deep Learning With Contextual Features,” IEEE Access, vol. 9, pp. 68609–68618, 2021, doi: 10.1109/ACCESS.2021.3076789.

[35] T. Ptáček, I. Habernal, and J. Hong, “Sarcasm Detection on Czech and English Twitter,” in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, Aug. 2014, pp. 213–223. Accessed: Oct. 29, 2022. [Online]. Available: https://aclanthology.org/C14-1022.

[36] A. Khatri, P. P, and D. A. K. M, “Sarcasm Detection in Tweets with BERT and GloVe Embeddings.” arXiv, Jun. 20, 2020. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/2006.11512.

[37] M. Choli and D. J. Kuss, “Perceptions of blame on social media during the coronavirus pandemic,” Comput. Hum. Behav., vol. 124, p. 106895, Nov. 2021, doi: 10.1016/j.chb.2021.106895.

[38] D. Vinoth and P. Prabhavathy, “An intelligent machine learning-based sarcasm detection and classification model on social networks,” J. Supercomput., vol. 78, no. 8, pp. 10575–10594, May 2022, doi: 10.1007/s11227-022-04312-x.

[39] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[40] S. Zhang, X. Zhang, J. Chan, and P. Rosso, “Irony detection via sentiment-based transfer learning,” Inf. Process. Manag., vol. 56, no. 5, pp. 1633–1644, Sep. 2019, doi: 10.1016/j.ipm.2019.04.006.

[41] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, “Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network,” IEEE Access, vol. 7, pp. 23319–23328, 2019, doi: 10.1109/ACCESS.2019.2899260.

[42] A. Kumar, V. T. Narapareddy, V. Aditya Srikanth, A. Malapati, and L. B. M. Neti, “Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM,” IEEE Access, vol. 8, pp. 6388–6397, 2020, doi: 10.1109/ACCESS.2019.2963630.

[43] Y. Diao et al., “A Multi-Dimension Question Answering Network for Sarcasm Detection,” IEEE Access, vol. 8, pp. 135152–135161, 2020, doi: 10.1109/ACCESS.2020.2967095.

[44] L. Ren, B. Xu, H. Lin, X. Liu, and L. Yang, “Sarcasm Detection with Sentiment Semantics Enhanced Multi-level Memory Network,” Neurocomputing, vol. 401, pp. 320–326, Aug. 2020, doi: 10.1016/j.neucom.2020.03.081.

[45] A. Kamal and M. Abulaish, “CAT-BiGRU: Convolution and Attention with Bi-Directional Gated Recurrent Unit for Self-Deprecating Sarcasm Detection,” Cogn. Comput., vol. 14, no. 1, pp. 91–109, Jan. 2022, doi: 10.1007/s12559-021-09821-0.

[46] A. Onan and M. A. Toçoğlu, “A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification,” IEEE Access, vol. 9, pp. 7701–7722, 2021, doi: 10.1109/ACCESS.2021.3049734.

[47] Y. Du, T. Li, M. S. Pathan, H. K. Teklehaimanot, and Z. Yang, “An Effective Sarcasm Detection Approach Based on Sentimental Context and Individual Expression Habits,” Cogn. Comput., vol. 14, no. 1, pp. 78–90, Jan. 2022, doi: 10.1007/s12559-021-09832-x.

[48] R. Akula and I. Garibay, “Interpretable Multi-Head Self-Attention Architecture for Sarcasm Detection in Social Media,” Entropy, vol. 23, no. 4, Art. no. 4, Apr. 2021, doi: 10.3390/e23040394.

[49] Chunyan Yin, Y. Chen, and W. Zuo, “Multi-Task Deep Neural Networks for Joint Sarcasm Detection and Sentiment Analysis,” Pattern Recognit. Image Anal., vol. 31, no. 1, pp. 103–108, Jan. 2021, doi: 10.1134/S105466182101017X.

[50] N. S et al., “Deep Learning Approaches for Cyberbullying Detection and Classification on Social Media,” Comput. Intell. Neurosci., vol. 2022, p. e2163458, Jun. 2022, doi: 10.1155/2022/2163458.

[51] R. Pandey, A. Kumar, J. P. Singh, and S. Tripathi, “Hybrid attention-based Long Short-Term Memory network for sarcasm identification,” Appl. Soft Comput., vol. 106, p. 107348, Jul. 2021, doi: 10.1016/j.asoc.2021.107348.

[52] Z. L. Chia, M. Ptaszynski, F. Masui, G. Leliwa, and M. Wroczynski, “Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection,” Inf. Process. Manag., vol. 58, no. 4, p. 102600, Jul. 2021, doi: 10.1016/j.ipm.2021.102600.

[53] S. Bhardwaj and M. R. Prusty, “BERT Pre-processed Deep Learning Model for Sarcasm Detection,” Natl. Acad. Sci. Lett., vol. 45, no. 2, pp. 203–208, Apr. 2022, doi: 10.1007/s40009-022-01108-8.

[54] “Automated sarcasm detection and classification using hyperparameter tuned deep learning model for social networks - Vinoth - Expert Systems - Wiley Online Library.” https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.13107 (accessed Oct. 14, 2022).

[55] P. Parameswaran, A. Trotman, V. Liesaputra, and D. Eyers, “Detecting the target of sarcasm is hard: Really??,” Inf. Process. Manag., vol. 58, no. 4, p. 102599, Jul. 2021, doi: 10.1016/j.ipm.2021.102599.

[56] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in Advances in Neural Information Processing Systems, 2014, vol. 27. Accessed: Oct. 14, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

[57] A. Vaswani et al., “Attention Is All You Need.” arXiv, Dec. 05, 2017. doi: 10.48550/arXiv.1706.03762.

[58] D. Jurafsky and J. Martin, Speech and Language Processing. 2022.

[59] H. Lee, Y. Yu, and G. Kim, “Augmenting Data for Sarcasm Detection with Unlabeled Conversation Context.” arXiv, Jun. 11, 2020. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/2006.06259.

[60] M. Shrivastava and S. Kumar, “A pragmatic and intelligent model for sarcasm detection in social media text,” Technol. Soc., vol. 64, p. 101489, Feb. 2021, doi: 10.1016/j.techsoc.2020.101489.

[61] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv, May 24, 2019. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/1810.04805.

[62] T. B. Brown et al., “Language Models are Few-Shot Learners.” arXiv, Jul. 22, 2020. Accessed: Jan. 28, 2023. [Online]. Available: http://arxiv.org/abs/2005.14165.

[63] C. I. Eke, A. A. Norman, and L. Shuib, “Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model,” IEEE Access, vol. 9, pp. 48501–48518, 2021, doi: 10.1109/ACCESS.2021.3068323.

[64] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, “Sarcasm as Contrast between a Positive Sentiment and Negative Situation,” in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, Oct. 2013, pp. 704–714. Accessed: Oct. 14, 2022. [Online]. Available: https://aclanthology.org/D13-1066.

[65] A. Ghosh and Dr. T. Veale, “Fracking Sarcasm using Neural Network,” in Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, San Diego, California, 2016, pp. 161–169. doi: 10.18653/v1/W16-0425.

[66] E. Savini and C. Caragea, “Intermediate-Task Transfer Learning with BERT for Sarcasm Detection,” Mathematics, vol. 10, no. 5, p. 844, Mar. 2022, doi: 10.3390/math10050844.

[67] J. Li, H. Pan, Z. Lin, P. Fu, and W. Wang, “Sarcasm Detection with Commonsense Knowledge,” IEEEACM Trans. Audio Speech Lang. Process., vol. 29, pp. 3192–3201, 2021, doi: 10.1109/TASLP.2021.3120601.

[68] M. Khodak, N. Saunshi, and K. Vodrahalli, “A Large Self-Annotated Corpus for Sarcasm.” arXiv, Mar. 22, 2018. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/1704.05579.

[69] S. Lukin and M. Walker, “Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue.” arXiv, Aug. 28, 2017. Accessed: Oct. 29, 2022. [Online]. Available: http://arxiv.org/abs/1708.08572.

[70] R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, “A Transformer-based approach to Irony and Sarcasm detection,” Neural Comput. Appl., vol. 32, no. 23, pp. 17309–17320, Dec. 2020, doi: 10.1007/s00521-020-05102-3.

[71] R. Ahuja and S. C. Sharma, “Transformer-Based Word Embedding With CNN Model to Detect Sarcasm and Irony,” Arab. J. Sci. Eng., vol. 47, no. 8, pp. 9379–9392, Aug. 2022, doi: 10.1007/s13369-021-06193-3.

[72] C. Van Hee, E. Lefever, and V. Hoste, “Exploring the fine-grained analysis and automatic detection of irony on Twitter,” Lang. Resour. Eval., vol. 52, no. 3, pp. 707–731, Sep. 2018, doi: 10.1007/s10579-018-9414-2.

[73] I. A. Basheer and M. Hajmeer, “Artificial neural networks: fundamentals, computing, design, and application,” J. Microbiol. Methods, vol. 43, no. 1, pp. 3–31, Dec. 2000, doi: 10.1016/S0167-7012(00)00201-3.

[74] D. Rao and D. Ravichandran, “Semi-Supervised Polarity Lexicon Induction,” in Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, Greece, Mar. 2009, pp. 675–682. Accessed: Oct. 14, 2022. [Online]. Available: https://aclanthology.org/E09-1077.

[75] E. Filatova, “Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing,” p. 7.

[76] P. Goel, R. Jain, A. Nayyar, S. Singhal, and M. Srivastava, “Sarcasm detection using deep learning and ensemble learning,” Multimed. Tools Appl., May 2022, doi: 10.1007/s11042-022-12930-z.

[77] A. Rajadesingan, R. Zafarani, and H. Liu, “Sarcasm Detection on Twitter: A Behavioral Modeling Approach,” in Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, Shanghai China, Feb. 2015, pp. 97–106. doi: 10.1145/2684822.2685316.

[78] S. Oprea and W. Magdy, “iSarcasm: A Dataset of Intended Sarcasm.” arXiv, May 01, 2020. Accessed: Oct. 22, 2022. [Online]. Available: http://arxiv.org/abs/1911.03123.

Current Development, Challenges and Future Trends in Cloud Computing: A Survey

Hazzaa N. Alshareef
College of Computing and Informatics, Saudi Electronic University, Riyadh, Kingdom of Saudi Arabia

***Abstract*—Cloud computing is a new paradigm in information and communication technologies (ICTs) that provides the ability to access shared pools of different computing resources that are related to many cloud users within a pay-per-use or on-demand approach. It has transformed the delivery model of ICT from a product to a service. This provides several different advantages for institutions, companies and users based on savings and reduced capital expenditure through lower operating expenses. This paper provides a comprehensive survey of cloud computing. It first develops an understanding of cloud computing in general and discusses its advantages, current development, challenges and future trends. Subsequently, a detailed discussion on the cloud computing architectures, services models, fault tolerance mechanisms, services selection methods, adoption by industry, and scheduling of cloud-based resources is also presented. Nonetheless, cloud computing has many obstacles which expose it to a number of limitations. Some of these challenges include security of data, fault tolerance, and load balancing. A number of techniques in literature are proposed to cope with these challenges which are discussed and analyzed. Experimental data and usage drift validates the popularity of cloud computing and its adoption in recent years. Future trends in cloud computing support the use of intelligent machine learning (ML) techniques and new technologies to cope with some of the challenges and making cloud computing more efficient, secure and commercially viable to be widely accepted.**

Keywords—Cloud computing; security challenges; machine learning; resource scheduling; information and communication technologies

I. INTRODUCTION

In the present digital age, computer systems and associated applications have become inextricable part of life. Concomitantly, the need for better, cheaper, more efficient and on demand application services and infrastructure is felt like never before. Cloud computing is an approach that provides on-demand access to a shared pool of customizable computing resources (e.g. applications, networks, storage, servers etc.) and services [1]. Service providers can disseminate these resources with only marginal interaction and little management effort. Obtaining dynamic computing resources within the cloud computing paradigm provides the ability to cooperate with and scale up/down the given services, taking the demands of clients into account as well as the cost of the leveraged resources. This effectively contributes to a decrease in the operational cost pertaining to IT services. The scalability of cloud services provides smaller businesses with the ability to take advantage of various state of the art expensive and computing-intensive facilities that were previously affordable by large companies only [2].

Cloud computing enables the provision of information services and network computing resources, such as applications, servers, and storage [3], over the internet without installing them or purchasing them on their own. In 2005, Intel, IBM, and various other enterprises (including universities) within the United States began to operate a cloud computing virtual laboratory enterprise. This type of enterprise began with several experiments at North Carolina State University, situated near the IBM headquarters. In 2007, Google and IBM cooperated to start the processing of a new network computing approach, called cloud computing [4]. The new conversional Intel and Microsoft computing method was tested and thereafter caught the attention of a considerable number of research organizations.

In terms of virtualization, computers, networks, storage, and databases can all be potential cloud computing resources according to certain rules and service agreements. Global giants in the IT industry such as Google, IBM, Amazon, Microsoft, Alibaba etc. are investing in advanced research and innovative ways of utilizing cloud computing most effectively and widely. After launching a cloud computing platform, a significant issue is demonstrating the operative distribution and management pertaining to the virtual sharing of resources based on user demand by improving the effectiveness of the resource usage.

Technology advances and market forces are two related and integrated factors that drive interest in cloud computing. Rapidly improving business cases are producing enhancements in computing infrastructure, which has motivated several enterprise applications and services to consider moving to the cloud. In terms of technology, the existence of lower-cost processors and lower-latency networks, integrated with significant progress in virtualization, has moved computation from local IT platforms to disseminated cloud infrastructures. Evidence shows that, although cloud computing is considered a main business path for the upcoming years, moving to the cloud paradigm is seen to encounter a range of challenges. Some of the prominent ones among these are the issues related to security of the cloud data, scheduling of resources, fault tolerance, load sharing and load balancing. For instance, financial institutions are motivated to shift on cloud computing due to many advantages. Howbeit, the intrinsic security issues and challenges related to resource acquisition for smooth services are still hindering the complete migration. In 2014, more than 50 million users' Dropbox accounts were hacked, which resulted in a wide trust-deficit in the security of cloud computing. When cloud computing is considered a viable alternative, it should offer a similar security level to that of the conventional systems. In order to fulfil this goal, a comprehensive awareness regarding attacks and their countermeasures is required in order to detect malicious activities [5].

In summation, the cloud computing landscape has been considerably improved. Not only have further service offerings and providers packed the space, but improvements have also been made in the infrastructure and services. Cloud computing is considered to form an epitome that provides the ability to access shared pools of different computing resources in a pay-per-use or on-demand manner. With its incorporation into conventional systems, the weaknesses of traditional servers are overwhelmed in terms of efficiency, speed and scalability. It also offers savings in capital expenditure through reduced operating expenses. However, a few obstacles still exist that impose limitations when the technology is used. Lack of security, data consolidation [6], load balancing [7] and fault tolerance [8] represent some of the most important restraints of cloud technology. In addition to these, the adoption and absorption of cloud computing into existing systems, especially in small and medium enterprises (SMEs) [9], poses great challenges in terms of system conversion, change acceptance and embracing a myriad range of accompanying technologies which were previously unavailable. This paper provides a review of the cloud computing paradigm and its main computing and implementation options. It also discusses benefits and challenges of moving to the cloud from users' and companies' perspectives. Many of the aforementioned challenges are discussed in detail and a number of suitable ways are presented, based on literature, to cope up with these challenges. It summarizes the proposed solutions and techniques to deal with cloud computing risks and limitations, and discusses how future technologies, such as artificial intelligence [10], [11] and block chain [12], can embark a new era of prevalence of this technology. Gill et al. [10] has used the concept of "Triumvirate: IoT + AI + Blockchain" to describe the influence and interdependence of AI, IoT and Blockchain technologies that are anticipated to shape the future. They have also highlighted that future companies need to be well informed using Big Data Analytics and Data Science techniques to understand market trends, customer preferences and correlations.

The remainder of the paper is organized as follows: Section II provides a background and overview of cloud computing, including a definition, its structural layers, as well as service and delivery models. Section III then discusses benefits of moving to cloud computing. Challenges pertaining to cloud computing and the ways to cope with them are discussed in Section IV. Section V highlights a number of research trends and directions in cloud computing, whereas Section VI provides a discussion on cloud research trends. Section VII gives a conclusion of this work.

II. Cloud Computing Overview

The term 'cloud', or 'fog' in cloud computing signifies the existence of a remote virtual space. This section presents the definition, architecture, service models, delivery models, and the main characteristics of cloud computing.

A. Definition

There is, as yet, no standard definition of cloud computing. This is due to the dynamic nature of the term and its vast area of application. Nonetheless, industry and academic players are making essential strides towards agreeing on a standard definition ([13], [14], [15], and [16]). For example, as declared by the United States National Institute for Standards and Technology (NIST) [2], "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" .

Madhavaiah and Bashir [17] analysed a number of definitions from both a business and research perspective and proposed a comprehensive definition: "Cloud computing is an information technology-based business model, provided as a service over the Internet, where both hardware and software computing services are delivered on-demand to customers in a self-service fashion, independent of device and location within high levels of quality, in a dynamically scalable, rapidly provisioned, shared and virtualized way and with minimal service provider interaction". Nonetheless, cloud computing is still an emerging technology with its ability to integrate and be integrated in new and associated technologies, which would significantly impact and form its true definition over a period of time.

B. Structural Layers

Majority of the researchers agree that the structure of cloud computing is based on four layers [1], [13]. The first layer is the Application Layer, which is located at the top of the architecture and is the layer that is the most visible for end-users. It consists of different applications and software packages related to the real cloud. As an example, office tools, storage management applications, email systems, and virus scanning and removal applications. The second layer is the Platform Layer, which provides the programming-level interface in accordance with different application approaches and operating systems (OS). This provision attempts to simplify the deployment of an application in the environment of the cloud. The third layer is the Infrastructure Layer, which is based on dynamically employing virtualization techniques for assigning the storage and computing resources needed. A well-known example of virtualization is VMware. The fourth layer is the Hardware Layer, which creates the data centers that contain various physical components, such as, cooling infrastructure, electrical power components, switches, routers, and servers. Fig. 1 shows a typical cloud computing ecosystem, whereby a number of hardware components and applications make the cloud, and network nodes comprising of different devices are the end users of cloud infrastructure and services.

C. Service Models

Cloud computing architecture represents a service-oriented approach, in which services are provided by every layer through to the top one. Accordingly, the services that are obtained in the cloud computing paradigm are classified based on three different types: the Software as a Service (SaaS) model, the Platform as a Service (PaaS) model, and the Infrastructure as a Service (IaaS) model.

The SaaS model is a cloud computing approach in which various applications remain on the service provider's cloud infrastructure and are provided to many users via web apps and interfaces. The major idea behind SaaS is based on eradicating the practice of applications residing locally on individual users' devices, where the computing power is insufficient to provide high computing performance and effectiveness to users [18]. The genesis of cloud computing can be traced to the SaaS approach [2]. Some examples of SaaS providers are Intercom [19], Trello[20], Hipchat, and Rackspace [21].

The PaaS model is a service approach for providing a platform that creates and operates different applications based on the programming interface that is obtained from and supported by the cloud provider [22]. Consequently, scalability issues, high server rapidity and storage capacities are all addressed under the PaaS approach. Therefore, PaaS users are able to create, operate and deliver their particular applications based on the use of remote IT platforms. Nonetheless, users are unable to monitor core cloud platforms (e.g., storage, OS or servers). An example of a PaaS provider is Microsoft Windows Azure [23].

The IaaS model is an approach whereby virtual infrastructures, such as virtual servers, storage and other fundamental computing resources, are offered by cloud service providers to users in order to enable them to disseminate and operate their particular applications or OS; and to download or upload files or software on the cloud. Using the IaaS approach, users can monitor their software, including applications, which is disseminated throughout the cloud. Nevertheless, such users have a limited ability to monitor the virtual infrastructure that is obtained from the cloud service provider. An example of an IaaS provider is Amazon EC2 [24].

D. *Delivery Models*

Cloud based services are disseminated to users via four main delivery models, based on, (i) the control required, (ii) the number of users, and (iii) privacy and security demands of the users [2], [25], [26]. These delivery models are categorized into: (i) Public cloud, (ii) Private cloud, (iii) Community cloud, and (iv) Hybrid cloud.

The Public cloud consists of a third party that possesses the physical resources in their entirety and delivers different cloud services to users via the Internet. Users who are supported by the cloud provider range from individuals to corporate institutions.

The Private cloud is provided exclusively to a particular institution. This model can assist in monitoring the performance of a system, security guidelines, and data. An institution is also able to disseminate its own particular cloud services, and a third-party institution can handle the model by itself.

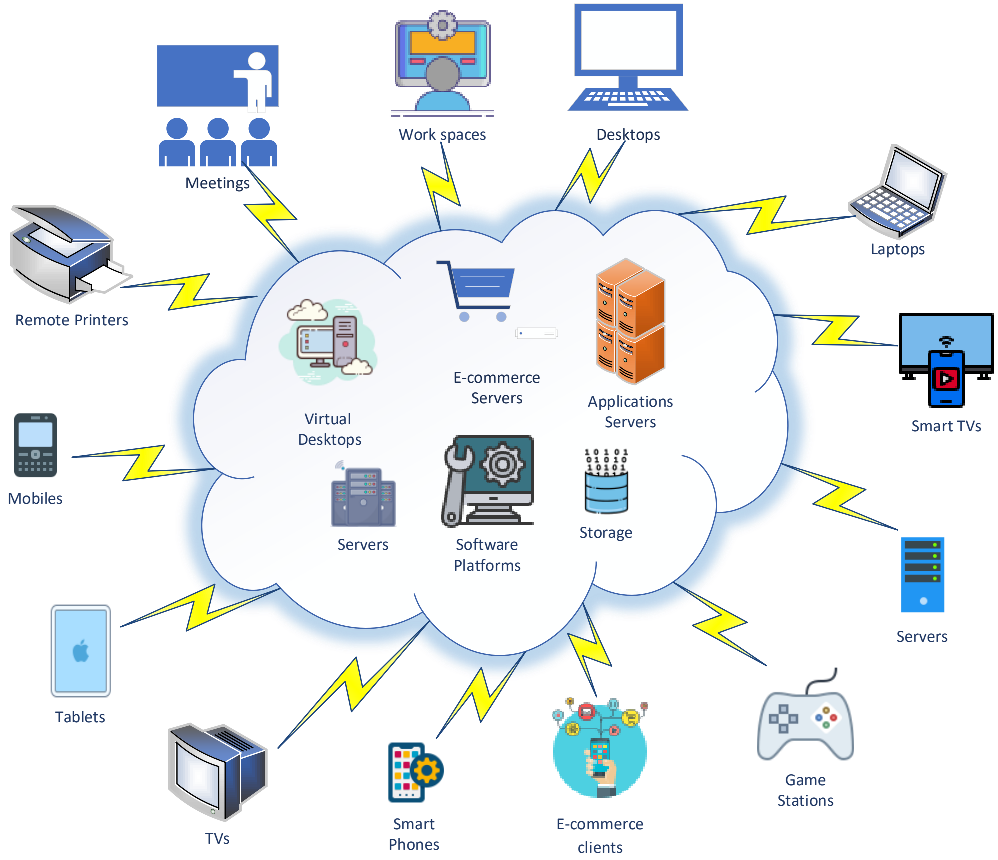


Fig. 1. Cloud computing technology ecosystem.

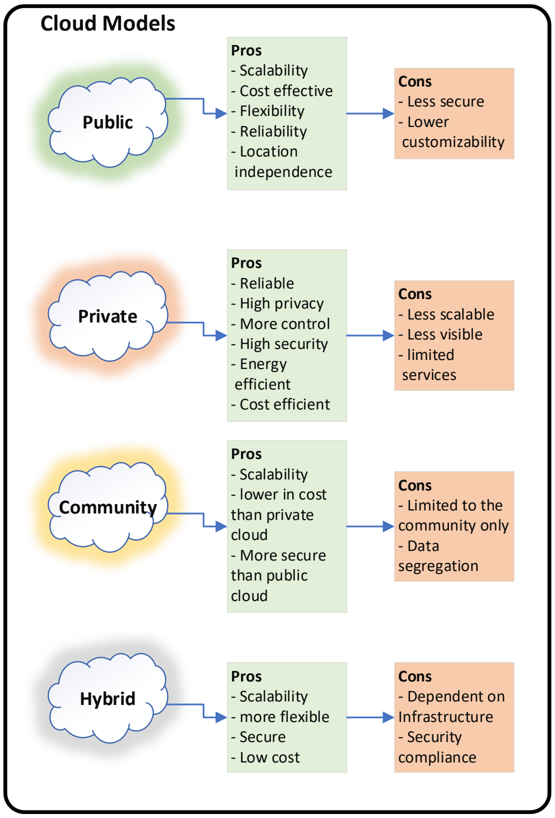


Fig. 2. Cloud delivery models.

The Community cloud represents the provision of different cloud services based on a particular group of institutions with the same mission, compliance conditions, policies, and security demands [22]. A community cloud demonstrates a generalization of the private cloud and, therefore, includes further institutions in each realization.

The merger and combinations of various cloud models (e.g., community, public and private cloud models) represents a hybrid cloud model. They have become very popular in recent years primarily due to popularity and wide usage of cloud services, which now faces complex dynamics of the corporate infrastructures and new business markets. Despite these various models being grouped with each other, they remain distinct and are included in exclusive standards and have distinctive standards and technology with respect to data operability and various applications. The Hybrid cloud inherits the advantages and disadvantages of community, public and private clouds. Consequently, it represents an optimal approach that makes a delicate balance between price and control, which are strong considerations for economic viability as well as user satisfaction with cloud services and applications. Fig. 2 demonstrates the advantages and disadvantages of the cloud delivery models.

E. Cloud Computing Characteristics

Cloud computing is a relatively recent term and the technology has emerged from the usage and trends of computer networks and its associated services and business models. In studies [14], [27]–[29], a number of cloud computing characteristics are discussed. More characteristics may emerge as the technology grows. We present some of the essential characteristics of cloud computing in this section, which define and advocate the core technology and its acceptance specifications.

1) Dynamic (Flexible): cloud resource platforms are dynamically scalable, meaning that, they are able to be enlarged or reduced in size based on user demand, which minimizes the investment risk related to the user and can satisfy the demands of many users. Cloud computing provides users with the sense that infinite computing resources are available to them.

2) Virtualization: cloud computing applications and platforms are created according to resource virtualization concept. Virtualization performs a significant task in developing the effectiveness of resource efficiency and raising the level of service security and reliability.

3) Economies of scale: cloud computing is dominated by large companies, such as IBM, Microsoft, Google, and Amazon, which have the ability to employ large-scale resources that enable them to minimize rental and exploitation costs. As a result, cloud computing companies can recruit as many users as possible. Since there is a large number of potential users involved in any cloud-based service, it becomes financially viable for the service provider to offer the service at a very low cost.

4) On-Demand service: cloud services and platforms are obtained and billed based on users' actual demands. Cloud computing eliminates the risk of a one-time large investment and permits users to acquire only the resources they need. Accordingly, services depend on short-term costs (e.g., on an hourly basis), whereby users release resources once they are no longer required.

5) Dynamic customization: cloud rental resources should be customizable to a considerable extent. For example, in the IaaS delivery model, users are permitted to disseminate virtual and specialized devices. Further services deliver low flexibility and are not applied to general purpose computing. However, it is expected that such services will continue to provide a particular level of customization.

6) High reliability: cloud computing platforms are required to ensure that customer data are secure, so that the application platform is seen to be reliable. In general, platform backups and multiple data are both applied in order to raise platform reliability. Dynamic network management approaches are also applied by cloud computing platforms in order to verify the effectiveness and status of every resource node. The reason for this is that nodes could be dynamically migrated when failure or low effectiveness is encountered. Another reason is to ensure that the performance of the entire system is unaffected in case of a fault.

III. BENEFITS OF MOVING TO THE CLOUD

Owing to the popularity of cloud computing in recent years, the technology incorporates a number of benefits which ascertain the next level of networks and applications sharing, and distribution of services and resources on an economically viable and efficient manner. A number of previous studies [9], [30]–[33] have highlighted some of these advantages. Cloud computing enables on-demand network access and a host of associated applications and infrastructure to a number of customizable computing resources such as servers, software applications, storage spaces, services and other networks. In line with the characteristics identified in the previous section, this section highlights the main benefits and challenges of shifting to the cloud, which mainly include:

A. *Optimum Resource Utilization*

Since most of the cloud computing is based on use-per-pay model, therefore resources are released after every use. This results in the overall utilization of all the computing resources in an efficient and optimized manner, leading to green computing. It is in line with the United Nations' (UN) Sustainable Development Goal (SDG) number 12 [34] that relates to responsible consumption and production of goods and services.

B. *Rent on Demand*

Another major benefit of cloud computing is the availability of all types of computing and infrastructure resources and services for everyone, anywhere, and at any time. Industry 4.0 [35], [36] provides opportunities for growth and sustainability for all kinds of businesses, whether large or small. Future businesses lie in innovation which is the key for success for agile companies of today. Being able to access and utilize state of the art infrastructure and computing resources, as and when needed, startups can easily compete with giants of the industry resulting in better and cheaper products and services for the masses.

C. *Minimized IT Staff*

Moving to cloud computing results in reduction of inhouse IT staff to maintain the existing systems. Some technical staff is, however, still required to work with pre-existing vendors, including specialized vendors, in order to manage particular outsourced applications.

D. *Minimized Infrastructure*

Relocating resources to the cloud, or accessing platform as a service, means that it is possible to maintain a smaller inhouse hardware infrastructure.

E. *Managed Costs*

Since most of the cloud computing service providers gain income on the basis of economies-of-scale, and try their best to cut down the costs, increase the customers and have the latest and updated hardware and software applications. Therefore, prices and licensing can be minimized when adopting cloud computing. The latest costs are based on predetermined services and is derived from the costs model used by the vendor.

F. *Enhanced Vulnerability Control*

Vulnerability control is the ability to track system activities and logs to provide greater control and minimize the risk of attacks by detecting and preventing its occurrence before happening. In the cloud computing ecosystem, the service provider provides services to numerous customers concomitantly and a large revenue is generated as a result. Therefore, the service providers make sure that all the systems are up to date with state-of-the-art technologies in place, and no vulnerabilities or security threats are there.

IV. Cloud Computing Challenges

A number of benefits related to cloud computing are discussed in the previous section. However, it is neither an optimal solution nor risk free, particularly when data is out of users' reach. Other disadvantages of moving to the cloud are reliability issues and system performance, since users are fully dependent on cloud resources. For instance, when accessing the cloud in order to seek a service, the time needed to execute the task (the round-trip time, or RTT) could be an issue for users. This can be exacerbated if the cloud is busy serving other instances or traffic is already congested. Privacy and security are other limitations that are widely known to render cloud computing which is a challenge for users. Some of the major challenges faced by cloud computing are provided as follows:

A. *Reliability*

A cloud computing platform must guarantee the reliability of the application platform and the integrity of customer data. When a large-scale system is experienced, an effective solution is expected in order to receive a high level of reliability. In addition, a dynamic network management system controls the effectiveness and status of the resource nodes, whereby ineffective or failed nodes are dynamically migrated. Consequently, the entire performance of the system is not influenced by these nodes. Ensuring that all systems are working perfectly and reliably poses continuous challenges for the cloud service providers.

B. *Resource Provisioning and Scheduling*

The dynamic deprecation and expansion of resources relies on users' demands, thus presenting new challenges for management systems and cloud platforms. In terms of provisioning of cloud resources, an effective cloud resource provisioning algorithm is highly needed that makes better resource utilization and allocation, reduces response time, and has robustness as well as fault tolerance capabilities. Similarly, scheduling for on-demand resource requirements, or long-term resource reservation is a challenging task, especially when the number of resources and the number of users are extremely large.

C. *Management Issues*

The process of managing a cloud computing platform is extremely complicated. Especially, resource consolidation is one of the key areas of research and have gained substantial attention from the research community in recent years. It involves managing the means of controlling the system's resources effectively, deploying and scheduling different resources in a dynamic manner, and managing clients, their billing systems and service agreements. Nonetheless, applying the approach of having one service provider creates obstacles, such as the following: (i) a lot of much energy is exploited via an enormous data centre in order to have it operational; (ii) centralized cloud data centres are affected by many single point failures; and (iii) data centres are geographically remote from their users, and data need to be moved from their source in order to be processed. This implies that personal or sensitive data generated through the use of different applications are kept in a location other than where they were produced.

D. Fault Tolerance

Fault tolerance refers to the continuity of cloud services even in the existence of any hardware or software malfunction. In case of failure of such components, it is a major challenge to keep all the system running and without performance degradation in presence of a fault.

E. Privacy and Transparency

Privacy and transparency of users' data and cloud services respectively is very crucial in any cloud computing system. In order to gain trust in a cloud-based service, where the users' data and all related credentials are stored in a virtual environment, privacy of data is very crucial. Similarly, transparency of cloud services and virtualization of all systems and infrastructure is of prime importance. Cloud service providers must inform the customers about how their data will be held, stored and transmitted. Things like what security and privacy schemes are deployed, and what internal policies and technologies are in place, are very important from the point of view of a client, especially, when the client is a big organization.

F. Security

Security of cloud computing is far most the biggest challenge in this technology. It deals with all kinds of challenges related to data security, information security, data integrity and confidentiality. Security is one of the main challenges and hurdles when cloud computing implementation and adoption is concerned. Therefore, a detailed discussion on it and the related issues is inevitable for the completeness of this work.

The following section provides a detailed discussion on works related to security challenges, including possible threats, attacks and their countermeasures.

V. Security Challenges and Cloud Computing

One of the potential hindrances in cloud computing adoption is that users are not informed of the physical location of their sensitive data. Since service providers locate cloud data centres in many geographical locations, it leads to a range of security issues and risks. The conventional security approaches, such as intrusion detection systems (IDSs), host-based antivirus software and firewalls, are not able to provide appropriate security through virtualized systems. Fast dissemination of risks derived from the virtualized environments produces different risks [37]. Subramanian and Jeyaraj [37] identify the top 12 threats to the cloud according to the Cloud Security Alliance (CSA), which include compromised credentials and broken authentication, and denial-of-service (DoS) attacks, as presented in Table I. Most of the threats identified are related to data breaching, representing the principal security problem that needs to be addressed.

Kamara and Lauter [38] indicate that many different risks emerge depending on the use of public clouds. Data integrity and confidentiality represent the highest risks and produce different but related issues. The authors propose a crypto-cloud architecture that contains three main features: the cloud storage service provider (CSSP), the consumer of the data, and the data authority (the user that possesses the related data). Encrypted files are uploaded by the data authority and the CSSP permits access to the files. The demanded file is then downloaded and decrypted based on the use of suitable credentials and tokens. This type of architecture faces various security issues at the service-level agreement (SLA), computation and communication levels. For instance, issues occur within the communication level because the same infrastructures and resources are shared through a virtual machine (VM), which increases the possibility of attacks.

TABLE I. Cloud Security Alliance's Top 12 threats

Threat Number	Threat Name
1	Compromised credentials and broken authentication
2	Malicious insiders
3	Denial-of-Service (DoS) attacks
4	Account hijacking
5	Inadequate diligence
6	Permanent data loss
7	The Advanced Persistent Threat (APT) parasite
8	Cloud service abuses
9	Data breaches
10	Exploited system vulnerabilities
11	Hacked interface and App. Program Interfaces
12	Shared technology, shared dangers

Bhadauria and Sanyal [39] categorize these problems into (i) network, (ii) application, and (iii) host levels. Attacks are determined based on the communication levels indicated. The main security risks at the network level are data integrity and confidentiality, where problems related to network security include reused IP addresses, sniffer attacks, Domain Name System (DNS) attacks, and prefix hijacking in the Border Gateway Protocol (BGP). At the application level, security is required in order to prevent attackers gaining control and compromising the application. Problems at this level include hacking, dictionary attacks, hidden field manipulation, CAPTCHA breaking, cookie poisoning and distributed DoS (DDoS) problems. Finally, the main host-level risks are foot printing, Trojan horses, unauthorized access, DoS, password cracking, profiling, worms, and viruses. Applying the aspect of cloud virtualization represents the largest of the computational-level issues.

Data is considered the key source of item related to any crypto-cloud approach. CSA regards a data breach as the highest security risk. Multi-tenancy and maintaining data storage on a remote place (i.e., out of your control) can cause data leakage. Chen and Zhao [40] produced a data life cycle: Generation => Transfer => Use => Share => Storage => Archival => Destruction, which requires protection during all the phases. Data-level security is categorized as data in rest and data in transit. Data in transit does not cause extra security threats in comparison with data in rest, as transmitting data can be performed based on a secured data transfer method. From the hacker's perspective, data in rest poses a high level of attraction.

As regards the cloud services are concerned, these are given by providers to consumers based on appropriate SLAs. Thus, major items related to the crypto cloud include the accountability of ensuring that SLAs are maintained. In practice, there is no exclusive standard for SLAs that are applied for all the requirements of security management. However, a few standards, such as the European Network and Information Security Agency (ENISA) and the European Commission Secure Provisioning of Cloud Services (SPECS), offer security by maintaining SLAs. Applying an SLA assists in obtaining an adequate level of quality of service.

Previous research has indicated that the above issues are encountered via three major attack vectors [5]: the network, the computing hardware, and the hypervisor (the computer software, firmware or hardware that creates and runs VMs). There also exist three kinds of attacker map of the three vectors: the cloud provider, internal users, and the external users. It is also possible for a cloud provider to act as an attacker. Permission and authority is given to employees working on the could, for example, might be exploited in order to steal sensitive user information based on either logical or physical manipulation of the hardware platform. External users can also have an impact on data integrity and confidentiality by interfering with communication channels or through lying dormant within the system in order to attack it later. Internal users, such as the owners of a VM instance, could use the hypervisor to attack other VM instances.

A. Major Cloud Attacks

The following subsection contains a discussion of the different attacks that have been discussed in previous research [41]–[44]. These types of attacks, which can be launched over a cloud infrastructure, are examined and presented along with the countermeasures that can be taken to control them.

1) Network-based attacks: The network represents a major vehicle of attack against applications that are being performed within a cloud platform. The majority of such attacks are closely related to the types of attacks typically recognized in conventional technology, although there are some network-based attacks that specifically relate to cloud computing.

2) Hardware-based attacks: Confidential data is protected from illegal access by being maintained within an encrypted form, interacting through different encrypted channels. However, data has to be decrypted sometimes for performing different computations from time to time. Attackers benefit from a multi-tenant environment in which they can simply access various physical resources (e.g., disk buses, memory buses, and instruction and data caches [L1, L2, L3]). Attackers explore and exploit decrypted data and the secret keys related to different common algorithms (e.g., RSA, DES, and AES) and VM instances.

3) Hypervisor-based attacks: The hypervisor is defined as the software layer which is located between the physical hardware and VMs in order to identify the fundamental architecture. The hypervisor is essential for ensuring the characteristics of cloud multi-tenancy. Moreover, it assigns several physical resources to guest VMs (e.g., peripherals, CPU and main memory). On the security side, hypervisors are the most important layer of protection within the cloud stack, since this is the highest privilege level. If attackers can gain control at the hypervisor level and compromise VM isolation, they can control any resource related to the host system.

B. Countermeasures against Cloud Attacks

Cloud providers apply well-known approaches to protect against network-based attacks (e.g., antivirus gateways, IDSs, and firewalls). Such approaches are currently being extensively disseminated within edge networks in order to protect end systems from various forms of attack and to control and check outgoing and incoming traffic. For hardware-based attacks, newly arising techniques (e.g., Arm TrustZone technology and Intel Software Guard Extensions [SGX]) could prevent side-channel attacks. Another prospect is based on protecting hardware using cryptography. A commonly agreed instruction by cloud providers is the Intel Advanced Encryption Standard New Instructions (AES-NI), which proceeds towards different cache-based software side-channel attacks. In the case of hypervisor-based attacks, several software and hardware isolation mechanisms offer resources secure separation, whereby some hardware mechanisms, such as AES-NI, can have an impact on security within this level and are considered a part of hardware-based hypervisor protection.

Senyo et al. [18] also present a valuable tool for navigation, which could be applied by IT personnel in order to gain further insight into security threats that are based on the use of cloud computing. Personnel can then weigh the advantages and disadvantages of any improved resolutions.

VI. Discussion on Cloud Research Trends and Future Technologies

Cloud computing technology is said to be one of the biggest revenue generators for the software companies in recent years. Fig. 3 shows the spending in public cloud computing IaaS hardware and software worldwide in US billion dollars as depicted by Forbes [45]. It also shows that share of this spending by PaaS and SaaS/ The projected values up to 2026 shows that the spending in this technology will remain on the rise.

One key weakness of cloud technology is indicated by the low level of control over data that is disseminated to the cloud provider. This weakness is a major hazard, causing various problems, such as DoS, malicious insiders, and account or service traffic hijacking, which represents an essential impairment to massive cloud adoption. When users are not able to enter different physical systems, they must rely solely on the infrastructure provider in terms of addressing issues that are incurred with regard to data security. Previously, a capable method was Trusted Computing (TC), by which Trusted Platform Module (TPM) characteristics were applied to offer integrity for the software stack (i.e., the VM layer). The Intel Trusted Execution Technology (TXT) is one example of this method. The Intel SGX also represents a promising technique and is defined as an instruction set architecture (ISA) extension, which provides the ability to execute a number of different instructions within a secure memory location, known as the secure enclave. Accordingly, SGX allows users to apply effective security for their own data and applications without having to trust the cloud operator. A similar characteristic is provided by a further research domain, called homomorphic cryptography [46], which permits different calculations to be performed on encrypted data without acquiring a secret key or any decryption of the data.

Another research trend relies on the use of containers for the purpose of abstracting different applications based on the underlying OS, thus providing the ability for more rapid improvement and simpler deployment. Docker [47] represents a well-known container, whereby spreading the container term depends on the support provided by large providers, such as Amazon, Google, and OpenStack. Containers can also be utilized for improving user application security.

For institutions that require an increased level of data security (e.g., banks, the military, and trading and insurance companies), the requirements with regard to customer information security are very high. Ensuring data security in cloud computing is a common issue of concern for such institutions. Presently, service providers and researchers offer several different resolutions. Within the new application domain, there exist several security concerns that should be resolved. Although a few protection methods have reached a particular level of practical maturity, other related methods remain in their infancy and are inappropriate for dissemination to an operating setup. Several different methods exist for addressing vulnerabilities to data breaches and shared technology risks; however, they require further improvement. It has been seen that solutions to attacks related to the network-level have led to improvements. In contrast, approaches at the application level to addressing DoS risks and account or service traffic hijacking need to be improved, and several researchers are still investing time in these areas. There are also several tasks involving identifying new resolutions that would provide protection against the various hypervisor-based attacks encountered. Furthermore, applications attempt to leverage the infrastructure of the cloud based on utilizing heterogeneous resources derived from several providers.

The broad trend is to aim at using infrastructure from several providers and compared with conventional cloud offerings from single providers, disperse computing apart from resources. Subsequently, new computing approaches that aim at fulfilling market needs are evolving. In practice, many different security problems have been identified in relation to SLAs, computation, and communication. As referred to earlier, there have recently been large and significant security problems, presenting a range of opportunities for hackers to break service cryptosystems. Cloud computing has been demonstrated to be inadequate when it comes to security problems, and cloud service providers need to take into account that security should form an inevitable and important factor, and not be an afterthought.

Recently, a lot of effort is made by the machine learning (ML) community to scale up and enhance the existing data mining and ML algorithms to meet the challenges of handling large amounts of data [48], [49]. The work by Kim et al. [50] demonstrates a network threat detection and classification method based on ML, thus paving a path to the intelligent threat analysis technology. Similarly, the use of artificial intelligence (AI) and ML algorithms for the analysis of cloud services, for security, and for predictive and prescriptive analytics is very promising.

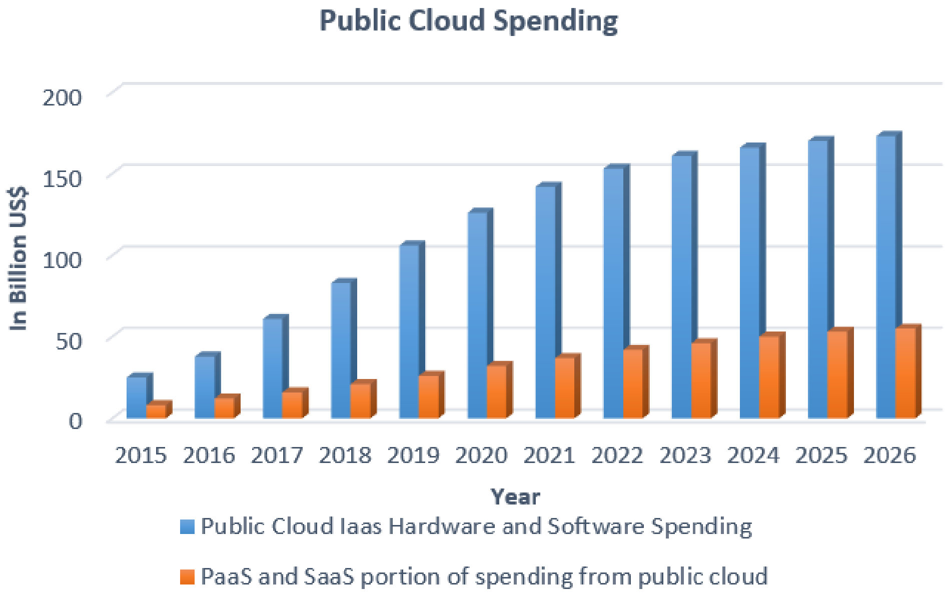


Fig. 3. Public cloud spending. Past, present and future.

VII. Conclusion

Cloud computing is an extremely rapidly developing technology in the domain of computing. There exist several benefits to applying cloud computing, such as anytime-anywhere accessibility, more effective geographical coverage, greater time efficiency, and reduced infrastructure costs. Nonetheless, there are also obstacles to applying cloud computing, such as lack of expertise and resources, cloud services management, privacy, and the need for data security. The majority of the services pertaining to the infrastructure of the hosting cloud, including storage and computing resources, exist in data centres. The hosting of applications in a single provider's cloud is seen to be simple and to offer various benefits. Nonetheless, there are a myriad of associated risks and challenges with cloud computing. Information privacy, security and data integrity are among the top of these. Trends and results from the literature shows that cloud computing is still emerging and new associated technologies are being developed to cope up with the existing challenges. The use of AI in cloud computing can mitigate some of the risks and provide solutions to previously unresolved issues.

Acknowledgment

The author highly appreciates the support provided by Saudi Electronic University for this research.

References

[1] A. Vafamehr and M. E. Khodayar, "Energy-aware cloud computing," Electr. J., vol. 31, no. 2, pp. 40–49, Mar. 2018, doi: 10.1016/j.tej.2018.01.009.

[2] "Final Version of NIST Cloud Computing Definition Published," NIST, Oct. 2011, doi: 10/final-version-nist-cloud-computing-definition-published.

[3] W. D. Tian and Y. D. Zhao, Optimized Cloud Resource Management and Scheduling: Theories and Practices, 1st edition. Waltham, MA: Morgan Kaufmann, 2014.

[4] K. D. Foote, "A Brief History of Cloud Computing," DATAVERSITY, Dec. 17, 2021. https://www.dataversity.net/brief-history-cloud-computing/ (accessed Jan. 04, 2023).

[5] L. Coppolino, S. D'Antonio, G. Mazzeo, and L. Romano, "Cloud security: Emerging threats and current solutions," Comput. Electr. Eng., vol. 59, pp. 126–140, Apr. 2017, doi: 10.1016/j.compeleceng.2016.03.004.

[6] L. Helali and M. N. Omri, "A survey of data center consolidation in cloud computing systems," Comput. Sci. Rev., vol. 39, p. 100366, Feb. 2021, doi: 10.1016/j.cosrev.2021.100366.

[7] V. Gehlot, D. S. P. Singh, and D. A. Saxena, "A Survey On Energy-Aware Load Balancing In Cloud Computing Environment," Int. J. Sci. Technol. Res., vol. 8, no. 12, pp. 4055–4060, Dec. 2019.

[8] P. Kumari and P. Kaur, "A survey of fault tolerance in cloud computing," J. King Saud Univ. - Comput. Inf. Sci., vol. 33, no. 10, pp. 1159–1176, Dec. 2021, doi: 10.1016/j.jksuci.2018.09.021.

[9] D. Widyastuti and I. Irwansyah, "Benefits And Challenges Of Cloud Computing Technology Adoption In Small And Medium Enterprises (SMEs)," Jan. 2018. doi: 10.2991/bcm-17.2018.46.

[10] S. S. Gill et al., "Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges," Internet Things, vol. 8, p. 100118, Dec. 2019, doi: 10.1016/j.iot.2019.100118.

[11] K. N. Qureshi, G. Jeon, and F. Piccialli, "Anomaly detection and trust authority in artificial intelligence and cloud computing," Comput. Netw., vol. 184, p. 107647, Jan. 2021, doi: 10.1016/j.comnet.2020.107647.

[12] M. R. Dorsala, V. N. Sastry, and S. Chapram, "Blockchain-based solutions for cloud computing: A survey," J. Netw. Comput. Appl., vol. 196, p. 103246, Dec. 2021, doi: 10.1016/j.jnca.2021.103246.

[13] M. Taghipour, E. Mowloodi, M. Mahboobi, and J. Abdi, "Application of Cloud Computing in System Management in Order to Control the Process," vol. 3, pp. 34–55, May 2020, doi: 10.31058/j.mana.2020.33003.

[14] "(PDF) Cloud Computing: A review of the Concepts and Deployment Models." https://www.researchgate.net/publication/317413701_Cloud_Computing_A_review_of_the_Concepts_and_Deployment_Models (accessed Jan. 03, 2023).

[15] S. Slimani, T. Hamrouni, and F. Ben Charrada, "Service-oriented replication strategies for improving quality-of-service in cloud computing: a survey," Clust. Comput., vol. 24, no. 1, pp. 361–392, Mar. 2021, doi: 10.1007/s10586-020-03108-z.

[16] P. T. Endo, M. Rodrigues, G. E. Gonçalves, J. Kelner, D. H. Sadok, and C. Curescu, "High availability in clouds: systematic review and research challenges," J. Cloud Comput., vol. 5, no. 1, p. 16, Oct. 2016, doi: 10.1186/s13677-016-0066-8.

[17] "Defining Cloud Computing in Business Perspective: A Review of Research - C. Madhavaiah, Irfan Bashir, Syed Irfan Shafi, 2012." https://journals.sagepub.com/doi/abs/10.1177/0972262912460153 (accessed Jan. 03, 2023).

[18] P. K. Senyo, E. Addae, and R. Boateng, "Cloud computing research: A review of research themes, frameworks, methods and future research directions," Int. J. Inf. Manag., vol. 38, no. 1, pp. 128–139, Feb. 2018, doi: 10.1016/j.ijinfomgt.2017.07.007.

[19] ["Making Internet Business Personal | Intercom." https://www.intercom.com (accessed Jan. 13, 2023).

[20] "Manage Your Team's Projects From Anywhere | Trello." https://trello.com/ (accessed Jan. 13, 2023).

[21] "Rackspace Technology | Multicloud Solutions Provider." https://www.rackspace.com/node/22215 (accessed Jan. 13, 2023).

[22] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing — The business perspective," Decis. Support Syst., vol. 51, no. 1, pp. 176–189, Apr. 2011, doi: 10.1016/j.dss.2010.12.006.

[23] "Cloud Computing Services | Microsoft Azure." https://azure.microsoft.com/en-us (accessed Jan. 13, 2023).

[24] "Cloud Computing Services - Amazon Web Services (AWS)," Amazon Web Services, Inc. https://aws.amazon.com/ (accessed Jan. 13, 2023).

[25] H. Mouratidis, S. Islam, C. Kalloniatis, and S. Gritzalis, "A framework to support selection of cloud providers based on security and privacy requirements," J. Syst. Softw., vol. 86, no. 9, pp. 2276–2293, Sep. 2013, doi: 10.1016/j.jss.2013.03.011.

[26] P.-F. Hsu, S. Ray, and Y.-Y. Li-Hsieh, "Examining cloud computing adoption intention, pricing mechanism, and deployment model," Int. J. Inf. Manag., vol. 34, no. 4, pp. 474–488, Aug. 2014, doi: 10.1016/j.ijinfomgt.2014.04.006.

[27] "The Role of Cloud Computing in the Development of Information Systems for SMEs," IBIMA Publishing. https://ibimapublishing.com/articles/JCC/2017/736545/ (accessed Jan. 03, 2023).

[28] R. O. Aburukba, M. AliKarrar, T. Landolsi, and K. El-Fakih, "Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing," Future Gener. Comput. Syst., vol. 111, pp. 539–551, Oct. 2020, doi: 10.1016/j.future.2019.09.039.

[29] "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing | EURASIP Journal on Wireless Communications and Networking | Full Text." https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-019-1557-3 (accessed Jan. 03, 2023).

[30] S. A. Bello et al., "Cloud computing in construction industry: Use cases, benefits and challenges," Autom. Constr., vol. 122, p. 103441, Feb. 2021, doi: 10.1016/j.autcon.2020.103441.

[31] T. Vasiljeva, S. Shaikhulina, and K. Kreslins, "Cloud Computing: Business Perspectives, Benefits and Challenges for Small and Medium

Enterprises (Case of Latvia),” Procedia Eng., vol. 178, pp. 443–451, Jan. 2017, doi: 10.1016/j.proeng.2017.01.087.

[32] A. Aljumah and T. A. Ahanger, “Cyber security threats, challenges and defence mechanisms in cloud computing,” IET Commun., vol. 14, no. 7, pp. 1185–1191, 2020, doi: 10.1049/iet-com.2019.0040.

[33] M. Humayun, “Role of Emerging IoT Big Data and Cloud Computing for Real Time Application,” Int. J. Adv. Comput. Sci. Appl. IJACSA, vol. 11, no. 4, Art. no. 4, Jun. 2020, doi: 10.14569/IJACSA.2020.0110466.

[34] “THE 17 GOALS | Sustainable Development.” https://sdgs.un.org/goals (accessed Jan. 20, 2023).

[35] M. Ghobakhloo, “Industry 4.0, digitization, and opportunities for sustainability,” J. Clean. Prod., vol. 252, p. 119869, Apr. 2020, doi: 10.1016/j.jclepro.2019.119869.

[36] T. Masood and P. Sonntag, “Industry 4.0: Adoption challenges and benefits for SMEs,” Comput. Ind., vol. 121, p. 103261, Oct. 2020, doi: 10.1016/j.compind.2020.103261.

[37] N. Subramanian and A. Jeyaraj, “Recent security challenges in cloud computing,” Comput. Electr. Eng., vol. 71, pp. 28–42, Oct. 2018, doi: 10.1016/j.compeleceng.2018.06.006.

[38] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” in Financial Cryptography and Data Security, Berlin, Heidelberg, 2010, pp. 136–149. doi: 10.1007/978-3-642-14992-4_13.

[39] R. Bhadauria and S. Sanyal, “Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques,” Int. J. Comput. Appl., vol. 47, no. 18, pp. 47–66, Jun. 2012, doi: 10.5120/7292-0578.

[40] “Data Security and Privacy Protection Issues in Cloud Computing | IEEE Conference Publication | IEEE Xplore.” https://ieeexplore.ieee.org/abstract/document/6187862 (accessed Jan. 03, 2023).

[41] H. A. Kholidy, “Detecting impersonation attacks in cloud computing environments using a centric user profiling approach,” Future Gener. Comput. Syst., vol. 117, pp. 299–320, Apr. 2021, doi: 10.1016/j.future.2020.12.009.

[42] H. Abusaimeh, “Security Attacks in Cloud Computing and Corresponding Defending Mechanisms,” Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 3, pp. 4141–4148, Jun. 2020, doi: 10.30534/ijatcse/2020/243932020.

[43] B. T. Devi, S. Shitharth, and M. A. Jabbar, “An Appraisal over Intrusion Detection Systems in Cloud Computing Security Attacks,” in 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Mar. 2020, pp. 722–727. doi: 10.1109/ICIMIA48430.2020.9074924.

[44] A. Saeed, P. Garraghan, and S. A. Hussain, “Cross-VM Network Channel Attacks and Countermeasures Within Cloud Computing Environments,” IEEE Trans. Dependable Secure Comput., vol. 19, no. 3, pp. 1783–1794, May 2022, doi: 10.1109/TDSC.2020.3037022.

[45] “Current Cloud Computing Statistics Send Strong Signal of What’s Ahead,” Insight. https://www.insight.com/en_US/content-and-resources/2016/11032016-current-cloud-computing-statistics.html (accessed Jan. 27, 2023).

[46] S. Q. Ren et al., “Secure searching on cloud storage enhanced by homomorphic indexing,” Future Gener. Comput. Syst., vol. 65, pp. 102–110, Dec. 2016, doi: 10.1016/j.future.2016.03.013.

[47] “Docker: Accelerated, Containerized Application Development,” May 10, 2022. https://www.docker.com/ (accessed Jan. 24, 2023).

[48] D. Pop, “Machine Learning and Cloud Computing: Survey of Distributed and SaaS Solutions.” arXiv, Mar. 29, 2016. doi: 10.48550/arXiv.1603.08767.

[49] U. A. Butt et al., “A Review of Machine Learning Algorithms for Cloud Computing Security,” Electronics, vol. 9, no. 9, Art. no. 9, Sep. 2020, doi: 10.3390/electronics9091379.

[50] H. Kim, J. Kim, Y. Kim, I. Kim, and K. J. Kim, “Design of network threat detection and classification based on machine learning on cloud computing,” Clust. Comput., vol. 22, no. 1, pp. 2341–2350, Jan. 2019, doi: 10.1007/s10586-018-1841-8.

A High-performance Approach for Irregular License Plate Recognition in Unconstrained Scenarios

Hoanh Nguyen
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

***Abstract*—This paper proposes a novel framework for locating and recognizing irregular license plates in real-world complex scene images. In the proposed framework, an efficient deep convolutional neural network (CNN) structure specially designed for keypoint estimation is first employed to predict the corner points of license plates. Then, based on the predicted corner points, perspective transformation is performed to align the detected license plates. Finally, a lightweight deep CNN structure based on the YOLO detector is designed to predict license plate characters. The character recognition network can predict license plate characters without depending on license plate layouts (i.e., license plates of single-line or double-line text). Experiment results on CCPD and AOLP datasets demonstrate that the proposed method obtains better recognition accuracy compared with previous methods. The proposed model also achieves impressive inference speed and can be deployed in real-time applications.**

Keywords—License plate recognition; deep learning; convolutional neural network; keypoint detector; YOLO detector

I. Introduction

Recognizing license plates is a crucial area of research due to its numerous practical applications, including monitoring road traffic, collecting tolls automatically, enforcing traffic laws, and more. A license plate recognition pipeline for recognizing irregular license plates typically includes four stages: license plate detection, perspective correction, segmenting characters and recognizing characters. License plate detection aims to extract license plate regions from input images. The accuracy of the entire system heavily relies on the accuracy of license plate detection, as the extracted regions are utilized in subsequent stages. As real-world images containing license plates are captured under different viewpoints, license plates may have arbitrary direction. As a result, perspective correction is performed to align the detected license plates. For recognizing characters, a segmentation approach is first used to decompose the aligned license plate image containing a sequence of characters into sub-images of individual character. Then, a character recognition approach is employed to classify each character. Classical approaches based on computer vision for license plate recognition [1], [2] primarily focus on extracting features of license plates based on the background color, contours and edges, and use these hand-crafted features for locating license plates and decomposing characters. Recently, numerous CNN-based approaches for license plate recognition have been proposed, leading to significant advancements. These methods first adopt CNN architectures to extract discriminative feature representations from input images. A network is then used to locate the location of license plates. With the detected license plates, a classifier is proposed to search for license plate characters and classify them. Since character segmentation is a challenge problem due to the effect of lighting conditions, shadows, and noise, various approaches [3], [4] have been proposed to directly recognize license plate characters without segmentation. With the success of CNN and text recognition, CNN-based license plate recognition methods have obtained great achievements in both accuracy and efficient. However, previous methods still depend on high-end GPUs or controlled environments such as specific viewing angles or simple backgrounds. Furthermore, with the growing number of license plate designs, license plate recognition systems that concentrate on single-line plates or frontal plate detection and recognition face increasing challenges.

In view of these issues, a novel framework for detecting and recognizing irregular license plates in real-world complex scene images is designed in this paper. The proposed model can locate and recognize various types of license plates with arbitrary shooting angles in difficult conditions. There are three stages in the proposed model: license plate detection, perspective correction, and recognizing characters. For license plate detection, this paper employs a state-of-the-art object detector and extends it for predicting four corner points of license plates, which are then used to rectify distorted license plates. For license plate recognition, this paper designs a segmentation-free model based on a fast and efficient object detection architecture for predicting license plate characters. The results of the experiments conducted on two extensive datasets show that the proposed model boasts both a high recognition accuracy and rapid inference speed.

This paper is structured as follows: Section II presents a literature review of license plate recognition. Section III offers an overview of the proposed approach. The details of the proposed pipeline are described in Section IV. The discussion of experimental results can be found in Section V. Lastly, the conclusion is outlined in Section VI.

II. Literature Review

This section provides a brief literature review on the topic of license plate recognition. This paper focuses on recent methods that are based on deep learning for end-to-end license plate recognition. For studies involving traditional image processing strategies or focused on license plate detection, please refer to [1], [2], [5], [30].

Since license plates usually occupy small portions of input images, various methods proposed to first detect vehicles and then locate license plate regions to improve license plate detection performance. For this purpose, Sergio and Claudio [6] proposed a novel CNN model that includes a YOLO-based network for vehicle detection and license plate detection and an optical character recognition module for character recognition. The model can detect and rectify multiple distorted license plates before feeding the rectified license plates to the optical character recognition module to obtain results. In [7], the authors presented an end-to-end license plate recognition system utilizing the YOLO detector [8], [9]. This approach first locates the vehicles in the input image by a YOLO detector and then detects their respective license plates in the vehicle patches by another YOLO detector. Afterward, the model detects and recognizes all license plate characters simultaneously by forwarding the license plate region into the CR-NET model [10]. The results showed that this approach obtains high accuracy and fast inference speed. However, the model only recognizes single-line license plate taken from a frontal angle.

Due to the effect of environment conditions, character segmentation is a challenge problem. Moreover, any incorrect character location produced by character segmentation will lead to misrecognition of the license plate characters. To solve this problem, various methods proposed to avoid character segmentation. Wang et al. [11] introduced a cascade approach (i.e., VSNet) for irregular license plate recognition. VSNet consists of a license plate detection network that makes predictions using multiple feature levels produced by a fusion network and a license plate recognition network that features an encoding layer for left-to-right feature extraction and a weight-sharing classifier for character recognition. In addition, a vertex-estimation branch is proposed to rectify distorted license plate images. In [12], the authors presented an end-to-end convolutional neural network for license plate recognition that eliminates the need for character segmentation. The network is implemented on FPGA with very fast processing speed. To enhance the accuracy of license plate recognition in unrestricted conditions, Zou et al. [13] proposed a robust license plate recognition framework that uses a combination of Bi-LSTM and contextual position information of license plate characters to locate the characters in the license plate. The authors utilized deep separable convolutions and a spatial attention mechanism for license plate feature extraction to activate the character feature regions and thoroughly extract the features of license plates.

In summary, although the above methods have achieved some significant accomplishments, they have not fully addressed the issue of irregular license plate recognition in unconstrained scenarios. Furthermore, these methods mostly require hardware with high-end GPUs, which is difficult to implement in practical applications.

III. Overview of the Proposed Framework

The proposed method includes three stages as outlined in Fig. 1. Specifically, the proposed method takes images as inputs and sequentially undergoes license plate detection, perspective correction, and character recognition to produce final license plate characters. Both stage 1 and stage 3 are based on simple and efficient deep CNN architectures for fast inference speed. Overview of each stage is described further.

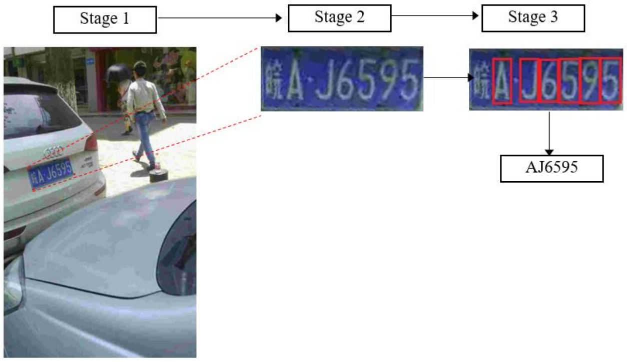


Fig. 1. Overview of the proposed framework.

Stage 1: License plate detection. As shown in Fig. 1, license plate detection aims to locate the four corner points of each license plate. For this purpose, this paper employs a lightweight deep CNN structure used for human pose estimation [14] and modifies it for predicting the four corner points of license plates.

Stage 2: Perspective correction. Perspective deformation images are corrected by applying perspective transformation. First, four corner points are predicted by the license plate detection network. Then, the homography between the camera and the license plate is recovered. Finally, the homography is used to warp the detected license plate into a rectified image as shown in Fig. 1.

Stage 3: Character recognition. For reading license plate characters, this paper considers character recognition as character detection problem and designs a lightweight character detection network that predicts all license plate characters on the detected license plates. The character detection network can predict license plate characters without depending on license plate layouts.

IV. Implementation Details

Fig. 2 illustrates the detailed pipeline of the proposed model. Input images are fed into a fast and efficient license plate detection network for predicting the four corner points of license plate (i.e., top-left, bottom-left, top-right, and bottom-right corners). Based on the detected corner points, perspective transformation is performed to align the detected license plate. Finally, a lightweight deep CNN network is employed to predict license plate characters. The following subsections illustrates the details of each stage.

A. License Plate Detection

This paper uses CenterNet [14] for extracting license plate regions and the corresponding corner points from input images. CenterNet considers the center point of a bounding box as an instance and uses this keypoint to predict the dimensions and offsets of the box. CenterNet strikes a desirable balance between precision and speed and is highly customizable and extensible. It can be easily extended to multiple computer vision tasks including 3D object detection, object tracking, human pose estimation, and many others. In this paper, CenterNet is used to predict the four corner points of license plates (i.e., top-left, bottom-left, top-right and bottom-right corners). Based on the predicted corner points, perspective correction is performed to get rectified license plate images. For this purpose, this paper employs the CenterNet structure used for human pose estimation [14] and modifies it for corner point estimation. The detailed pipeline of the license detection model based on CenterNet is shown in Fig. 3. Consider an input image $I \in \mathbb{R}^{H\times W\times 3}$, where W and H represent the width and height of the input image, respectively, a fully convolutional encoder-decoder architecture is first used to produce feature representations from input images and generate output results. Three heads are produced after one forward pass from the feature extraction network as shown in Fig. 3 (i.e., keypoint heatmap head, corner location head, and corner offset head). All the heads are predicted with the same dimensions (i.e., height and width) $(H/s, W/s)$, where s represents the output stride ($s = 4$ in this paper).

1) Feature extractor. This paper adopts RestNet-50 [15] for feature extraction. ResNet blocks are then augmented with three up-convolutional layers to incorporate higher resolution output feature maps. In addition, a 3×3 deformable convolutional layer is used before each up-sampling layer.

2) Keypoint heatmap head. Keypoint heatmap head is used for predicting the center point of license plates. In this paper, the keypoint heatmap $\hat{K} \in \mathbb{R}^{\frac{H}{S}\times\frac{W}{S}\times 1}$ has one channel since only one class is predicted by the license plate detection network. After one forward pass, a Sigmoid layer is utilized on the keypoint heatmap, and the calculated value at each keypoint is viewed as the certainty score for it being the center of the license plate.

3) Corner locations head. Corner location head predicts the four corner locations of license plate (i.e., top-left, bottom-left, top-right, and bottom-right corners). Each corner is considered as a 2-dimensional property of the center keypoint and parameterized by an offset to the center keypoint. The dimensions of this head are $(\frac{H}{S}\times\frac{W}{S}\times 8)$.

4) Corner offset head. The Corner offset head is employed to rectify the quantization error resulting from the down-sampling of the input. After one forward pass, the coordinates of predicted center keypoints are mapped to a higher resolution input image. This results in a deviation in values because the original image coordinates are whole numbers, whereas the actual center points ought to be decimal numbers. As a result, the local offsets $\hat{O} \in \mathbb{R}^{\frac{H}{S}\times\frac{W}{S}\times 2}$ are predicted for each center point to recover the discretization error.

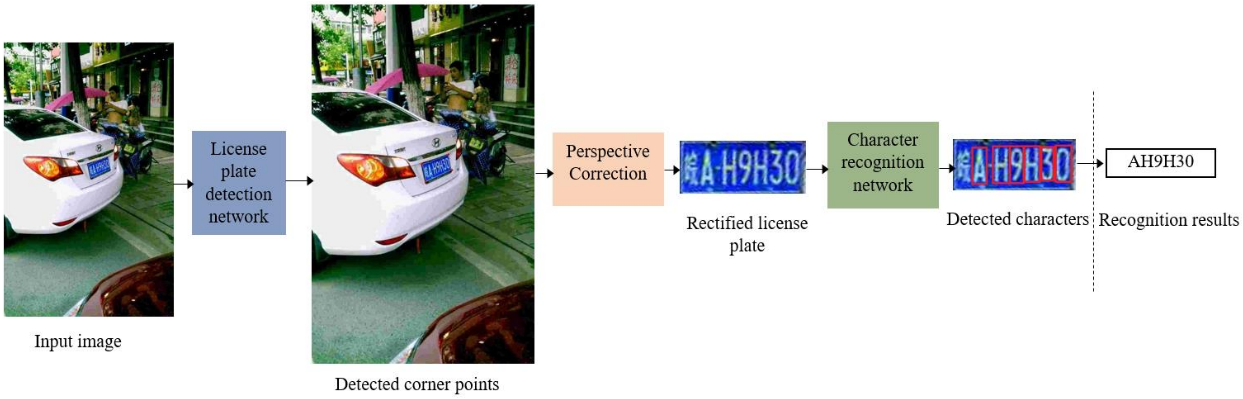


Fig. 2. The detailed structure of the proposed method.

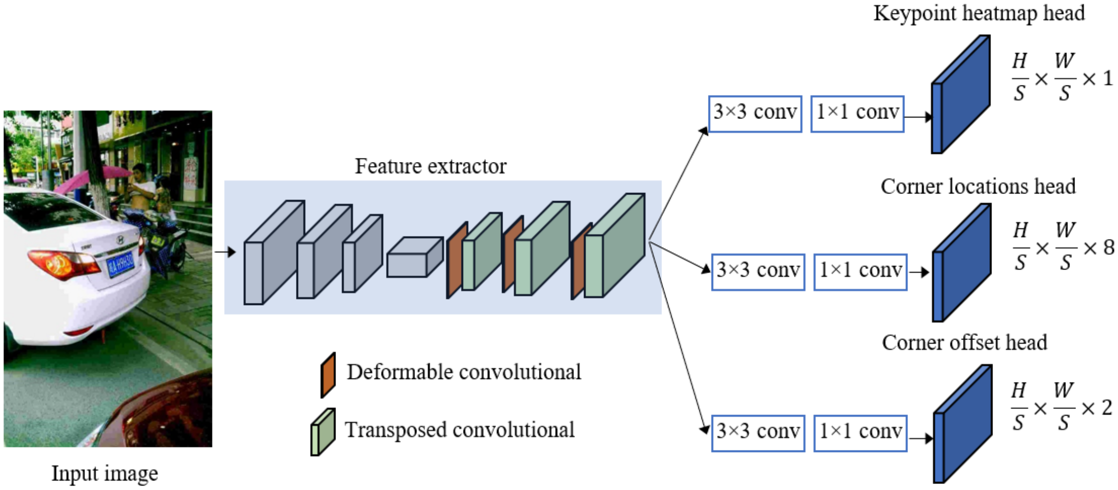


Fig. 3. The architecture of the license plate detection network.

B. Perspective Correction

As license plates can sometimes be difficult to read due to the viewpoint, perspective correction is performed to align the detected license plates. Based on the four corner points generated by the license plate detection network, the homography between the camera and the license plate is first recovered. Then, the homography is used to warp the detected license plate into a rectified image as shown in Fig. 1. To be more specific, based on the detected corner points from input image, this paper first identifies x_{max} and y_{max}, which represent the maximum horizontal and vertical distances between the corner points. Then four corresponding vertices of rectified image are calculated as follow:

$$(tl, tr) = ((0,0), (x_{max} - 1,0)) \quad (1)$$

$$(br, bl) = ((x_{max} - 1, y_{max} - 1), (y_{max} - 1,0)) \quad (2)$$

where tl, tr, br, bl represent the top-left, top-right, bottom-left, and bottom-right corners of the rectified license plate.

Following [16], the perspective transformation matrix H is calculated from the detected corner points and corresponding vertices as follow:

$$\begin{bmatrix} t_i x_i' \\ t_i y_i' \\ t_i \end{bmatrix} = H. \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (3)$$

where:

$$dst(i) = (x_i', y_i');\ src(i) = (x_i, y_i), i = 0, 1, 2, 3 \quad (4)$$

and

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (5)$$

Finally, the rectified license plate region is formed as follow:

$$dst(x, y) = src\left(\frac{H_{11}x + H_{12}y + H_{13}}{H_{31}x + H_{32}y + H_{33}}, \frac{H_{21}x + H_{22}y + H_{23}}{H_{31}x + H_{32}y + H_{33}}\right) \quad (6)$$

C. Character Recognition

Character recognition aims to identify each character on the rectified license plates. For this purpose, this paper considers character recognition as character detection problem and designs a lightweight character detection network that predicts each license plate character without depending on license plate layouts (i.e., license plates of single-line or double-line text). Specifically, the lightweight character detection network is trained to detect 35 classes (i.e., ‘A-Z’, ’0-9’, the digit ‘0’ is recognized jointly with the letter ‘O’) based on the rectified license plates as well as the bounding box and class of each character as inputs. In the case of Chinese license plates, the initial symbol is a Chinese character that signifies the province. As stated in [17] and [7], the character detection network proposed in this work has not been trained to identify Chinese characters because assigning the category to such characters is not a straightforward task.

Table I showcases the design of the suggested lightweight character detection network. The design of the network is influenced by the Fast-YOLOv4 model [18], which is a tiny deep neural architecture that obtains very fast detection speed without sacrificing much accuracy. As shown in Table I, 3×3 convolution layers are used to extract features from previous layers followed by 1×1 convolution layers for reducing the feature channels. In addition, max pooling layers are used to decrease the feature dimensions. The number of channels is multiplied by two following each max pooling layer. Following [9], [19], this paper applies detection head at different scales to predict license plate characters. Specifically, character prediction is performed at layer 13 and layer 20, where the output size is 24×8 and 48×16, respectively. This detection approach is crucial for character recognition because the characters on the license plate may take up either a small or large area of the license plate region, as depicted in Fig. 4. It is worth mentioning that the proposed license plate character recognition system accurately detects and identifies license plates with either single-line or double-line text as it predicts all characters on the rectified license plate.

TABLE I. THE STRUCTURE OF THE PROPOSED LIGHTWEIGHT CHARACTER DETECTION NETWORK

Layer	Type	Filter size, stride	Input size	Output size
0	Conv	3×3×32,1	384×128×3	384×128×32
1	Max-pool	2×2, 2	384×128×32	192×64×32
2	Conv	3×3×64,1	192×64×32	192×64×64
3	Max-pool	2×2, 2	192×64×64	96×32×64
4	Conv	3×3×128,1	96×32×64	96×32×128
5	Max-pool	2×2, 2	96×32×128	48×16×128
6	Conv	3×3×256,1	48×16×128	48×16×256
7	Conv	1×1×128,1	48×16×256	48×16×128
8	Conv	3×3×256,1	48×16×128	48×16×256
9	Max-pool	2×2, 2	48×16×256	24×8×256
10	Conv	3×3×512,1	24×8×256	24×8×512
11	Conv	1×1×256,1	24×8×512	24×8×256
12	Conv	3×3×512,1	24×8×256	24×8×512
13	Conv	1×1×45,1	24×8×512	24×8×45
14	Detection			
15	Route [11]			24×8×256
16	Conv	1×1×256,1	24×8×256	24×8×256
17	Upsample	2×	24×8×256	48×16×256
18	Route [17, 6]			48×16×512
19	Conv	3×3×512,1	48×16×512	48×16×512
20	Conv	1×1×45,1	48×16×512	48×16×45
21	Detection			

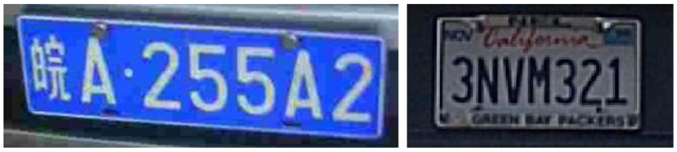


Fig. 4. License plate characters may occupy either small (Right) or large portion (Left) on the license plate region depending on type of license plate.

V. Experimental Results

All experiments were carried out on a computer equipped with an Intel Core i7-10700 CPU, a single NVIDIA GeForce GTX 1080Ti GPU, and 64GB of RAM. All models are designed and evaluated under the framework of PyTorch [20] and mmdetection [21].

A. Dataset and Evaluation Metrics

To assess the proposed method, this paper evaluates experiments on two extensive public datasets: CCPD [22] and AOLP [23].

CCPD [22] consists of 290k images captured under diverse illuminations, environments, and weather conditions. This dataset is more challenge than other datasets for license plate recognition since each image is captured from different positions and angles, which makes license plates have arbitrary direction. The dataset provides sufficient annotations for training the proposed model, including bounding boxes, vertices of each license plate, and license plate characters. Images in the dataset have the resolution of 720 (width) × 1160 (height) × 3 (channels). Following [22], this paper employs 100k images of CCPD-Base subset for training both detection and character recognition network. The remaining 100k images from the CCPD-Base subset and the 80k images from the CCPD-DB, CCPD-FN, CCPD-Rotate, CCPD-Tilt, CCPD-Weather, and CCPD-Challenge subsets are utilized for testing. Additionally, the CCPD dataset also includes the CCPD-Characters subset, consisting of over 1000 individual images for every possible license plate character. This paper utilizes the CCPD-Characters subset for further training the license plate recognition network.

AOLP (Application-Oriented License Plate database) [23] contains 2049 images. The images in this dataset are classified into three categories based on the capturing conditions: access control (AC), traffic law enforcement (LE), and road patrol (RP). The AC subset consists of 681 images of license plates captured in scenarios where vehicles move through a fixed passage at a slower pace or come to a complete stop. The LE subset consists of 757 images of license plates captured by a roadside camera during instances of traffic law violations. The RP subset includes 611 images of license plates captured from vehicles with random viewpoints and distances, making it more challenging for license plate recognition due to the heavily distorted license plates. Since the AOLP dataset only provides annotations for the coordinates of license plate bounding boxes and numbers, this paper manually annotates the four corners of each license plate. In line with [11], this paper trains on the LE and RP subsets and uses the RP subset for testing.

For the evaluation metric, this paper calculates the accuracy of license plate recognition by dividing the number of correctly recognized license plates by the total number of license plates in the test set. A recognition is considered correct only if the IoU_{poly} is greater than 0.5 and all characters have been correctly recognized. Here, IoU_{poly} is calculated as follow:

$$IoU_{poly} = \frac{DP \cap GP}{DP \cup GP} \quad (7)$$

where DP is the detected polygon of license plate and GP is the ground truth polygon of the license plate. DP and GP are calculated based on the detected corner points and ground truth corner points, respectively.

B. Results on CCPD

Table II provides recognition results of the proposed approach and recent approaches on the CCPD dataset. The results demonstrate that the proposed approach obtains the best recognition accuracy on most of the subsets. To be more specific, the proposed model obtains recognition accuracy at 99.7%, 99.2%, 99.1%, 99.6%, and 99.6% on CCPD-Base, CCPD- DB, CCPD- FN, CCPD- Rotate, and CCPD- Tilt, respectively, which outperform all previous methods, including method proposed by Zhang et al. [3]. For CCPD- Challenge subset, method proposed by Zhang et al. [3] obtains the best recognition accuracy. Since CCPD-Challenge subset comprises the most difficult images for the recognition of license plates, the simple and efficient license plate detection network cannot locate some license plates (Fig. 6), which leads to wrong recognition results by the character recognition network. In the future, this paper will investigate more effective fusion strategies to enhance the feature representation of the license plate detection network, which would improve detection results. It is noteworthy that the majority of the comparison methods in Table II determine the recognition results by setting the IoU threshold to 0.6. Additionally, it is observable that the proposed method attains the most significant improvements in the CCPD-Rotate and CCPD-Tilt subsets. Specifically, the proposed network improves recognition accuracy by 3.2% and 2% on CCPD-Rotate and CCPD-Tilt subsets compared with that of model proposed by Zhang et al. [3]. Given that the CCPD-Rotate and CCPD-Tilt subsets contain images with significant perspective distortion, these results show that the proposed model excels at detecting and recognizing license plates that have undergone distortion or rotation. For recognition speed, since the proposed model is designed based on fast and efficient architectures, it obtains the fastest recognition speed among comparing methods. To be more specific, the proposed model needs 8.3ms for processing an image based on single NVIDIA GeForce GTX 1080Ti GPU. The results indicate that the proposed model is efficient and well-suited for real-time applications. As depicted in Fig. 5, this study showcases the recognition results of the proposed approach on the CCPD dataset, including the detection of the four corners of the license plates, the rectified license plates after perspective correction, and the recognition of the characters on the license plates. It is evident that the proposed model performs effectively under various conditions. Fig. 6 shows some failure cases where the proposed model cannot locate license plates in challenging environments or fails to recognize some similar license plate characters.

TABLE II. THE RECOGNITION RESULTS OF THE PROPOSED MODEL AND OTHER MODELS AS EVALUATED ON THE CCPD DATASET

Method	Recognition Accuracy (%)							Inference Time (ms)
	Base	DB	FN	Rotate	Tilt	Weather	Challenge	
SSD [24] + HC [25]	98.3	96.6	95.9	88.4	91.5	87.3	83.8	25.6
YOLO9000 [8] + HC [25]	98.1	96.0	88.2	84.5	88.5	87.0	80.5	23.8
Faster-RCNN [26] + HC [25]	97.2	94.4	90.9	82.9	87.3	85.5	76.3	57.6
TE2E [17]	97.8	94.8	94.5	87.9	92.1	86.8	81.2	310
LPRNet [27]	97.8	92.2	91.9	79.4	85.8	92.0	69.8	17.8
RPnet [22]	98.5	96.9	94.3	90.8	92.5	87.9	85.1	11.7
Zou et al. [13]	99.3	98.5	98.6	92.5	96.4	99.3	86.6	-
DAN [28]	98.9	96.1	96.4	91.9	93.7	95.4	83.1	19.3
Zhang et al. [3]	99.6	98.8	98.8	96.4	97.6	**98.5**	**88.9**	24.9
Proposed method	**99.7**	**99.2**	**99.1**	**99.6**	**99.6**	**98.5**	88.5	8.3

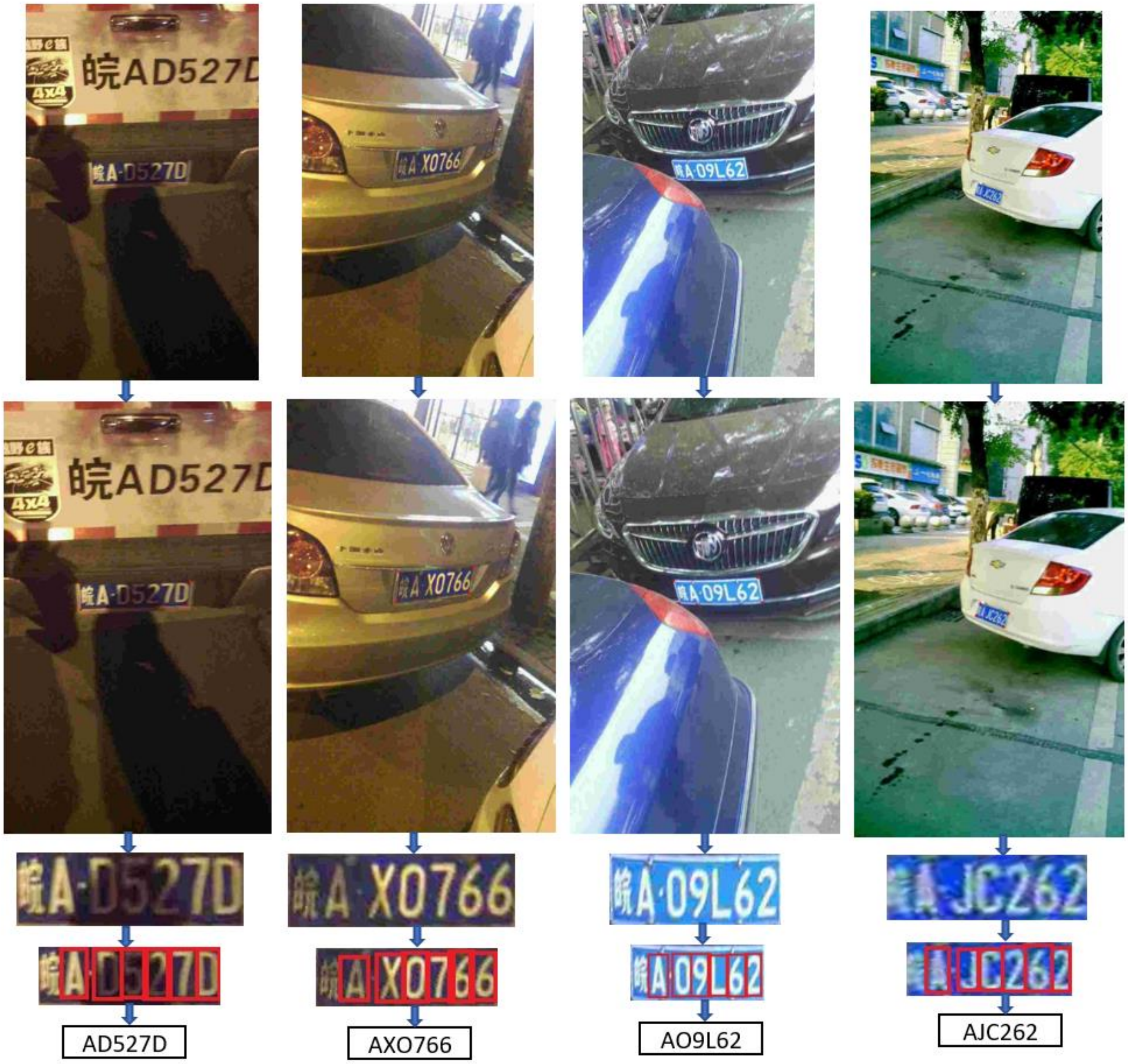


Fig. 5. Visual representation of the recognition outcomes of the proposed model on the CCPD dataset.

TABLE III. COMPARISONS OF RECOGNITION ACCURACY ON THE AOLP DATASET

Method	Recognition Accuracy (%)
Li et al. [29]	88.4
TE2E [17]	83.6
Zhang et al. [3]	91.9
Sergio et al. [6]	98.4
Zou et al. [13]	93.4
Proposed method	**98.8**

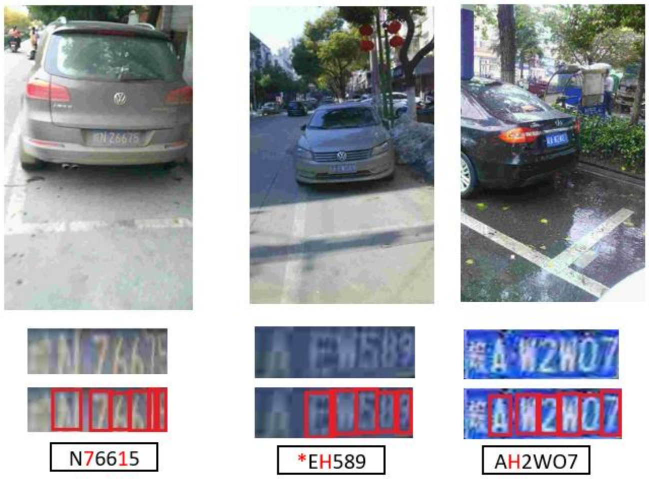


Fig. 6. Some examples of failure cases.

C. Results on AOLP

For the AOLP dataset, AOLP-RP subset is employed to evaluate the proposed model since this subset is more challenging (most of images contain severe perspective deformation license plates). Table III presents the comparison of recognition accuracy on the AOLP dataset. The proposed model emerges as the top performer in terms of recognition accuracy on the AOLP dataset, outperforming other methods. Specifically, the recognition accuracy of the proposed model surpasses that of the method proposed by Sergio et al [6] by 0.4%. This result further strengthens the claim of the proposed model's capability in recognizing license plates that have an irregular shape.

VI. CONCLUSION

This study presents a CNN-based approach for detecting and recognizing license plates with irregular shapes in complex real-world images. The proposed model employs a CenterNet-based CNN structure to predict the four corners of the license plates, followed by perspective correction to align the detected license plates. For character recognition, a YOLO-based segmentation-free model is designed to predict the characters on the license plate. The effectiveness of the proposed method is verified through experiments on the CCPD and AOLP datasets. Specifically, experimental results on two datasets show that the proposed method obtains the best recognition accuracy with the fastest recognition speed. This result demonstrates that the proposed method is highly suitable for intelligent traffic management applications that require real-time processing. In the future, the study intends to investigate additional fusion techniques for extracting more discriminative features from the input images, which can enhance the accuracy of the license plate detection network.

REFERENCES

[1] Anagnostopoulos, Christos-Nikolaos E., Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas. "License plate recognition from still images and video sequences: A survey." *IEEE Transactions on intelligent transportation systems* 9, no. 3 (2008): 377-391.

[2] Du, Shan, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy. "Automatic license plate recognition (ALPR): A state-of-the-art review." *IEEE Transactions on circuits and systems for video technology* 23, no. 2 (2012): 311-325.

[3] Zhang, Linjiang, Peng Wang, Hui Li, Zhen Li, Chunhua Shen, and Yanning Zhang. "A robust attentional framework for license plate recognition in the wild." *IEEE Transactions on Intelligent Transportation Systems* 22, no. 11 (2020): 6967-6976.

[4] Chen, Song-Lu, Chun Yang, Jia-Wei Ma, Feng Chen, and Xu-Cheng Yin. "Simultaneous end-to-end vehicle and license plate detection with multi-branch attention neural network." *IEEE Transactions on Intelligent Transportation Systems* 21, no. 9 (2019): 3686-3695.

[5] Nguyen, Hoanh. "Predicted anchor region proposal with balanced feature pyramid for license plate detection in traffic scene images." *Complexity* 2020 (2020).

[6] Silva, Sergio Montazzolli, and Claudio Rosito Jung. "License plate detection and recognition in unconstrained scenarios." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 580-596. 2018.

[7] Laroca, Rayson, Luiz A. Zanlorensi, Gabriel R. Gonçalves, Eduardo Todt, William Robson Schwartz, and David Menotti. "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector." *IET Intelligent Transport Systems* 15, no. 4 (2021): 483-503.

[8] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271. 2017.

[9] Redmon, Joseph and Ali Farhadi. “YOLOv3: An Incremental Improvement.” *ArXiv* abs/1804.02767 (2018).

[10] Montazzolli, Sérgio, and Claudio Jung. "Real-time brazilian license plate detection and recognition using deep convolutional neural networks." In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pp. 55-62. IEEE, 2017.

[11] Wang, Yi, Zhen-Peng Bian, Yunhao Zhou, and Lap-Pui Chau. "Rethinking and designing a high-performing automatic license plate recognition approach." *IEEE Transactions on Intelligent Transportation Systems* (2021).

[12] Wang, Zhichao, Yu Jiang, Jiaxin Liu, Siyu Gong, Jian Yao, and Feng Jiang. "Research and Implementation of Fast-LPRNet Algorithm for License Plate Recognition." *Journal of Electrical and Computer Engineering* 2021 (2021).

[13] Zou, Yongjie, Yongjun Zhang, Jun Yan, Xiaoxu Jiang, Tengjie Huang, Haisheng Fan, and Zhongwei Cui. "A robust license plate recognition model based on bi-lstm." *IEEE Access* 8 (2020): 211630-211641.

[14] Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." *arXiv preprint arXiv:1904.07850* (2019).

[15] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[16] Andrew, Alex M. "Multiple view geometry in computer vision." *Kybernetes* (2001).

[17] Li, Hui, Peng Wang, and Chunhua Shen. "Toward end-to-end car license plate detection and recognition with deep neural networks." *IEEE Transactions on Intelligent Transportation Systems* 20, no. 3 (2018): 1126-1136.

[18] A. Bochkovskiy, “Fast-YOLOv4,” https://github.com/AlexeyAB/darknet/blob/master/cfg/yolov4-tiny.cfg, accessed: 2021-03-03.

[19] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection.” ArXiv abs/2004.10934 (2020).

[20] Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch." (2017).

[21] Chen, Kai, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun et al. "MMDetection: Open mmlab detection toolbox and benchmark." *arXiv preprint arXiv:1906.07155* (2019).

[22] Xu, Zhenbo, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang. "Towards end-to-end license plate detection and recognition: A large dataset and baseline." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 255-271. 2018.

[23] Hsu, Gee-Sern, Jiun-Chang Chen, and Yu-Zu Chung. "Application-oriented license plate recognition." *IEEE transactions on vehicular technology* 62, no. 2 (2012): 552-561.

[24] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.

[25] Špaňhel, Jakub, Jakub Sochor, Roman Juránek, Adam Herout, Lukáš Maršík, and Pavel Zemčík. "Holistic recognition of low quality license plates by CNN using track annotated data." In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-6. IEEE, 2017.

[26] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[27] Zherzdev, Sergey and Alexey Gruzdev. “LPRNet: License Plate Recognition via Deep Neural Networks.” *ArXiv* abs/1806.10447 (2018).

[28] Wang, Tianwei, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. "Decoupled attention network for text recognition." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, pp. 12216-12224. 2020.

[29] Li, Hui and Chunhua Shen. “Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs.” *ArXiv* abs/1601.05610 (2016).

[30] Nguyen, Hoanh. "An Efficient License Plate Detection Approach Using Lightweight Deep Convolutional Neural Networks." *Advances in Multimedia* 2022 (2022).

Proxy Re-encryption Scheme based on the Timed-release in Edge Computing

Yifeng Yin[1], Wanyi Zhou[2], Zhaobo Wang[3], Yong Gan[4], Yanhua Zhang[5]
School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China[1, 2, 3, 5]
Zhengzhou Institute of Technology, Zhengzhou, China[4]

***Abstract*—With the growth of Industrial Internet, various types of data show explosive growth. Data is being moved from the cloud to the edge for computing more frequently and edge computing becomes an important factor affecting the deep application of Industrial Internet platforms. However, the security issue of data transmission sharing is not addressed. Therefore, this paper proposes a security scheme based on timed-release, multi-dimensional virtual permutation, and proxy re-encryption(PRE), to protect the confidentiality of the data during the transmitter; symmetric cryptography is employed to encrypt the transmission data. At the same time, a time server is used, making it impossible for data receivers to get the information about the data before the specified time arrives, it solves the application scenario of data transmission and sharing with timed-release requirements and the efficiency of a large number of data is solved, and the security of data is improved. Finally, the security of the scheme was proved theoretically. Compared to existing PRE schemes, this scheme adds timed-release controlled access, has resistance to ciphertext attacks and end-to-end security features, and uses fewer bilinear pair operations in the algorithm. The performance was tested experimentally and the results show that the scheme improves efficiency while ensuring security and has significant advantages in terms of data security and private data protection.**

Keywords*—*Timed-release; edge-computing; multi-dimensional virtual permutation; proxy re-encryption; symmetric encryption

I. Introduction

Information technologies like big data, cloud computing, and artificial intelligence have developed so quickly in recent years, new ways of manufacturing and structuring are driving the intellectual conversion of the global industrial system [1]. Thus, with the benefits of both the industrial and internet revolutions combined, the industrial internet arose. As a result of big data and artificial intelligence, nowadays, if there is a lack of effective data, massive industrial data will be buried, and the Industrial Internet will lose its value of existence[2], [3]. At the same time, with the accelerating pace of the construction of Industrial Internet platforms, users' demand for security is more urgent. It has gradually evolved from the security of simple industrial equipment use to the security of data stored in the equipment and data transmitted in the network, from tangible security to intangible security[4], [5]. In contrast to the established cloud computing model, edge computing model makes up for the disadvantage of cloud computing away from terminal devices, it is a fundamental technology in the Industrial Internet and helps the Industrial Internet evolve[6]. Edge computing needs to provide the new capabilities of "device openness and data sharing" needed in industrial transformation and upgrading, and reliability-wise, edge computing can satisfy the industrial Internet's development requirements[7], [8]. Data may pass via numerous message-forwarding nodes in the edge computing environment before it reaches the collaborating nodes instead of being delivered directly from one device to another[9]. The data is conveyed by a third party, and if the communicated data is maliciously altered, it will have a significant impact on data security issues. Therefore, in such a scenario, it is crucial to guarantee the security of data transmission between users.

Proxy re-encryption technology is an encryption cryptosystem with ciphertext conversion function, which does not need data decryption in this process, so it can be used to achieve data transmission security[10]. PRE technology ensures the secure access and sharing of data without revealing any relevant information about the plaintext[11], [12]. However, in some application scenarios with timed-release requirements, this technology is not perfect, and the ciphertext generated by PRE needs to have the characteristics of timed-release[13]. Time-dependent encryption includes methods such as timed-release encryption, which can be used in time-sensitive scenarios[14]. Even the selected receiver is unable to decrypt the ciphertext prior to the semi-trusted time server opening the trapdoor linked to the publisher's chosen release time[15], which can be used as a cryptographic primitive to specify the future decryption time, and there are many scenarios in real life where the decryption time needs to be specified in advance[16], [17]. Therefore, PRE technology and timed-release encryption work better together to improve the confidentiality and flexibility of the scheme.

PRE was first proposed by Blace[18], it enables a partially trusted proxy to convert the publisher's ciphertext into the receiver's ciphertext, and increases the flexibility of data sharing. Since it was proposed, numerous PRE techniques have been put forth. Peng et al. [19]presented an identity-based conditional broadcast PRE system that enables users to communicate encrypted data with other users using the fine-grained approach, allows users to generate broadcast ciphertexts for numerous recipients, and share their encrypted data to multiple recipients in batches. Kaitai et al.[20] proposed a conditional proxy broadcast re-encryption method with timed-release, which allows the delegator to delegate the decryption right to multiple delegates in a fine-grained manner without losing security, and the delegator has the ability to schedule the delegation of the decryption right of the broadcast encryption to a certain group of receivers in addition to delegating it. Massive privacy-sensitive data transmitted to edge nodes may attract attacks from network attackers, so data leakage and attacks are a great loss and danger for the Industrial Internet.

To address the aforementioned issues, this work proposes the timed-release based PRE scheme for edge computing. Since the efficiency of public key cryptography is not as high as the symmetric encryption algorithm, it is not suitable for the encryption of large data files. Therefore, a multi-user data-sharing scheme is designed by combining symmetric cryptography, time-release encryption and PRE techniques, and depends on a multi-dimensional virtual substitution mechanism. It effectively protects this publisher's privacy information, prevents malicious tampering and theft, meets the requirements of timeliness and multi-user sharing data, ensures the security of data transmission, reduces the time cost, and improves the operation efficiency.

The rest of the paper is organised as follows. In Section II, we describe the constructed scheme model. In Section III, we describe the design steps of the scheme in detail. In Section IV, we perform security analysis and performance evaluation of the scheme. Section V is discussion. Section VI concludes the paper.

II. Model Design

The traditional PRE scheme uses an asymmetric cryptosystem, which requires a large amount of computation, large consumption and waste of resources for large data files. And in some application scenarios, it is necessary to meet the requirements of time. Thus, a PRE scheme based on timed-release for edge computing is created to ensure the security of data information and address the issue of timeliness. According to the actual application of data information sharing, a multi-user data sharing model is established as shown in Fig. 1.

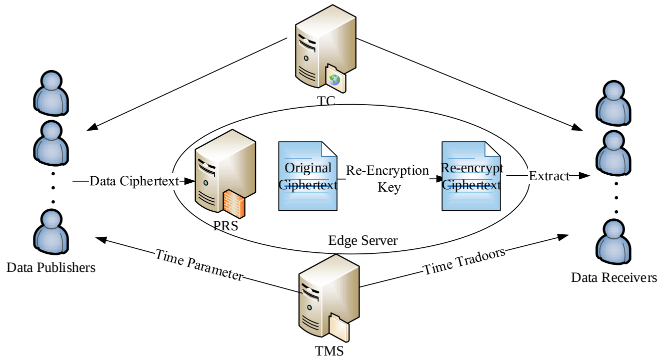


Fig. 1. Multi-user data sharing model.

Both data publishers and data receivers can access data in the network and interact with the proxy re-encryption server (PRS), trusted center (TC) and time server (TS) to create and access data. The third trusted time server generates time parameters according to the requirements, which is used to encrypt the plaintext. The generated time trapdoor algorithm can be used to confirm that the information has not been tampered with. The data publishers upload the data cipher text to the proxy server and then encrypt the original cipher text into the re-encrypted ciphertext by the re-encryption key, from which the data receivers can extract the re-encrypted cipher text without dumping and storing the data that has already been used.

III. Scheme Design

The symbols used in the scheme of this paper and their corresponding meanings are shown in Table I.

TABLE I. Symbols and Meanings in the Text

Symbol	Mention
sk_{ts}/pk_{ts}	Time server's private and public key
sk_i/pk_i	Data publisher $i's$ private/public key
sk_j/pk_j	The private/public key by data receiver j
T_G	Timed-release key
Sk_i'	Initial key array for data publisher i
$Sk_i[n]$	Key control array for data publisher i
$IVF(i)$	Virtual permutation function missing i-th argument
$MVPF$	The complete virtual permutation function
$MITE^{(m)}$	A virtual iteration function that iterates m times
C_T	Original ciphertext
C_S	Session key ciphertext
C_P	Re-encryption of ciphertext
$RK_{i\to j}$	Proxy re-encryption key for users i to j

Multi-user data sharing through timed-release proxy re-encryption algorithm can solve the time limitation problem in data sharing compared to traditional PRE schemes. This topic uses edge server as a tool to realize data sharing, the time server is introduced to create the release time parameters, and the multi-dimensional virtual replacement mechanism is combined in the symmetric key generation stage, which can improve confidentiality and efficiency. According to the multi-user data sharing model, the scheme can be realized as described below.

A. System Initialization

Choose a prime q , a finite cyclic group G and G_T of order q , a bilinear mapping $e: G \times G_T \to G_T$, where g is the generating element of G . Then define the hash functions $H_0: \{0,1\}^* \to G$, $H_1: \{0,1\} \to G$, $H_2: G_T \to \{0,1\}^L$, and randomly select an element γ from Z_q^* as the private key for the time server $sk_{ts} = \gamma$, then the public key for the time server $pk_{ts} = g^{\gamma}$ can be calculated.

B. Key Generation

Randomly select the elements x_i and x_j from Z_q^* as the private keys of data publishing user i and data receiving user j respectively, and make $sk_i = x_i$ and$sk_j = x_j$, then the public keys corresponding to data publishing user and data receiving user are $pk_i = g^{x_i}$ and $pk_j = g^{x_j}$ respectively.

C. Generate Time Trapdoor

Firstly, the time parameter $T \in \{0,1\}^*$ can be generated by the time server to display the publishing time. Then through the calculation of the private key sk_{ts} by the time server and the time parameter T, this time trapdoor can be obtained, corresponding to the timing release key T_G, as shown in the formula.

$$T_G = H_0(T)^{\gamma} \tag{1}$$

D. Session Key Generation

This phase first requires constructing a multi-dimensional key space, then displacing the security subsystem, constructing a multi-dimensional virtual substitution function, and iterating over the security subsystem it displaces in a certain order to finally obtain a session key that conforms to the security rules. This process uses a lightweight cryptographic algorithm based on stream ciphers, and the specific steps are described as shown below.

1) Constructing a multi-dimensional key space: The private key $\mathrm{Sk_i}$ of data publishing user i and its private key array $\mathrm{Sk_i'}$ can be hashed and mapped to generate a key control array $\mathrm{Sk_i[n]}$ by the following formula. Then it is sent to other data publishing users, each data publishing user can get an incomplete virtual iterative function IVF(i) that lacks its own private key, where the user's private key array $\mathrm{Sk_i'} = [\mathrm{k_{i1}}, \mathrm{k_{i2}}, \dots, \mathrm{k_{in}}]$, because it has no actual ability of control function, so it does not pass to other users.

$$Sk_i[n] = \{SK_i \oplus \textstyle\sum_{a=1}^{n} H(Sk_i'[a])\} \tag{2}$$

Data publishing user i substitutes the key control array $Sk_i[n]$ into the incomplete virtual iterative function to obtain a multi-dimensional virtual iterative function $MVPF$ with complete key parameters, mapped into an n-dimensional spatial network with equal probability of security subsystems mapped into the n-dimensional space to establish an n-dimensional key space, where each small space maps a security subsystem as shown in the formula.

$$S = MVPF(Sk_1[n], Sk_2[n], \cdots, Sk_i[n], \cdots, Sk_n[n]) \tag{3}$$

2) Symmetric key generation: If each data publishing user randomly selects m key elements in the key control array, then the m secure subsystems can be noted as $\tau_1, \tau_2, \cdots, \tau_m$, respectively, and a secure symmetric key K can be obtained by iterative operations in a certain order according to the formula.

$$K = \tau_m(MITE^{(m-1)}) \tag{4}$$

E. Transformation Key Generation

The private key sk_i by data publishing user i, the public key pk_j by data receiving user j, through hash mapping, can get RK_1. Then the time server generates the publishing time parameter T and the public key pk_{ts} through the calculation of the following formula, RK_2 can be obtained, and finally the conversion key is obtained by the calculation of the formula, which can also be called this re-encryption key $RK_{i\to j}$, and put online via the proxy server.

$$RK_1 = sk_i \cdot (H_2(g^{\beta}, pk_j)^{\rho})$$

$$RK_2 = \varphi \oplus e(H_0(T), pk_{ts}) \tag{5}$$

$$RK_{i\to j} = (RK_1, RK_2)$$

F. Proxy Re-encryption

1) The original ciphertext generation: The data publishing user has the ability to safely encrypt the data plaintext M using the publicly available symmetric encryption algorithm based on the symmetric key generated in the previous session key phase, which can obtain the desired data ciphertext C_1. Then a random number $\beta \in Z_q^*$ and $\varphi \in G_T$ is chosen and the hash function $H_3: \{0,1\}^* \to Z_q^*$ is defined such that $\rho = H_3(M, \beta)$, and the ciphertext C_2 can be obtained after the calculation by the formula, and the ciphertext C_3 can be obtained through the time server's public key pk_{ts} and the published time parameter T. Finally, these obtained ciphertexts can be calculated by the following formula to get the required original ciphertext C_T and uploaded to the proxy server at the same time.

$$C_1 = SymEnc_K(M)$$

$$C_2 = (M||\beta) \oplus g^{H_2(g^{\beta}, \varphi)}$$

$$C_3 = \varphi \oplus e(H_0(T), pk_{ts})^{\rho} \tag{6}$$

$$C_T = H_1(C_1||C_2||C_3)$$

2) Session key ciphertext: This stage uses the encryption of the session key and then generates the key cipher, which can be used in the subsequent stage of using the proxy server for this PRE algorithm to generate this re-encrypted cipher, where the session key and the public key of the data publishing user are calculated by the following formula to obtain the ciphertext C_S of this session key and upload to the proxy server.

$$C_S = {pk_i}^{\beta} \cdot K \tag{7}$$

3) Data sharing: The PRE ciphertext C_P is obtainable by using that session key ciphertext C_S and the transformation key $RK_{i\to j}$, along with the time server's publication time parameter T. A proxy server that has some level of confidence performs this process, which does not disclose any message about the ciphertext, so the security of this scheme can be guaranteed.

$$C_P = C_S||e(H_1(T), RK_{i\to j})^{\beta} \tag{8}$$

The data recipient can get the plaintext M computing the decryption of that PRE ciphertext. Firstly, the PRE ciphertext C_P and the data receiver's private key sk_j are decrypted to obtain the session key K. Then the original ciphertext C_T is decrypted using the session key K to obtain the plaintext, as shown in the following formula.

$$K = Dec_{sk_j}(C_P)$$

$$M = Dec_K(C_T) \tag{9}$$

IV. Security Analysis and Performance Evaluation

A. Security Analysis

This scheme combines the multi-dimensional virtual permutation mechanism and timed-release encryption and uses the PRE algorithm to implement the transfer of data information, so as to solve the protection of data in the industrial Internet. The common attacks that can be dealt with and the specific analysis are as follows.

1) Confidentiality: In this program, data plaintext is converted into data ciphertext using symmetric encryption algorithm, which is stored in the proxy server, as well as using the public key to encrypt the symmetric key to generate key ciphertext. Because data information and session keys are transmitted in ciphertext throughout the communication process, a network attacker cannot decrypt its data information and restore the ciphertext to plaintext. To decrypt the ciphertext and acquire the plaintext data, only the data publisher and the data receiver are capable. If the proxy server tries to decrypt the data information, then it must get $RK_{i\to j}$ and pk_j, but the proxy server only has pk_j, as a result, it is unable to access information about the original data. This system thus achieves the security of private and sensitive data on the Industrial Internet by guaranteeing the secrecy of the data.

2) Replay attack protection: In this scheme, only within the set time validity period, the user who is receiving the data can get the plaintext data by decrypting the ciphertext that has been re-encrypted. The data-receiving user can get the corresponding reverse key only after getting the ciphertext that was re-encrypted and decrypting it, and because there is a certain time limit for the reverse key at this time, this reverse key can be saved only until this time arrives. However, when a network attacker performs a replay attack, the same reverse key can be obtained by decrypting the data message, but then it discards the message and does not deliver the data, and also cannot obtain the server's authentication, which can be used to ensure that the time trapdoor is always time-limited. Therefore, this scheme can resist the replay attack.

3) End to end: End-to-end security truly entails that the server won't be aware of the communication information shared by cooperating nodes, and the proxy server relies on the edge server, which $RK_{i\to j}$ has decided the conversion rule of decryption, so the server cannot convert the cipher text into other decryption combinations. Secondly, since the data information and session keys are always transmitted in cipher text during the communication, even the proxy server cannot decrypt them and get the data information. By decrypting the ciphertext that has been re-encrypted, only the users who publish data and those who receive it can access the plaintext data, so even if this server is attacked, this data information sent between users will not be disclosed. Therefore, the scheme satisfies end-to-end security.

4) Protection against collusion attack: In this scheme, the private key is secure against collusion attacks. The data receiver and the proxy cannot conspire to gain the private key of the data publisher, assuming that the proxy sends the re-encryption key to the data receiver for malicious collusion. Since $sk_i \cdot (H_2(g^{\beta}, pk_j)^{\rho})$ in β is random. Using the re-encryption key as a base, the data receiver cannot determine the data publisher's private key information. At the same time, the agent cannot conspire with the data publisher to get the recipient's private key, if the publisher of the data and the agent join together, because the re-encryption key $RK_{i\to j} = (RK_1, RK_2)$ does not have any private key information of the data recipient, and cannot get its private key through calculation, so the combination of the two cannot acquire the data receiver's private key. This proves that even if this proxy server and the data receiver or data publisher maliciously collude, they cannot get each other's private keys to decrypt the ciphertext without access rights, so this scheme resists the collusion attack.

5) Select ciphertext security: In arbitrary probabilistic polynomial time, the benefit of addressing the hypothetical issue of DBDH is negligible, so the scheme proposed in this study is to choose ciphertext attack security. Assuming the existence of attacker A and challenger C.

Firstly, A sends queries to C about system initialization, time trapdoor generation, key generation, session key generation, original ciphertext generation, session key ciphertext generation, transformation key generation and proxy re-encryption ciphertext generation, and then C returns the query results to A. C creates the corresponding public and private keys through this system initialization and key generation algorithm, as well as the time server's private key, and performs this re-encryption key algorithm on them to get the re-encryption key, then transmit it to A. A chooses two equal-length plaintexts M_0 and M_1 and sends them to C to challenge it. When A asks C for the original ciphertext generation algorithm, where the session key is kept secret. C randomly picks bits $d \in \{0,1\}$, computes the ciphertext $C^* = Encryption(T_G^*, pk_{ts}^*, K^*, M_d, T^*)$. A then returns a guess $d_1 \in \{0,1\}$ to C. At this point, if $d_1 = d$, the challenge is successful. Assuming that the probability superiority of A overcoming the challenge is ε and $\varepsilon = |\Pr[d_1 = d] - 1/2|$ is negligible. the probability that A guessing correctly in probabilistic polynomial time is negligible, then A fails the challenge, so it can be shown that the scheme of this paper is to choose ciphertext security.

B. Performance Evaluation

The scheme of this paper was analyzed in comparison with some relevant studies in recent years, and the following data were used to compare the time consumption with the literature, as shown in Table II.

Through the existing relevant references, the theoretical computational overheads of the schemes are compared with those of this paper, and the theoretical execution time comparison results of each scheme are shown in Table III. The time required for bilinear is much larger than that of other operations. Because the bilinear pairing used in reference [19] and [20] are more than those used in this scheme, the time used in these two schemes will be more. According to the comparison results of the theoretical time overhead in Table II, we can get that the scheme in this chapter is smaller than reference [19] and [20] in terms of computational overhead, so our proposed scheme has higher efficiency and also higher security, which shows that this scheme has significant advantages in security and efficiency.

TABLE II. THE EXECUTION TIME OF EACH OPERATION

Parameter	Description	Value(ms)
T_p	Bilinear pairing operation time	4.211
T_m	Point multiplication operation time	0.015
T_e	Exponentiation operation time	3.886
T_h	Hash operation time	0.0001
$T_{E/D}$	Symmetric encryption/decryption operation time	0.0046

TABLE III. COMPARISON OF TIME COMPLEXITY OF EACH SCHEME

Reference	Data Processing	Proxy Re-encryption	Sum
Peng [19]	$(n+2)T_e + 2T_p + 5T_m + (n+1)T_h$	$3T_m + 2T_p + 3nT_h + (n+4)T_e$	$(2n+6)T_e + 4T_p + 8T_m + (4n+1)T_h$
Kaitai[20]	$2nT_e + T_p + (2n+5)T_h$	$(n+1)T_e + 8T_p$	$(3n+1)T_e + (2n+5)T_h + 9T_p$
Zheng[21]	$(n+3)T_p + 2nT_e$	$nT_m + nT_p$	$(2n+3)T_p + nT_m + 2nT_e$
Proposed	$(n+2)T_h + T_p$	$nT_e + T_p + 2T_{E/D} + (n+1)T_h$	$2T_{E/D} + 2T_p + nT_e + (2n+3)T_h$

In order to confirm the effectiveness of the plan put forth in this paper, an experimental environment is built for the experimental testing of the algorithm. Experimenting with Charm-Crypto 0.5, PBC 0.5.14 library in Python 3.9 under a Linux system with an Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz and 16GB RAM. Simulation of the time spent comparing the literature with the solution proposed in this paper, and the simulation's outcomes are displayed in Fig. 2. After simulating the time required to implement each scheme. The results of the experiments reveal that the system suggested in this study has a much lower overall computing overhead than the research [19], [20] and [21] schemes.

For study [19], although the dot product operation is added compared to reference [20], it does not have much impact on the time overhead as it is negligible; it also reduces the bilinear pairing operation, which reduces the time overhead and improves the efficiency to some extent.

For study [20], although it uses more hashing operations compared to [21], it does not have as much time overhead as [21] because the execution time of the hashing operation is negligible. However, compared to the scheme in this paper, it still requires more time cost.

For research [21], it uses bilinear pairing operations, which require a lot of time to compute each bilinear pairing, and uses a lot of dot product operations and power operations. Although it uses less time consumption at the beginning, the time overhead increases as the number of users increases. Therefore, the most computationally expensive of these schemes.

Compared with the schemes proposed in the studies [19], [20] and [21], the time consumption of this scheme is the least. Although this paper uses one more symmetric encryption/decryption operation, it does not increase the time overhead because its execution time magnitude is much smaller than the other operations and negligible, and uses a small number of bilinear pairing operations and power operations, in a specific test of execution time overhead the lowest. This scheme uses lightweight operations, which reduces the overall time spent. Because this solution uses fewer bilinear operations, which are precisely the functions with the highest computational overhead, reducing the number of calls to bilinear operations in the algorithm has a clear advantage in terms of efficiency. The lightweight proxy re-encryption algorithm used in this scheme ensures secure data transfer and improves efficiency.

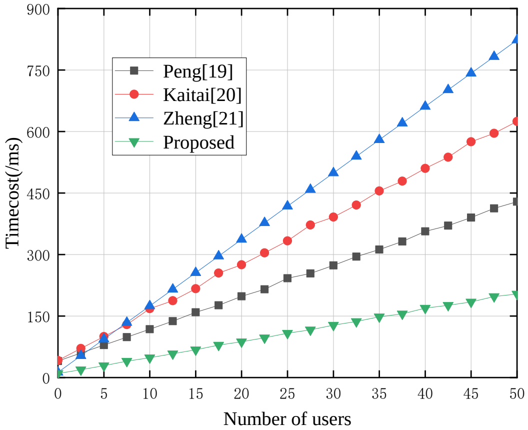


Fig. 2. Comparison of schemes' time cost.

V. DISCUSSION

The scheme in this paper introduces a third-party fully trusted time server to encrypt the plaintext by inserting a time attribute, which can only be decrypted by an authorised recipient within a set timeliness range to obtain the plaintext. Compared to existing PRE schemes, this scheme is more flexible, provides end-to-end data security and reduces the trust requirement for proxies. The security analysis shows that this scheme is resistant to ciphertext attacks and thus has higher security. The computational overhead comparison and simulation experiments show that this scheme can save a lot of transmission time and has better computational overhead performance and is more efficient.

VI. CONCLUSION

This paper analyzes the security needs for data transmission and sharing in an environment using edge computing. For the current PRE study is difficult to meet the efficiency and security for the transmission of many data files, and the requirements of time constraints, this paper proposes a PRE scheme based on timed-release, which offers efficient, flexible and secure data sharing services. The secure sharing of data information is completed by symmetric cryptographic encryption and proxy re-encryption, and time servers are introduced to solve the problem of timeliness and protect privacy and security. On the premise of ensuring data security and confidentiality, reducing time consumption overhead and improving the efficiency of communication, this scheme is appropriate for data sharing from larger files, which can be completed in a short time, and has great advantages and practical value.

ACKNOWLEDGMENT

This paper is supported in part by the Natural Science Foundation of Henan Province of China under Grant No.202300410508, the Natural Science Foundation of Henan Province under Grant No.222300420371, the Key Research Project of Higher Education of Henan Province under Grant No. 22A520047, the National Natural Science Foundation of China under Grant U1804263, and key foundation of Science and Technology Development of Henan Province under Grant No.142102210081.

REFERENCES

[1] W. Qin, S. Chen, and M. Peng, "Recent advances in Industrial Internet: insights and challenges," Digit. Commun. Netw., vol. 6, no. 1, pp. 1–13, Feb. 2020.

[2] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges," IEEE Commun. Surv. Tutor., vol. 19, no. 3, pp. 1504–1526, 2017.

[3] X. Zhang and X. Ming, "Implementation path and reference framework for Industrial Internet Platform (IIP) in product service system using industrial practice investigation method," Adv. Eng. Inform., vol. 51, p. 101481, Jan. 2022.

[4] L. Jun and L. Lan, "Research on Security Detection and Data Analysis for Industrial Internet," in 2019 Companion of the 19th Ieee International Conference on Software Quality, Reliability and Security (qrs-C 2019), Los Alamitos, 2019, pp. 466–470.

[5] L. Wang, Z. Ye, R. Zhang, J. Lin, F. Chen, and F. Tang, "The Growth Model of Industrial Internet Platform in Industrial 4.0," Wirel. Commun. Mob. Comput., vol. 2022, p. 5145641, Mar. 2022.

[6] M. Wei, X. Yang, J. Mao, and K. Kim, "Secure Framework and Security Mechanism for Edge Nodes in Industrial Internet," in Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (imcom) 2019, Cham, 2019, vol. 935, pp. 254–266.

[7] Y. Yin, Z. Wang, W. Zhou, Y. Gan, and Y. Zhang, "Group key agreement protocol for edge computing in industrial internet," Math. Biosci. Eng., vol. 19, no. 12, pp. 12730–12743, 2022.

[8] S. Zhu, K. Ota, and M. Dong, "Green AI for IIoT: Energy Efficient Intelligent Edge Computing for Industrial Internet of Things," Ieee Trans. Green Commun. Netw., vol. 6, no. 1, pp. 79–88, Mar. 2022.

[9] Q. Zhang, J. Cui, H. Zhong, and L. Liu, "Toward Data Transmission Security Based on Proxy Broadcast Re-encryption in Edge Collaboration," Acm Trans. Sens. Netw., vol. 18, no. 3, p. 48, Aug. 2022.

[10] H. Guo, Z. Zhang, J. Xu, N. An, and X. Lan, "Accountable Proxy Re-Encryption for Secure Data Sharing," Ieee Trans. Dependable Secure Comput., vol. 18, no. 1, pp. 145–159, Jan. 2021.

[11] J. Li, J. Peng, and Z. Qiao, "A Ring Learning with Errors-Based Ciphertext-Policy Attribute-Based Proxy Re-Encryption Scheme for Secure Big Data Sharing in Cloud Environment," Big Data.

[12] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A Verifiable and Fair Attribute-based Proxy Re-encryption Scheme for Data Sharing in Clouds," IEEE Trans. Dependable Secure Comput., pp. 1–1, 2021.

[13] Q. Huang, Y. Yang, and J. Fu, "Secure Data Group Sharing and Dissemination with Attribute and Time Conditions in Public Cloud," Ieee Trans. Serv. Comput., vol. 14, no. 4, pp. 1013–1025, Aug. 2021.

[14] A. F. Loe, L. Medley, C. O'Connell, and E. A. Quaglia, "TIDE: A Novel Approach to Constructing Timed-Release Encryption," in Information Security and Privacy, Acisp 2022, Cham, 2022, vol. 13494, pp. 244–264.

[15] K. Yuan, Y. Wang, Y. Zeng, W. Ouyang, Z. Li, and C. Jia, "Provably Secure Security-Enhanced Timed-Release Encryption in the Random Oracle Model," Secur. Commun. Netw., vol. 2021, p. 5593363, May 2021.

[16] X. A. Wang, A. K. Sangaiah, N. Nedjah, C. Shan, and Z. Wang, "On the Security of a CCA-Secure Timed-Release Conditional Proxy Broadcast Re-encryption Scheme," in Advances on P2p, Parallel, Grid, Cloud and Internet Computing, 3pgcic-2018, Cham, 2019, vol. 24, pp. 192–198.

[17] K. Emura, A. Miyaji, and K. Omote, "A Timed-Release Proxy Re-Encryption Scheme," Ieice Trans. Fundam. Electron. Commun. Comput. Sci., vol. E94A, no. 8, pp. 1682–1695, Aug. 2011.

[18] W.-B. Kim, S.-H. Kim, D. Seo, and I.-Y. Lee, "Certificateless Group to Many Broadcast Proxy Reencryptions for Data Sharing towards Multiple Parties in IoTs," Wirel. Commun. Mob. Comput., vol. 2022, p. 1903197, Jun. 2022.

[19] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional Identity-Based Broadcast Proxy Re-Encryption and Its Application to Cloud Email," Ieee Trans. Comput., vol. 65, no. 1, pp. 66–79, Jan. 2016.

[20] K. Liang, Q. Huang, R. Schlegel, D. S. Wong, and C. Tang, "A Conditional Proxy Broadcast Re-Encryption Scheme Supporting Timed-Release," in Information Security Practice and Experience, Berlin, Heidelberg, 2013, pp. 132–146.

[21] T. Zheng, Y. Luo, T. Zhou, and Z. Cai, "Towards differential access control and privacy-preserving for secure media data sharing in the cloud," Computers & Security, vol. 113, p. 102553, Feb. 2022.

Automatic Detection of Software Defects based on Machine Learning

Nawal Elshamy[1], Amal AbouElenen[2], Samir Elmougy[3]
Computer Science Department-Faculty of Computer and Information Science, Mansoura University, Mansoura, Egypt[1]
Computer Science Department-Faculty of Computer and Information Science, Mansoura University, Mansoura, Egypt[2, 3]

***Abstract*—Defects in software are one of the critical problems in software engineering community because they provide inaccurate results and negatively affect the quality and reliability of the software. These defects must be detected in the early stages of software development. Researchers had used Software Defect Detection (SDD) techniques to allow predicting module fault-proneness. By implementing the hyperparameter optimization techniques and exploiting data imbalances in predicting defects, this paper proposes and develops an SDD model with high performance and generalization capability. To classify defects in software modules, machine learning algorithms and ensemble learning techniques are used on the balanced datasets. The balanced datasets are obtained through using a hybrid of synthetic minority oversample (SMOTE) and Support Vector Machine (SVM). To obtain the optimal hyperparameters needed for the used classifiers and for the dataset balanced algorithms, Non-dominated Sorting Genetic Algorithm II (NDSGA-II) is used. To reduce the time and save other used resources, Hyperband technique, which is a multi-fidelity optimization, is used in NDSGA-II. A 10-fold Cross Validation (CV) is applied to overcome the overfitting and underfitting problems. The accuracy, recall, F-measure, and ROC AUC metrics are used to evaluate the SDD model. The results show that the proposed model predicts defects more accurately than the compared studies.**

Keywords—Software defect detection; NDSGA-II; hyperband; imbalance dataset

I. Introduction

One of the critical topics in the software engineering community is the development of high software quality and reliability while making effective use of limited resources. The Software Development Lifecycle (SDLC) is a structured method developed to ensure the production of stable, high-quality software. To ensure a timely and effective software system, it is essential to follow the SDLC's stages, which include requirement collecting, requirement analysis, system design, system development, and maintenance. A software fault may be a human error or a system-related error, failure, or crash. Defects have a significant effect on software quality and even the economics of software. So, fixing defects is an important part of software maintenance, but it also wastes time and resources. Detecting software faults before software deployment is crucial, as the correct detection of faulty software modules or components allows good use of resources and time [1] [2].

"Defect detection technology" is the ability to find bugs in every code change that developers send. SDD used ML approaches to software defect datasets characterized by software metrics (as features) to identify software module or component problems. Researchers had developed and implemented ML approaches for SDD.

One of the most important steps in creating a reliable ML model is tuning the model's hyperparameters. It's important to note that the tuning process differs for categorical, discrete, and continuous hyper-parameters. Manual testing is a common method for altering hyper-parameters, but it requires a comprehensive understanding of ML algorithms and their hyper-parameter settings. Due to the high number of hyper-parameters, the complexity of the models, the length of time required to evaluate the models, and the non-linear interactions between the hyper-parameters, manual tuning is often ineffective [3]. These considerations have motivated more studies in Hyper-parameter Optimization (HPO) approaches especially when working with huge datasets or when using complex ML algorithms with a large number of hyper-parameters. The main goal of HPO is to automate this process to improve the performance of the ML model, find the best ML model for a specific problem, and reduce the amount of human effort needed. To find ideal hyper-parameters, it is critical to use the optimal optimization technique. Because many HPO issues are often non-convex or non-differentiable optimization issues, traditional optimization techniques may be inadequate for them, resulting in a local rather than a global optimum [4]. A Non-Dominated Sorted Genetic Algorithm (NDSGA-II) is used in this paper for hyperparameter search hybrid with hyperband speed configuration.

SDD models were developed using Machine Learning (ML) classifiers. Nevertheless, SDD datasets contain more nondetectable than detectable occurrences; this is known as the "class imbalance problem." In addition, multiple studies on defect detection models revealed that minority classes contain more instances of faults than do majority classes that are defect-free. Hence, applying ML algorithms to such unbalanced data yields biased outcomes for minority-class occurrences. To successfully manage an imbalance in datasets, oversampling techniques are utilized [5].

This paper implements the SMOTE-SVM algorithm to balance the imbalanced data with tuned parameters based on the NDSGA-II hybrid with the Hyperband algorithm. Decision Tree (DT), Random Forest (RF), and ensemble learning with Adaboost (AB) and Bagging (BG) classifiers are the ML classifiers used in this work. The proposed model is evaluated on nine defect detection NASA datasets. This work is structured as follows: The second and third sections provide

"Background" and "literature review," respectively. The methodology and requirements for the experiment are discussed in Section IV. Experiments and their results are presented in Section V. The final section of the paper discusses the conclusion and what comes next.

II. Background

Several facets of SDD are discussed in this section. We describe optimization strategies for hyperparameter tuning and data-imbalance-resolving algorithms.

A. *Class Imbalanced Problem*

Problems with class imbalance arise in datasets where the values for different classes are distributed unevenly. Imagine a dataset in which 95% of the class values are from one class and only 5% are from another; this dataset is unbalanced. Minority classes and values are rarely reported, compared to hundreds, thousands, or even millions of majority examples. When ML classifiers are applied to such datasets, models lose detection capacity and provide off-target results. Biased outcomes are produced when a predictive model is used for classification on an uneven dataset. ML procedures function best with an evenly distributed dataset. In this way, ML models fail to accurately predict class values in unbalanced datasets. For minority class norms, this phenomenon is especially prevalent. Effectively addressing class imbalance issues is critical because minority class values are regarded as more significant than majority class values [6].

By simulating or synthesizing instances from underrepresented groups, defect detection experts have found that sampling strategies can produce more representative datasets. This research reconstructed instances of under-represented groups by using oversampling techniques. As a result, the oversampling techniques significantly improved the ML classifiers' detection performance.

1) SMOTE-SVM: SMOTE is a popular oversampling technique for achieving more equitable class distribution by simulating the emergence of new instances of the minority class along roads connecting existing instances of the minority class to their nearest neighbors. To aid in the establishment of class boundaries, SVM-SMOTE generates new instances of the minority class along boundary lines. The synthetic sampling technique with data generation (Synthetic Minority Oversampling Technique, SMOTE) [7] is one of the efficient special algorithms for re-establishing class parity after oversampling by increasing the number of objects in the minority class. Using k-nearest Neighbor (KNN) approach, SMOTE algorithm generates minority-class synthetic items from similarities in the feature space between existent objects. With this method, we can manufacture an arbitrary number of artificial objects that are "similar" to those in the minority class but are otherwise unique [8].

2) Support Vector Machine (SVM): It is a supervised learning algorithm that works by mapping low-dimensional data points into a high-dimensional feature space to make them linearly separable and then using an optimal separating hyperplane as the classification boundary to partition the data by increasing the difference between the two classes. With the assumption of n data points, SVM's objective function is [4][9][10]:

$$arg\,min_m \left\{\frac{1}{n}\sum_{i=1}^{n} \max\{0,1-\ y_i f(x_i)\} + C\boldsymbol{w}^T\,\boldsymbol{w}\right\} \quad (1)$$

where w is a normalization vector, and C is the penalty parameter of the error term, which is an important hyper-parameter of all SVM models.

Several kernels are available to be used in SVM models to determine the similarity between two data points, x_i and x_j. Hence, the type of kernel would be a crucial hyperparameter to adjust. The most common kernels used in SVM include linear kernels, Radial Basis Function (RBF) kernels, polynomial kernels, and sigmoid kernels. The different kernel functions can be denoted as follows [11]:

- Linear kernel

$$f(x) = \ x_i^T x_j \quad (2)$$

- Polynomial kernel

$$f(x) = \ (\gamma\, x_i^T x_j\ + r)^d \quad (3)$$

- RBF kernel

$$f(x) = \exp\left(-\gamma\, \left||x - x'|\right|^2\right) \quad (4)$$

- Sigmoid kernel:

$$f(x) = \tanh\left(-\gamma\, x_i^T x_j\ + r\right) \quad (5)$$

After deciding on a kernel type, further hyper-parameters must be adjusted, as demonstrated by the corresponding kernel function equations. When the "kernel type" hyper-parameter is set to polynomial, RBF, or sigmoid, the coefficient is the conditional hyper-parameter; r is the conditional hyper-parameter of polynomial and sigmoid kernels. Extra conditioned hyper-parameter, d, defines the "degree" of the polynomial kernel function.

As part of the training process for an SVM classifier, the kernel function type *k* (x_i, x_j) values and the value of the regularization parameter C are calculated so that a trade-off may be made between increasing the distance between classes and reducing the total error [12].

B. *Hyperparameter Optimization (HPO)*

The training of a ML model is subject to number of hyperparameters. These hyperparameters determine the model's strategy for learning a given connection between input and detections. Using hyperparameter optimization (HPO), a model, that has been fine-tuned using the most effective hyperparameters, could be obtained. This model ought to be able to produce minimum-loss solutions. The challenge with optimization problems is that the search space is typically infinite, while the resources available to carry out the search are finite (maximum x amount of time or iterations). Consequently, for an algorithm to search for the global minimum effectively, it must incorporate strategies to make the most of the available funds. HPO's slow execution time is a serious drawback, especially when dealing with a wide variety of hyper-parameter configurations or massive datasets [4].

1) Non-dominated sorting GA-II: As non-dominated sorting is used to generate a preliminary coarse ranking of the population, NSGA-II (Pareto dominance-based) continues to be one of the most well-liked algorithms. The non-dominated solutions can be found using this technique and then moved to the next class before being discarded. The solutions in each obtained class are sorted from best to worst according to the objective's crowding distance, the total of the differences between an individual's left and right neighbors. It's best to have a lot of space between individuals. Non-dominated sorting becomes less discriminative and the left and right neighbors in each goal are generally different solutions when there are more than two objectives, hence this strategy fails for problems with more than two objectives [13].

2) Multi-fidelity optimization technique: In order to work around the problem of having insufficient time or resources, multi-fidelity optimization techniques are frequently used. The original dataset or the features used can be reduced to a subset to save time [14]. Low-fidelity and high-fidelity evaluations are combined in multi-fidelity for use in the real world [15]. Low-fidelity evaluations are those that only test a small fraction of the data and so are very inexpensive, but have poor generalization performance. Better generalization performance is achieved at the expense of increased cost in high-fidelity assessments, where a larger subset is assessed. Poor performing configurations are eliminated from multi-fidelity optimization methods at each iteration of hyper-parameter evaluation on newly generated subsets, leaving only the best performing configurations to be evaluated on the full training set. Bandit-based algorithms are a kind of multi-fidelity optimization algorithms, and they make use of strategies like sequential halving [16] and Hyperband approach [17].

- Successive Halving: By testing each possible combination of hyper-parameters, successive halving can determine which one works best. However, in real-world applications, numerous considerations must be taken into account, such as time and resource constraints. The term "budget" is used to describe these considerations (B). The following is the primary procedure for employing successive halving algorithms for HPO. The first assumption is that there are n sets of hyper-parameter combinations to test and that these sets are tested using equally distributed resources ($b = {}^{B}/_{n}$). Finally, at the end of every cycle, half of the underperforming hyper-parameter configurations are thrown out, while the other half are passed on to the next cycle with double budgets ($b_{i+1} = 2 * b_i$). The preceding steps are continued until the best possible set of hyperparameters is found. The trade-off between the feasible hyper-parameter configurations and available budgets affects the cost-benefit analysis of succeeding halves [18]. With this in mind, the main issue with consecutive halving is deciding how to divide the budget, specifically between testing fewer configurations with a larger budget for each and testing more configurations with a smaller budget for each.

- Hyperband approach [17] is considered a solution to the issue of successive halving methods by dynamically picking an appropriate number of configurations. It seeks to strike a balance between the total budget (B) and the number of hyper-parameter configurations (n) by apportioning a portion of the total budget to each configuration ($b = {}^{B}/_{n}$). Each batch of random configurations is routinely halved in order to get rid of the inefficient hyperparameter setups and boost performance. The total quantity of data points, the minimal number of instances required to train a meaningful model, and the available budgets all contribute to the restrictions b_{min} and b_{max} . After that, we use b_{min} and b_{max} to get the total number of configurations n and the budget size for each (max). Based on n and b, a random sample of configurations is generated and fed into the shown successive halving model. Each iteration of the consecutive halving method takes the top half of the configurations and discards the bottom half, keeping only the best-performing ones. This is done again until the best possible set of hyperparameters is found.

C. Ensemble Learning (EL)

1) Random Forest (RF): RF is a bagging approach that generates many independent, tiny decision trees from the dataset at random. Selecting a small number of attributes at each node in order to find the best branching technique allows for a deeper tree structure. Dataset and feature randomization makes overfitting less likely. In a classification problem, the majority vote is used to determine the final class label [19].

2) Bagging: The goal of "bagging" [20] is to increase the reliability of ML algorithms, especially decision trees. It helps prevent data from being overfit and lowers the model's variance. It takes the original training dataset and randomly selects n subsets of features and data samples, with replacement, to create n new datasets. Parallel models for making detections, using each of the n sub datasets as a single input, are built. The bagging classifier selects as its output the label class predicted by the majority of the base models.

*3) Adaptive Boosting (AdaBoost**):*** It is a technique used to increase the accuracy and performance of numerous weak classifiers by combining them into a single strong classifier. Adaboost's weak classifiers frequently use decision stumps, a variant of the decision tree that consists of a single node (the root) and two branches. Various classes of faulty classifiers are taught. When a less accurate classifier misclassifies a sample, the sample is assigned more weight in the next classifier. Further, the aggregate weight is based on how well each poor-quality classifier performed. Adaboost classifier's ultimate output is just a weighted sum of the results obtained by the individual weak classifiers [21].

D. Classification Algorithm

1) Decision Tree (DT): DT is a predictive modeling algorithm that can partition features in a dataset in multiple ways based on different conditions, resulting in a tree-like structure. The tree is made up of a decision node, which seeks the best feature split, and leaves (termina l nodes), which are used to create a final detection. Different criteria, such as gini and entropy, are used to divide the features. One significant consideration while constructing terminal nodes for the decision tree is deciding when to stop developing trees and when to create more terminal nodes. This can be done using two criteria: maximum tree depth (the number of nodes in the tree after the root node) and minimum node records (the number of training patterns represented by a given node). Once a node is built, the same method can be used on all created data by dividing the dataset into subsets to produce child nodes. The detection process entails selecting the appropriate node in a decision tree and then proceeding to walk down it with the relevant row of data.

III. Literature Review

Several SDD studies had recently been conducted to accurately detect early-stage developmental defects. Researchers had attempted to improve the model’s performance using various methods, algorithms, and measurements. Some of the most current papers relevant to this work are discussed here.

Benala and Tantati [22] examined the effects of five oversampling methods to solve the imbalanced data, including random oversampling, SMOTE, adaptive synthetic sampling (ADASYN), Safe-Level-SMOTE (SL-SMOTE), and SVM-SOMTE, to for SDD using nine imbalanced NASA datasets. They used decision tree (j48-classifier), the RF, Naive Bayes (NB), and EL classifiers. They concluded that SVM-SMOTE is the best oversampling technique due to its ability to produce minority-class instances in a region bordered by support vectors. Kassaymeh et al. [23] suggested a combination of Salp Swarm Algorithm (SSA) and Backpropagation Neural Network (BPNN) to solve the SDD problem. They used BPNN to find the best BPNN parameters. Different performance measures are used to evaluate the results, in which the results of their experiments showed that the combined work is often the best way to solve SDD problems. Goyal [24] proposed a filtering method, called FILTER to accurately predict defects. They used SVM-based classifiers (linear, polynomial, and radial basis functions) and a suggested filtering technique. They declared that their work improves the accuracy, AUC, and F-measure by 16.73%, 16.80%, and 7.65%, respectively, based on five datasets. Azzeh et al. [25] studied the effect and consistency of four kernel functions with feature selection on the performance of SVM for SDD. Four kernel functions, 10 feature subset selection thresholds based on the information gain technique, thirty-eight publicly available datasets, and a single evaluation measure were used in this comprehensive study. Since then, 1520 experiments had been conducted. Since the performance of other kernel functions is constrained, the results showed that SVM with an RBF kernel is the best option for defective datasets. Sharma et al. [26] analyzed the application of the ensemble method of ML technique in the field of SDD. They focused on the global state during the period 2018–2021, which had been examined from a multidimensional perspective, including the selection of a specific ML algorithms, and the research gap that may lead to the future scope of the work that can be accomplished. Ye et al. [27] developed a multi-objective immunity optimization method based on a thorough fitness evaluation mechanism, which allows it to efficiently tackle the used model. Two objectives are optimized: defect detection rate and false alarm rate for defects. Their proposed algorithm is based on comprehensive fitness evaluation, which has a better selection ability to attain the predicted effect of population evolution software and further assists decision makers in selecting a better scheme that meets their needs. In addition, to validate the efficacy of their proposed algorithm, they compared it against eight distinct public data sets, in which the results showed that their suggested work handles the multi-objective undersampling SDD problem more effectively. Shafiq et al. [28] developed an approach for SDD using ML to enhance software quality. They used PC1 data set as input data. Ant Colony Optimization (ACO) is used to determine which characteristics are the most crucial. The chosen characteristics are fed into SVM. The author declared that their results showed that ACO-based SVM performs better than SVM, NB, and KNN classifiers in solving SDD.

IV. Proposed Approach

The proposed approach in this work classifies the defects that exist in the software system and discovers the defect modules during the software development process, so the model that suffers from defects have high priority during quality assurance checks. The framework of the proposed approach in this work is shown in Fig. 1, which is logically divided into the following four phases. The preprocessing, which is Phase I, includes the standardization features and label encoding target features. Phase-II includes oversampling the dataset to balance the class distribution by applying SMOTE-SVM and hyperparameter tuning optimization using NDSGA-II and the Hyperband approach. Phase-III is focusing on training the SDD models using the balanced dataset and making the detections. Phase IV is concerned with evaluating the performance of SDD models and conducting comparative analyses.

This work aims to develop a high-quality approach to detect bugs in the early stages of the software development life cycle, by balancing the modules affected by defects in the dataset using the high-performance balanced algorithm. The SMOTE oversampling approach uses KNN similarity measures between items to generate synthetic instances in the defect class. This leads to the creation of an unknown number of artificial items that are "similar" to those in the current defect class but do not duplicate them. This algorithm has some disadvantages, such as sample overlap, noise interference, and blindness of neighbor selection. To address these problems, SMOTE is hybridized with SVM, so SVM-SMOTE generates new instances of defect classes near borderlines with SVM to help establish boundaries between classes.

However, the behavior of these algorithms is controlled by a set of parameters that remain static during the training process. The quality of the detection can be improved by fine-tuning these parameters to achieve the best possible results. Here, we apply an optimization algorithm called NDSGA-II to search for and fine-tune the hyperparameters of these algorithms and then pick the best possible parameters. Since NDSGA-II is resource-intensive and time-consuming algorithm, it has been integrated with Hyperband approach, to speed up the configuration evaluation by getting rid of ineffective parameters that don't have a global minimum. Testing the model's intermediate scores for a given set of hyperparameters is how it functions. For instance, after a fixed number of rounds, one could examine all the intermediate scores and eliminate the least effective parameters.

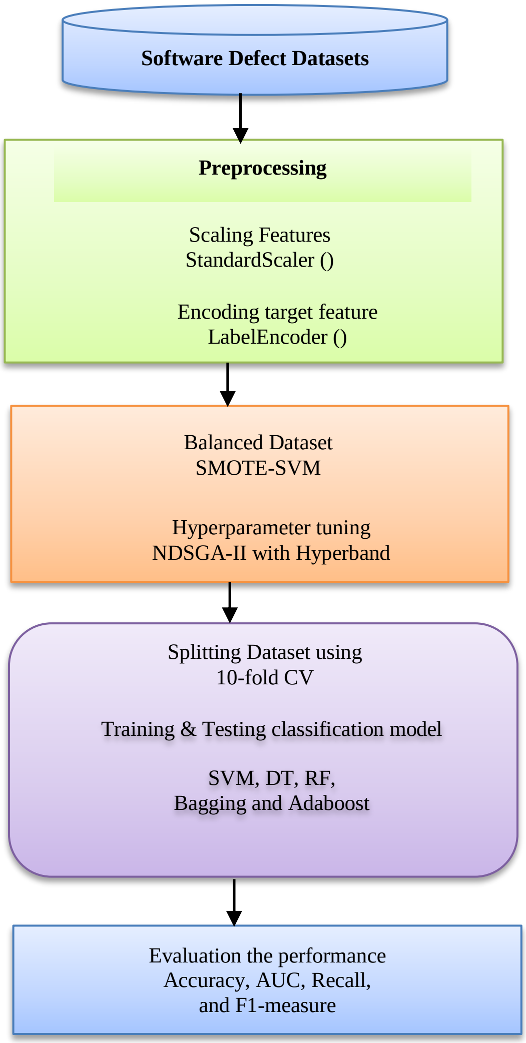


Fig. 1. The proposed approach.

In Fig. 2, the steps of the search optimization technique used to obtain the optimal parameters for the SMOTE-SVM algorithm are illustrated. This process is described in stages as follows:

- Stage 1: Initialize the hyperparameters search space that represents the population (P), which consists of a combination of input variables (V).
- Stage 2: Generate random parent population.
- Stage 3: Apply the SMOTE-SVM-based model described in Algorithm 1 for each variable.
- Stage 4: Each V's accuracy is calculated using 10-fold CV to identify the fitness vectors, and each vector was given a fitness rank proportional to its non-domination frequency.
- Stage 5: Assign ranks to all V in a P by first selecting all of the non-dominated solutions from P and placing them in rank 1, then selecting all of the remaining solutions and placing them in rank 2, and so on.
- Stage 6: Sorts each V according to a dominance rule that said:

A variable ($\ddot{X}$) is said to dominate another variable ($\dot{X}$), if

- ✓ There is no objective of $\ddot{X}$ worse than that objective of $\dot{X}$.
- ✓ There is at least one objective of $\ddot{X}$ better than that objective of $\dot{X}$.

- Stage 7: Offspring resulting from recombination between two unrelated parents enter the progeny P. Throughout the process of mutation, the child's values shift. That process is continued until the P is twice as large as it was at the outset.
- Stage 8: Nondenominational criteria are used to re-classify P. In this way, a new generation will be selected according to established hierarchy.
- Stage 9: In the next iteration, crowding-sort will be used to determine the density of solutions if the partially included case holds. Less dense trials are selected for the next generation until the population count is back to its starting point.
- Stage 10: Iteratively producing and checking poorly configured parameters, then discarding their offspring, is repeated until the maximum number of generations is reached, at which point the optimal hyperparameter is returned.

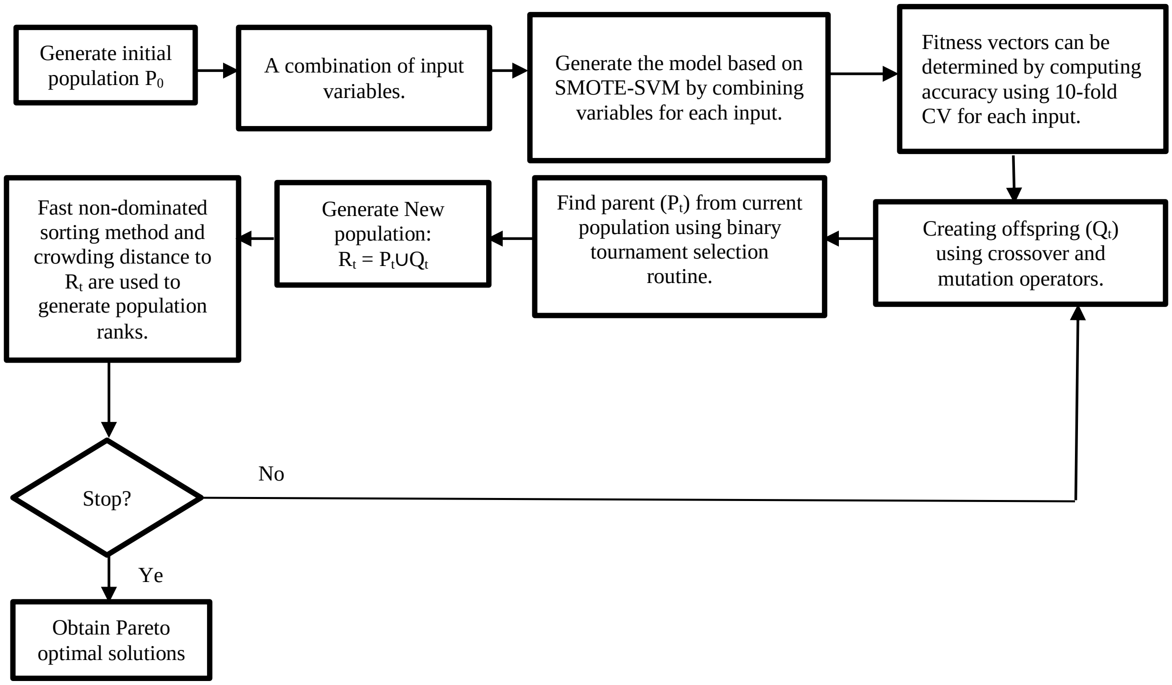

Generate initial population P_0
A combination of input variables.
Generate the model based on SMOTE-SVM by combining variables for each input.
Fitness vectors can be determined by computing accuracy using 10-fold CV for each input.
Creating offspring (Q_t) using crossover and mutation operators.
Find parent (P_t) from current population using binary tournament selection routine.
Generate New population: $R_t = P_t \cup Q_t$
Fast non-dominated sorting method and crowding distance to R_t are used to generate population ranks.
Stop?
No
Ye
Obtain Pareto optimal solutions

Fig. 2. Main steps of obtaining the optimal parameters of SMOTE-SVM algorithm.

```
Algorithm 1: The proposed balancing datasets (SMOTE-SVM-based model)

Input:
    Software defects dataset (Đ):
        where:
            D = Defects ∪ non_defects;
            Defects represents instances in the dataset with defect class
            Non_defects represent instances in the dataset from non_defects class.
Output:  The balanced dataset

for D' ∈ Đ do
begin
    X = instances
    y = target class
    // Encoding target features that contain categorical data
    y_Encode = LabelEncoder (y);
    // Appling standardization technique to be in the same range
    X_Scale = StandardScaler (X);
for each data point <$x_{i,}$ , $y_i$>
begin
    // <xi,, yi>: the defect instances which denotes the minority class in Đ
    Defect support vectors = SVM algorithm (defect subsets, i.e., minority subsets);
    m = kNN (defect support vector);
    If   number of majority neighbors < $\frac{m}{2}$
        $X_{new} = X_i + (\dot{X}i , X_i) * \Re$, where $\Re \in [0,1], Xi \in Smax$,
                        $\dot{X}i \in$ KNN of $X_i$ , and Smax, is a majority instance
    End if
    Else
        $X_{new} = X_i + (\dot{X}i_{-}, X_i) * \Re$, where $\Re \in [0,1], X_i \in S_{min,}$
                        $\dot{X}i \in$ one of KNN of $X_{i,}$ which $S_{min,}$ is minority instance
    $\check{D} = X_{new} \cup Đ$
End for
    return $\check{D}$
End for
```

The following paragraphs discuss the SDD model that is presented in Algorithm 1, where the used defective datasets are imbalanced and the output is the balanced dataset. The necessary preprocessing steps are executed for each dataset. Encoding the categorical data in the chosen dataset is represented by yEncode, and xScale represents the features after standardization. The SVM model is applied in the minority subset, and neighbors of the defect support vector are obtained. If the number of non-defects class neighbors is less than half of the nearest defect support vector, a new object of the majority class is generated; otherwise, the new object will be of the minority class.

V. Experimentations and Results

The methodology behind the experiments is discussed here, as well as the infrastructure used and the datasets that were examined.

A. Environment

The proposed approach was tested on a laptop running Microsoft Windows 10 Pro 64-bit and an Intel(R) Core (TM) i7-8565U Processor at 1.80 GHz (92 MHz). Oversampling methods are taken from the imblearn library, and ML classifiers are imported into Sklearn 1.0.2 using an x64 processor running Jupyter notebook 6.1.4 with Python 3.8.5 and OPTUNA (HPO framework).

B. Description to Dataset and Software Metrics used

This work used nine NASA datasets and was investigated by Shepperd et al. [29]. Halstead metrics, McCabe metrics, size metrics, and other properties are included in these datasets to help establish the quality of a software model. If a dataset has bad class values, it probably is bad. The model is fine if the class value is "0" or "no," but it's broken if it's "1" or "yes." Minority class samples range from being extremely under-represented to being evenly distributed across the datasets. The nine NASA datasets are illustrated in Table I.

TABLE I. NASA DATASET DESCRIPTION

Database	# Of Features	# Of Instances	# Of Defective	# Of Non-Defective	Defective %
KC1	22	1183	314	869	15.45
KC2	22	522	104	415	20.5
KC3	40	194	36	158	18.55
CM1	22	344	49	295	12.21
PC1	22	705	77	644	8.03
PC2	37	1585	16	1569	1
MC1	39	1988	46	1942	2.31
MW1	38	253	27	226	10.67
JM1	22	7782	326	1783	21.48

C. Parameter Setting for SDD Models

Table II details the various classifier parameter settings.

TABLE II. PARAMETER SETTINGS OF THE USED ALGORITHMS

Parameters	Values
NDSGA-II	
iterations	100
Population size	Hyperparameter space
Crossover rate	1
Mutation rate	$\frac{1}{generated\ groups}$
SVM	
Kernel type	Rbf
gamma	0.61719
C	16.1657
degree	1
RandomState	104
KNN	
K	8
m	10
DT, RF	
criterion	entropy

D. Evaluation Criteria

The proposed classifiers' efficacy is measured using standard metrics including the confusion matrix, ROC, AUC, accuracy mean, and F-measure.

- Accuracy: It is a ratio of the number of correct detections to the total number of observations, as illustrated in Eq. (6).

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad (6)$$

- Precision is the ratio of correct positive detections out of the total number of positive detections as given in Eq. (7)

$$precision = \frac{\text{TP}}{\text{TP+FP}} \quad (7)$$

- Recall is the ratio of correct positive observations to all positive observations made in class, as given in Eq. (8).

$$recall = \frac{\text{TP}}{\text{TP+FN}} \quad (8)$$

- F1 score is a weighted average of precision and recall. F1 is usually preferable to accuracy, particularly if the class distribution is uneven, as calculated from Eq. (9).

$$F1\ score = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall+Precision}} \quad (9)$$

- AUC-ROC It shows TPR versus FPR at different thresholds to differentiate "signal" from "noise." It separates classes and summarizes the ROC curve. High AUC means the model separates positive and negative groups well.

E. Results

1) Results of applying the decision tree algorithm: the results of DT for defect detection using the proposed approach are shown in Table III. The best results obtained on the PC2, MC1, and CM1 datasets, respectively, are as follows: 1) the accuracy achieved 98.18%, 95.59%, and 94.28%. 2) The AUC achieved 9.9798, 0.9546, and 0.9466. 3) F-measure achieved 0.9736, 0.936, and 0.9259.

2) Results of applying the Random Forest algorithm: The results of RF for defect detection using the proposed approach are shown in Table IV. The best results obtained on the KC3, MC1, and PC2 datasets, respectively, are as follows: 1) the accuracy achieved 100 %, 99.66 %, and 99.06 %. 2) AUC achieved 1.0, 0.9974, and 0.9868. 3) F-measure achieved 1.0, 0.9950, and 0.9866.

3) Results of applying the Adaboost algorithm: The results of Adaboost for defect detection using the proposed approach are shown in Table V. The best results obtained on the KC3, PC2, and MC1 datasets, respectively, are as follows: 1) The achieved accuracy is 100%, 99.09%, and 97.28%. 2) The achieved AUC 1.0, 0.9868, and 0.9746. 3) The achieved F-measure achieved 1.0, 0.9866, and 0.9611.

TABLE III. RESULTS OF APPLYING DT ALGORITHM

DATASETS	ACCURACY (%)	AUC	RECALL	F1-MEASURE
KC1	88.20	0.8820	0.8707	0.8806
KC2	89.15	0.8922	0.8333	0.8860
KC3	93.548	0.9354	0.93333	0.9333
CM1	**94.285**	**0.9466**	**0.9615**	**0.9259**
PC1	90.77	0.9077	0.9126	0.9082
PC2	**98.18**	**0.9798**	**0.9736**	**0.9736**
MC1	**95.59**	**0.9546**	**0.9504**	**0.9365**
MW1	86.66	0.8656	0.9130	0.8749
JM1	83.47	0.8347	0.85535	0.8381

TABLE IV. Results of Applying RF Algorithm

Datasets	Accuracy (%)	AUC	Recall	F1-Measure
KC1	92.415	0.9241	0.9325	0.9247
KC2	95.180	0.9520	0.92857	0.95121
KC3	**100**	**1.0**	**1.0**	**1.0**
CM1	92.857	0.9431	1.0	0.9122
PC1	95.14	0.9514	0.9417	0.9509
PC2	**99.09**	**0.9868**	**0.9736**	**0.9866**
MC1	**99.66**	**0.9974**	**1.0**	**0.9950**
MW1	93.33	0.9328	0.9565	0.9361
JM1	89.80	0.8980	0.89977	0.8982

TABLE V. Results of Applying Adaboost Algorithm

Datasets	Accuracy (%)	AUC	Recall	F1-Measure
KC1	82.022	0.82022	0.7640	0.8095
KC2	91.566	0.9155	0.92857	0.91764
KC3	**100**	**1.0**	**1.0**	**1.0**
CM1	88.571	0.90122	0.96153	0.8620
PC1	89.80	0.8980	0.9029	0.8985
PC2	**99.09**	**0.9868**	**0.9736**	**0.9866**
MC1	**97.28**	**0.9746**	**0.9801**	**0.9611**
MW1	88.88	0.8873	0.9565	0.8979
JM1	75.27	0.7527	0.71070	0.74197

4) Results of applying the bagging algorithm: The results of bagging for defect detection using the proposed approach are shown in Table VI. The best results obtained on the MC1, PC2, and KC3 datasets, respectively, are as follows: 1) the accuracy achieved 98.64%, 98.18%, and 96.77%. 2) The achieved AUC is 0.9825, 0.9736, and 0.9666. 3) The achieved F-measure 0.98, 0.9729, and 0.9655.

TABLE VI. Results of Applying Bagging Algorithm

Datasets	Accuracy (%)	AUC	Recall	F1-Measure
KC1	91.85	0.9185	0.8876	0.9159
KC2	91.566	0.9160	0.8809	0.9135
KC3	**96.77**	**0.9666**	**0.93333**	**0.9655**
CM1	91.428	0.9239	0.96153	0.89285
PC1	94.17	0.9417	0.9320	0.9411
PC2	**98.18**	**0.9736**	**0.9473**	**0.9729**
MC1	**98.64**	**0.9825**	**0.9702**	**0.98**
MW1	95.555	0.9555	0.9565	0.9565
JM1	88.54	0.8854	0.8747	0.8842

5) Results of applying the svm algorithm: The results of SVM for defect detection using the proposed approach are shown in Table VII. The best results obtained on the PC2, KC2, and MC1 datasets, respectively, are as follows: 1) the accuracy achieved 99.09 %, 98.79 %, and 98.64 %. 2) The AUC achieved 0.9868, 0.9878, and 0.973. 3) F-measure achieved 0.9866, 0.9882, and 0.9803.

TABLE VII. Results of Applying SVM Algorithm

Datasets	Accuracy (%)	AUC	Recall	F1-Measure
KC1	87.92	0.8792	0.8707	0.8781
KC2	**98.79**	**0.9878**	**1.0**	**0.9882**
KC3	93.54	0.9354	0.9333	0.9333
CM1	91.42	0.9160	0.92307	0.8888
PC1	97.57	0.9757	0.9805	0.9758
PC2	**99.09**	**0.9868**	**0.9736**	**0.9866**
MC1	**98.64**	**0.9873**	**0.9900**	**0.9803**
MW1	95.55	0.9565	0.9130	0.9545
JM1	82.16	0.8216	0.8018	0.8181

VI. Result Analysis and Discussion

Different experiments were conducted to find out how the oversampling and HPO techniques affected the performance of SDD models. We first investigated how well SDD models with NDSGA-II hybrid using Hyperband approach for HPO and SMOTE-SVM balanced datasets performed. Tables III to VII display the results for various performance measures for the available NASA software defect dataset. Therefore, it can be asserted that we have successfully implemented all the classification algorithms with high performance. From these tables, it is clear that RF is the most accurate method to classify defects, while Adaboost is the least accurate of all evaluation measures. With KC3 dataset, Adaboost and RF classifier were able to achieve 100% accuracy. Table VIII presents the accuracy results of the proposed work with and without parameter tuning on different datasets, to illustrate the influence of tuning these parameters on the model's performance. Also, Fig. 3 and Fig. 4 show the comparison between these methods. These results show that tuning the algorithm's parameters helped make the models more accurate than the other methods which don't tune the parameters.

TABLE VIII. Accuracy Measure for All Classifiers with and without Hyperparameter Tuning

Datasets	Accuracy Without PARAMETER TUNING (%)		Accuracy With PARAMETER TUNING (%)	
KC1	SVM	77.52	SVM	87.92
	DT	85.95	DT	88.20
	RF	91.57	RF	92.415
	Adaboost	80.33	Adaboost	82.022
	Bagging	91.01	Bagging	91.85

KC2	SVM	85.54	SVM	98.79
	DT	89.15	DT	89.15
	RF	96.38	RF	95.180
	Adaboost	91.566	Adaboost	91.566
	Bagging	90.36	Bagging	91.566
KC3	SVM	90.32	SVM	93.54
	DT	90.32	DT	93.548
	RF	100	RF	100
	Adaboost	90.32	Adaboost	100
	Bagging	96.77	Bagging	96.77
CM1	SVM	79.72	SVM	91.42
	DT	85.13	DT	94.285
	RF	89.18	RF	92.857
	Adaboost	81.08	Adaboost	88.571
	Bagging	93.24	Bagging	91.428
PC1	SVM	79.12	SVM	97.57
	DT	91.74	DT	90.77
	RF	95.14	RF	95.14
	Adaboost	91.26	Adaboost	89.80
	Bagging	94.17	Bagging	94.17
PC2	SVM	93.27	SVM	99.09
	DT	97.47	DT	98.18
	RF	98.31	RF	99.09
	Adaboost	96.63	Adaboost	99.09
	Bagging	96.63	Bagging	98.18
MC1	SVM	89.9	SVM	98.64
	DT	96.84	DT	95.59
	RF	99.05	RF	99.66
	Adaboost	96.21	Adaboost	97.28
	Bagging	98.73	Bagging	98.64
MW1	SVM	77.77	SVM	95.55
	DT	88.88	DT	86.66
	RF	86.66	RF	93.33
	Adaboost	80	Adaboost	88.88
	Bagging	86.66	Bagging	95.555
JM1	SVM	69.23	SVM	82.16
	DT	82.1	DT	83.47
	RF	89.51	RF	89.80
	Adaboost	75.78	Adaboost	75.27
	Bagging	87.92	Bagging	88.54

These results also demonstrated that the proposed method outperformed the method proposed in [22] that also used SMOTE-SVM for balanced defect datasets by an average of 10.59% with DT, 8.0246% with RF, 2.25% with Adaboost, and 14.8276% with bagging. This is due to the ability of the proposed algorithm with using the optimal parameters, shown in Table II, obtained from applying NDSGA-II algorithm. Hyperband approach [17] is used in NDSGA-II to reduce the time and save other used resources.

Table IX compares the method described in this paper to several other methods for detecting software defects.

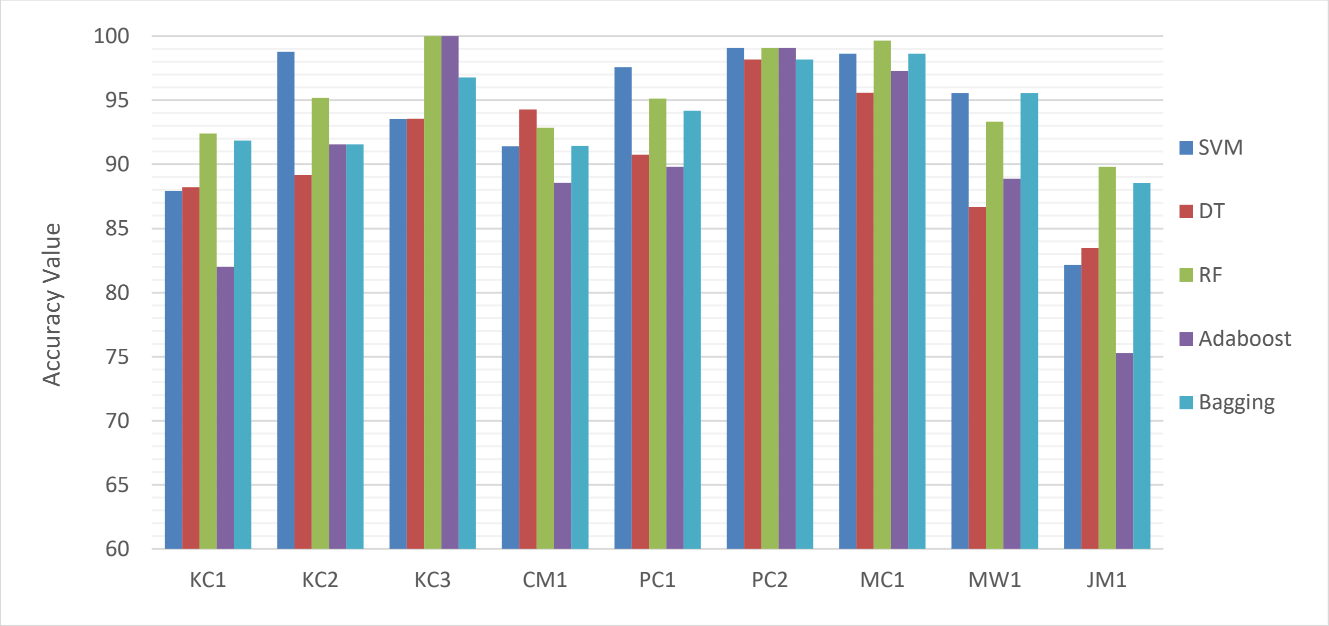


Fig. 3. Comparison accuracy values of five classifiers based on the proposed approach.

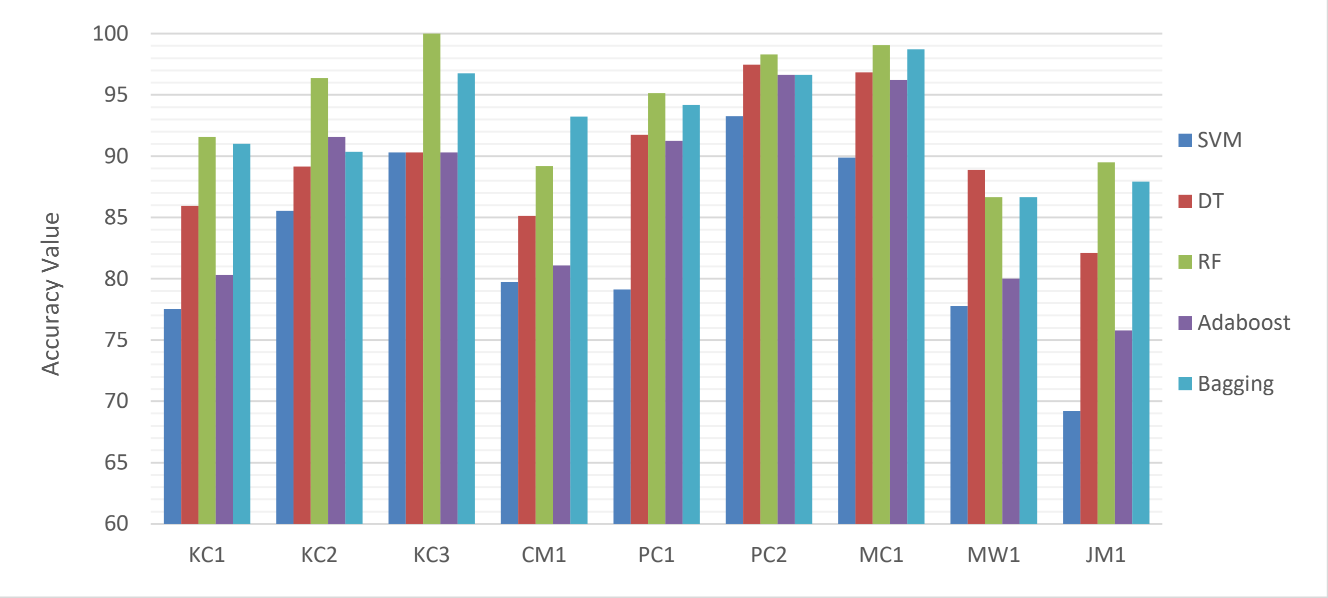


Fig. 4. Comparison accuracy values of five classifiers based on the proposed approach without tuning algorithm parameters.

TABLE IX. PERFORMANCE COMPARISON OF MULTIPLE METHODS FOR SDD ON NASA DATASET

Ref.	METHOD	CLASSIFIER	DATASET	ACCURACY	AUC	F-MEASURE
[22]	SMOTE-SVM	DT	KC2	80		
			KC3	87		
			CM1	84		
			PC1	73		
			PC2	79		
			MC1	85		
			MW1	88		
			JM1	68		
		RF	JM1	80		
			KC2	82		
			KC3	79		
			CM1	89		
			PC1	95		
			PC2	88		
			MC1	97		
			MW1	88		
		Adaboost	JM1	80		
			KC2	82		
			KC3	79		
			CM1	86		
			PC1	95		
			PC2	97		
			MC1	97		
			MW1	88		
		Bagging	KC1	68		
			KC2	78		
			KC3	64		
			CM1	87		
			PC1	91		

			PC2	79		
			MC1	85		
			MW1	82		
[23]	SSA-BPNN		KC1	88.92	0.79	
			KC2	88.54	0.85	
			KC3	93.48	0.92	
			CM1	88	0.85	
			PC1	90.69	0.79	
			PC2	99.64	0.93	
			JM1	82.03	0.70	
			MW1	94.21	0.93	
	Proposed model	DT	KC1	88.20	0.8820	0.8806
			KC2	89.15	0.8922	0.8860
			KC3	93.548	0.9354	0.9333
			CM1	94.285	0.9466	0.9259
			PC1	90.77	0.9077	0.9082
			PC2	98.18	0.9798	0.9736
			MC1	95.59	0.9546	0.9365
			MW1	86.66	0.8656	0.8749
			JM1	83.47	0.8347	0.8381
		RF	KC1	92.415	0.9241	0.9247
			KC2	95.180	0.9520	0.95121
			KC3	100	1.0	1.0
			CM1	92.857	0.9431	0.9122
			PC1	95.14	0.9514	0.9509
			PC2	99.09	0.9868	0.9866
			MC1	99.66	0.9974	0.9950
			MW1	93.33	0.9328	0.9361
			JM1	89.80	0.8980	0.8982
		Adaboost	KC1	82.022	0.9185	0.9159
			KC2	91.566	0.9160	0.9135
			KC3	100	0.9666	0.9655
			CM1	88.571	0.9239	0.89285
			PC1	89.80	0.9417	0.9411
			PC2	99.09	0.9736	0.9729
			MC1	97.28	0.9825	0.98
			MW1	88.88	0.95553	0.9565
			JM1	75.27	0.8854	0.8842
		Bagging	KC1	91.85	0.9185	0.9159
			KC2	91.566	0.9160	0.9135
			KC3	96.77	0.9666	0.9655
			CM1	91.428	0.9239	0.89285
			PC1	94.17	0.9417	0.9411
			PC2	98.18	0.9736	0.9729
			MC1	98.64	0.9825	0.98
			MW1	95.555	0.95553	0.9565
			JM1	88.54	0.8854	0.8842

VII. Conclusion and Future Work

In this work, a new software defect detection (SDD) approach is proposed and developed as an efficient and smart way to find software defects. This approach uses SMOTE-SVM algorithm, to address the issue of imbalanced behavior in NASA datasets. The proposed method used NDSGA-II algorithm with Hyperband approach for hyperparameter optimization of SMOTE-SVM algorithm, followed by using standard ML methods and ensemble techniques for training. The experimental results were assessed using NASA datasets, in which the results showed that our proposed work outperforms the conventional techniques and methods in predicting software faults based on accuracy, AUC, recall, and F-measures. Also, the results showed that RF performed the best with 95.2746% average accuracy, while Adaboost performed the lowest with 90.2754% average accuracy. As future work, we plan to investigate the impact of using deep learning on the improvement of SDD when imbalanced data is used. Also, we plan to use other techniques for HPO, and other assessment measures such as G-measure, balance, and Matthews' Correlation Coefficient (MCC).

References

[1] G¨orkem Giray, Kwabena Ebo Bennin,¨Omer K¨oksal,¨Onder Babur, and Bedir Tekinerdogan. On the use of deep learning in software defect prediction. Journal of Systems and Software, 195:111537, 2023.

[2] Wei Zheng, Tianren Shen, Xiang Chen, and Peiran Deng. Interpretability application of the just-in-time software defect prediction model.Journal of Systems and Software, 188:111245, 2022.

[3] Amal Alazba and Hamoud Aljamaan. Software defect prediction using stacking generalization of optimized tree-based ensembles. Applied Sciences, 12(9):4577, 2022.

[4] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing,415:295–316, 2020.

[5] Tirimula Rao Benala and Karunya Tantati. Efficiency of oversampling methods for enhancing software defect prediction by using imbalanced data. Innovations in Systems and Software Engineering, pages 1–17,2022.

[6] Ruchika Malhotra and Shine Kamal. An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. Neurocomputing, 343:120–140, 2019.

[7] Shujuan Wang, Yuntao Dai, Jihong Shen, and Jingxue Xuan. Research on expansion and classification of imbalanced data based on smote algorithm. Scientific reports, 11(1):1–11, 2021.

[8] Hien M Nguyen, Eric W Cooper, and Katsuari Kamei. Borderline over-sampling for imbalanced data classification. International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1):4–21, 2011.

[9] Jian Zhang, Rong Jin, Yiming Yang, and Alexander Hauptmann. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. 2003.

[10] Li Yang. Comprehensive visibility indicator algorithm for adaptable speed limit control in intelligent transportation systems. PhD thesis,University of Guelph, 2018.

[11] Omar S Soliman and Amira S Mahmoud. A classification system for remote sensing satellite images using support vector machine with non-linear kernel functions. In 2012 8th International Conference on Informatics and Systems (INFOS), pages BIO–181. IEEE, 2012.

[12] Ga¨el Varoquaux, Lars Buitinck, Gilles Louppe, Olivier Grisel, Fabian Pedregosa, and Andreas Mueller. Scikit-learn: Machine learning without learning the machinery. GetMobile: Mobile Computing and Communications, 19(1):29–33, 2015.

[13] Ankita Golchha and Shahana Gajala Qureshi. Non-dominated sortinggenetic algorithm-ii–a succinct survey. International Journal of Computer Science and Information Technologies, 6(1):252–255, 2015.

[14] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127, 2015.

[15] Si Zhang, Jie Xu, Edward Huang, and Chun-Hung Chen. A new optimal sampling rule for multi-fidelity optimization via ordinal transformation.In 2016 IEEE International Conference on Automation Science and Engineering (CASE), pages 670–674. IEEE, 2016.

[16] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In International Conference on Machine Learning, pages 1238–1246. PMLR, 2013.

[17] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh,and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research, 18(1):6765–6816, 2017.

[18] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. Automated machine learning: methods, systems, challenges. Springer Nature, 2019.

[19] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. Machine learning, 63:3–42, 2006.

[20] Leo Breiman. Bagging predictors. Machine learning, 24:123–140,1996.

[21] Yoav Freund. Boosting a weak learning algorithm by majority. Information and computation, 121(2):256–285, 1995.

[22] Tirimula Rao Benala and Karunya Tantati. Efficiency of oversampling methods for enhancing software defect prediction by using imbalanced data. Innovations in Systems and Software Engineering, pages 1–17, 2022.

[23] Sofian Kassaymeh, Salwani Abdullah, Mohammed Azmi Al-Betar, and Mohammed Alweshah. Salp swarm optimizer for modeling the software fault prediction problem. Journal of King Saud University-Computer and Information Sciences, 34(6):3365–3378, 2022.

[24] Somya Goyal. Effective software defect prediction using support vector machines (svms). International Journal of System Assurance Engineering and Management, 13(2):681–696, 2022.

[25] Mohammad Azzeh, Yousef Elsheikh, Ali Bou Nassif, and Lefteris Angelis. Examining the performance of kernel methods for software defect prediction based on support vector machine. Science of Computer Programming, 226:102916, 2023.

[26] Tarunim Sharma, Aman Jatain, Shalini Bhaskar, and Kavita Pabreja. Ensemble machine learning paradigms in software defect prediction. Procedia Computer Science, 218:199–209, 2023.

[27] Ye, T., Li, W., Zhang, J. and Cui, Z., 2023. A novel multi-objective immune optimization algorithm for under sampling software defect prediction problem. Concurrency and Computation: Practice and Experience, 35(4), p.e7525.

[28] Muhammad Shafiq, Fatemah H Alghamedy, Nasir Jamal, Tahir Kamal, Yousef Ibrahim Daradkeh, and Mohammad Shabaz. Scientific programming using optimized machine learning techniques for software fault prediction to improve software quality. IET Software, 2023.

[29] Martin Shepperd, Qinbao Song, Zhongbin Sun, and Carolyn Mair. Data quality: Some comments on the nasa software defect datasets. IEEE Transactions on Software Engineering, 39(9):1208–1215, 2013.

Balancing Technological Advances with User Needs: User-centered Principles for AI-Driven Smart City Healthcare Monitoring

Ali H. Hassan[1], Riza bin Sulaiman[2], Mansoor A. Abdulgabber[3], Hasan Kahtan[4]
Institute of IR 4.0 (IIR4.0), Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia[1, 2]
College of Computer and Cyber Sciences, University of Prince Mugrin, Madinah 41499, Saudi Arabia[1]
Department of Software Engineering, Üsküdar University, Istanbul, Turkey[3]
Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff CF5 2YB[4]

***Abstract*—In recent years, the integration of artificial intelligence (AI) technologies has greatly benefited smart city healthcare, meeting the growing demand for affordable, efficient, and real-time healthcare services. Patient monitoring is one area where artificial intelligence has shown great promise. Improved health outcomes have been made possible by the advancement of AI-based monitoring systems, which enable more personalized and continuous patient monitoring. However, to fully maximize the benefits of these systems, a user-centered approach is essential, which prioritizes patients' needs and experiences while ensuring their privacy and autonomy are respected. This study focuses on the application of user-centered design principles in the development and deployment of AI-driven monitoring systems in smart city healthcare. Addressing the challenges and opportunities of AI-driven monitoring systems, the article considers issues such as privacy and security concerns, data accuracy, and user acceptance. Finally, some possible future directions to the challenges are suggested. A user-centered approach to AI monitoring systems is recommended for healthcare providers to enhance patient experience in smart city healthcare.**

Keywords—Smart healthcare; patient monitoring; smart city; artificial intelligence; user-centered

I. Introduction

The advancement of technology in the modern era has had a significant impact on the delivery of healthcare services. With the increasing demand for affordable, efficient, and real-time healthcare, smart cities have become a hub for the integration of artificial intelligence (AI) technologies. Smart cities are urban areas that use technology to improve their residents' quality of life. A healthcare system consists of certain groups (such as patients, primary care physicians, pharmacists, and other specialists) and a variety of stages (such as medical issue screening, sickness determination, clinical therapy, and recovery) [1]. Together, all these parties contribute to the implementation of the smart healthcare ecosystem. Global healthcare practices are changing because of a shift in digital technologies. The integration of intelligent technology and healthcare systems enables easy access to healthcare data and connects resources for effective management of unpredictable health care demand [2]. Unlike traditional healthcare, smart healthcare utilizes IoT, 5G wireless technologies, cloud computing, big data and artificial intelligence [3]. These technologies are the most effective communication framework since they allow more accurate collection, recording, and analysis of patient data and direct knowledge sharing between healthcare service providers [4].

Healthcare is an important aspect of smart city development, and the incorporation of AI technologies has the potential to revolutionize healthcare service delivery in these areas. One area where AI has shown significant potential is in patient monitoring. The development of AI-driven healthcare monitoring systems has enabled continuous and personalized patient monitoring, leading to improved health outcomes. Healthcare monitoring is important in maintaining healthcare services for patients such as the elderly or those suffering from chronic diseases. It can reduce the need for medical intervention [5], will rely less on traditional physical health facilities like hospitals and long-term care centers, and will be more individualized as a result. Numerous studies have been conducted on smart healthcare to monitor patient health, for example, Parkinson's disease [6] and mental health [7] and cardiovascular disease [8]. However, to maximize the benefits of AI-driven monitoring systems in smart city healthcare, it is crucial to adopt a user-centered approach. This approach prioritizes patients' needs and experiences while ensuring their data is collected and analyzed in a manner that respects their privacy and autonomy. This is particularly important in smart cities where populations are diverse and have unique healthcare needs. Industry 5.0 and 5G telecommunication technology have made it possible to create low-cost sensing technologies for real-time monitoring and data acquisition [9]. These systems could help increase patient experience and promote health outcomes by emphasizing the needs and experiences of patients.

As the demand for healthcare services in smart cities continues to grow, the integration of user-centered AI monitoring systems has the potential to play a critical role in meeting that demand. By prioritizing the needs and experiences of patients, these systems have the potential to enhance patient experience and improve health outcomes. Therefore, this study aims to investigate the role of user-centered AI monitoring systems in enhancing patient experience in smart city healthcare. The study seeks to understand how we best can overcome the challenges associated with employing AI-driven monitoring systems in smart city healthcare, with a focus on incorporating user-centered principles into the design and implementation of these systems. The remainder of this study is organized as following: Section I summarize the background and motivation for the study. Section II gives a description of the key elements and qualities of a smart city. Section III reviews the adoption of AI in healthcare. Section IV explores AI-driven monitoring system, including remote, mobile and wearable health monitoring systems. Section V shows research studies that have adopted a user-centred approach to develop AI-based monitor systems. In Section VI, the challenges, opportunities, and potential future directions are presented. Finally, Section VII concludes the paper, summarizes the main points and suggests guidelines for areas of future work.

II. Method

This systematic review aims to synthesize the existing literature on the role of user-centered AI monitoring systems in smart city healthcare. The search strategy for this review included electronic databases such as Web of Science, PubMed, Science Direct, Google Scholar, IEEE Xplore and ACM Digital Library as well as manual searches of reference lists of relevant articles. The search terms included “AI-based monitor systems”, “smart city”, “smart health”, “user-centred design in healthcare”, “challenges and opportunities of AI monitor systems”. The inclusion criteria for this review were studies that involved the development of AI-based monitor systems and studies which adopted a user-centred approach to the development of AI monitor systems. Review studies were included to understand the past and current state of the research topic. Exclusion criteria included studies that did not use a AI-based innovations. Two reviewers independently screened titles and abstracts to assess their eligibility, and full text articles were evaluated to be included. A narrative synthesis approach was used to summarize the key findings of the included studies, identifying patterns and themes across studies. The data was analyzed to better understand the challenges, opportunities, and impact of user-centered AI monitoring systems in smart city healthcare. The Prisma model shown in Fig. 1 illustrates the complete process of article search and selection.

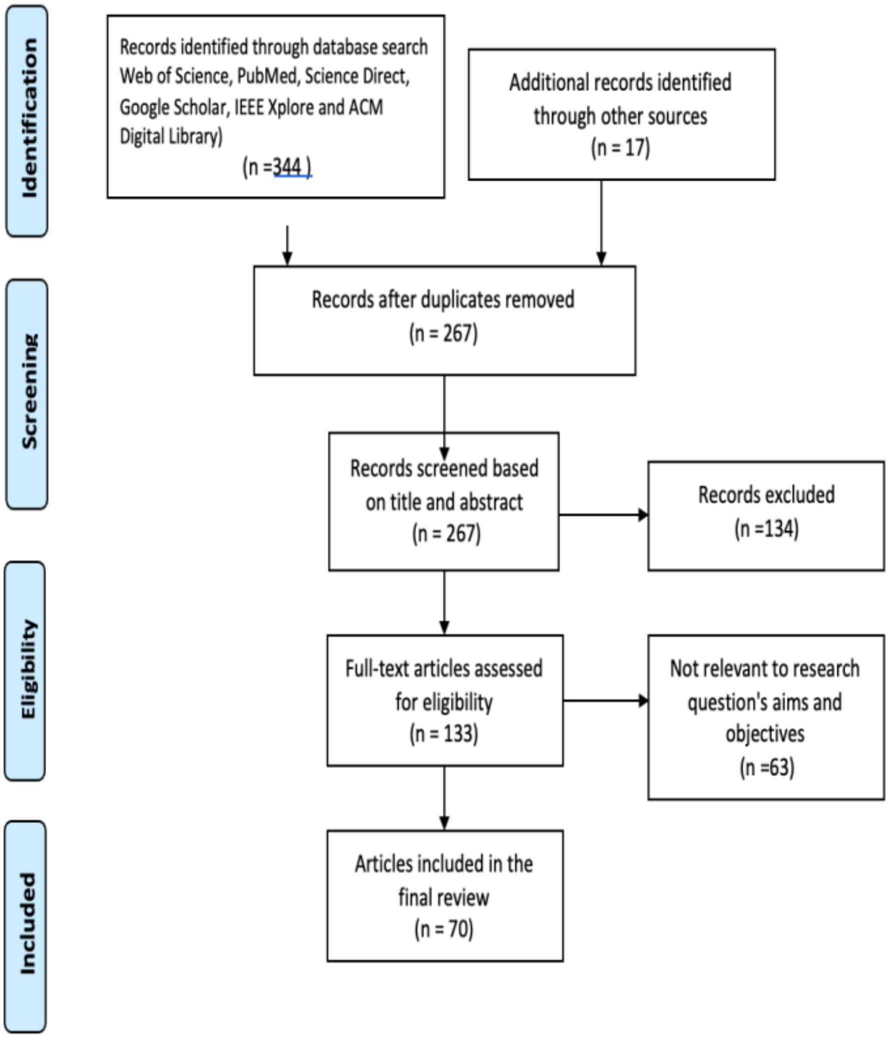

Identification
Screening
Eligibility
Included
Records identified through database search Web of Science, PubMed, Science Direct, Google Scholar, IEEE Xplore and ACM Digital Library) (n =344)
Additional records identified through other sources (n = 17)
Records after duplicates removed (n = 267)
Records screened based on title and abstract (n = 267)
Records excluded (n =134)
Full-text articles assessed for eligibility (n = 133)
Not relevant to research question's aims and objectives (n =63)
Articles included in the final review (n = 70)

Fig. 1. Prisma model shows number of record included and excluded.

III. Literature Review

A. *Smart Cities*

Cities consist of various interacting entities, such as citizens, businesses, modes of transportation, communication networks, services, and utilities. The development of cities poses numerous challenges, such as waste management, lack of resources, air pollution, and health concerns [10]. Smart cities are thought to provide opportunities to solve these problems [11]. The main goal of a smart city includes connecting residents, facilitating communication, improving government performance and economic growth, promoting sustainability and improving citizens' quality of life using smart technologies [12], [13].

1) Smart cities infrastructure: A smart city's infrastructure is increasingly dependent on advanced technologies to address population growth-related issues. AI and IoT-enabled devices will improve the quality of life and maximize the efficiency of a wide range of daily services in smart cities, such as smart transport, smart energy, and smart water management [14]. Fig. 2 illustrates the infrastructure of smart cities.

a) Smart healthcare: Smart cities aim to provide smart healthcare solutions. Sensors and smart devices have significantly contributed to the availability of high-quality patient care using smart health technologies.

b) Smart education: In smart education, the focus is on and learner-driven, personalized, and adaptable learning services, interactive, collaborative, and collaboration resources and services. In addition, a smart learning environment facilitates the realization by learners of an effective, efficient, and meaningful learning experience [15].

c) Smart transportation: Smart transportation is a technological advancement in the conventional transportation system that improves public life by delivering optimized services through mobile devices, city-installed devices, or sensors built into cars. These sensors and devices enable easy parking reservations, efficient street lighting, optimal route suggestions, public transit telematics, accident prevention, and autonomous driving [16].

d) Smart buildings: In smart cities, energy efficiency is emphasized, and buildings are designed to function according to individual fulfilment, so that they are more sustainable and energy efficient; for example, the use of smart devices in smart buildings allows many tasks to be handled, including controlling climate, lighting, and many other aspects based on detection of presence [17].

e) Smart energy systems: A smart energy system is built on technological advancements with the goal of decreasing energy consumption and boosting the use of renewable resources. The primary goal of smart energy is to conserve non-renewable resources for use in emergencies [11].

f) Smart manufacturing: Smart RFID tags make it easy to trace a product from the factory to the retailer, greatly reducing time and costs. Smart packaging can also include benefits like brand protection, quality assurance, and customer customization [16].

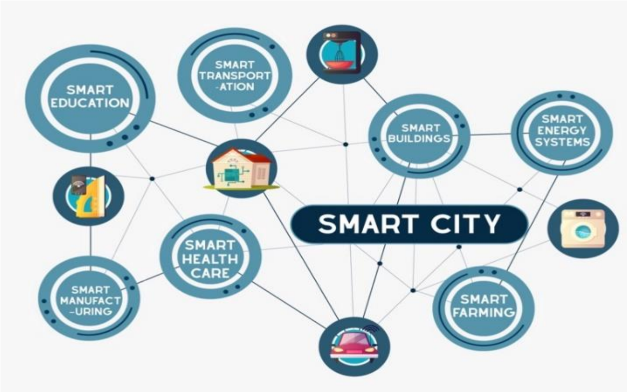


Fig. 2. Components of a smart city.

g) Smart farming: Effective agriculture production is crucial for our people to prevent a future potential food crisis. The use of IoT technology in agriculture may undoubtedly contribute to securing enough food demand and improving the overall efficiency of agricultural production processes. Significant crop information could be gathered and used for yield monitoring and the early detection of diseases that could have a major effect on crop yield and quality. Soil and nutrient monitoring would optimize agricultural production methods and result in water savings, which are valuable in some geographic areas [14].

2) Overview of smart cities characteristics: Smart cities can help to enhance city residents' standard of living while also making the city more sustainable and efficient. With the help of technology and data, smart cities can improve city infrastructure and services and solve some of the most pressing problems facing modern cities. Smart cities have several key characteristics that make them innovative and efficient as illustrated in Fig. 3.

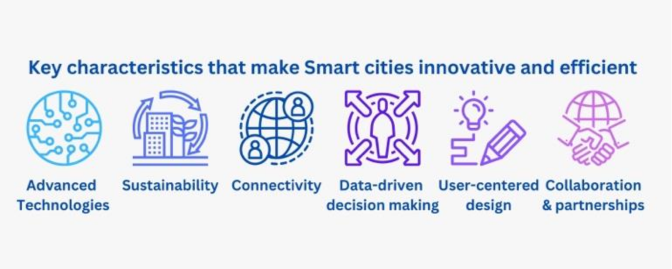

Key characteristics that make Smart cities innovative and efficient
Advanced Technologies
Sustainability
Connectivity
Data-driven decision making
User-centered design
Collaboration & partnerships

Fig. 3. Key characteristics that make smart cities innovative and efficient.

a) Advanced technologies: Smart cities infrastructure are based on innovative technologies to combat urban challenges with the help of IoT devices, artificial intelligence (AI), cloud computing, and Big Data analytics, [14], [18], [19].

b) Sustainability: Smart cities are designed to minimize their environmental impact including reducing energy consumption, encouraging the use of green spaces, and improving air and water quality [10], [20].

c) Connectivity: Smart cities are highly interconnected, with advanced communication networks connecting residents, businesses, and government agencies. This enables real-time data exchange and seamless communication [13].

d) Data-driven decision making: In smart cities, data is the driving force behind decisions and services. The data collected through advanced technologies are used to identify trends and improve services to better the quality of live for its residents [21], [22].

e) User-centered design: Smart cities are designed with citizens' needs in mind, with the goal of improving the wellbeing and welfare of their citizens [23]–[25]. Accessible and affordable transportation, green spaces, and sustainable housing are part of this approach.

f) Collaboration and partnerships: For smart cities to succeed, governments, businesses, and residents must collaborate. Collaboration, transparency and sharing of data and resources are essential to solving urban problems and creating a more efficient and sustainable city [20], [25].

Smart cities are the future of urban development, designed to address some of the most pressing issues that cities face. IoT devices, cloud computing, big data and artificial intelligence are examples of advanced technologies that enable real-time data exchange and data-driven decision making. Smart cities are also designed to reduce environmental impact, promote sustainability, improve connectivity, foster collaboration, and focus on the needs of citizens. A user-centered approach will help to ensure that the needs of citizens are considered when designing the city. Collaboration and partnerships are critical to smart city success because governments, businesses, and residents must collaborate to solve urban problems and create a more efficient and sustainable city. Transparency and resource sharing are critical to achieving this goal. By embracing these principles, cities can improve the quality of life for their citizens and ensure a sustainable future for future generations.

3) Smart city for a smart healthcare: Smart cities and smart health are two interrelated concepts that are changing the way we live and manage our health. A smart city is a framework that employ technology and data to handle the issues of waste management, traffic congestion, air pollution and energy use etc. to improve the quality of life for citizens [19], [20]. Smart health, on the other hand, refers to the use of cutting-edge technologies. The Internet of Things (IoT), artificial intelligence (AI), cloud computing, and an array of digital tools has enabled us to improve the way we manage and maintain our health [3], [19]. Combining the benefits of smart cities and smart health can help to create healthier and more sustainable environments for their citizens. Artificial intelligence and machine learning technologies are key tools utilized in smart healthcare, where it can be used to analyze large amounts of data and thus improve diagnostic and treatment precision. It can also be used to predict diseases and disease progression, allowing for earlier intervention and improved patient care [26]. Cloud computing is utilized for numerous disease diagnosis applications, allowing stakeholders to make informed medical decisions [27]. IoT devices can collect and monitor patient data in real time, improving care and reducing the need for in-person hospitalization [28], [29]. These devices can also aid in disease prevention by enabling the user to monitor their own health status, assess their situation, and receive advice and recommendations [3]. Robotic technology can be used to improve surgical precision and reduce the risk of surgical complications [30].

Smart healthcare can provide numerous benefits to both patients and healthcare professionals. Patients will benefit from reduced wait times for doctor's appointments as well as more personalized treatment and care. Furthermore, the technologies can help reduce healthcare costs while improving diagnostic and treatment precision and efficiency [31]. Although smart healthcare has many advantages, there are also challenges to overcome. Such as, privacy and security when using smart healthcare technologies. The collection and sharing of personal health information has the potential to violate privacy and security. As a result, it is critical to have adequate security measures in place to protect patient data [32], [33]. Moreover, Healthcare professionals may also require new training and skills as a result of the technologies, which can be challenging [34], [35]. Smart healthcare is an innovative approach to the healthcare sector in which technology is used to improve patient care and healthcare quality. AI, IoT, and robotics technologies can provide benefit to both patients and healthcare professionals. However, there are obstacles to overcome, such as, training requirements, and concerns about privacy and security. If appropriate safeguards are in place, smart healthcare technologies may continue to have the capability to enhance healthcare services and patient care.

B. Advancements in AI- Driven Healthcare Monitoring Systems

1) Artificial intelligence: Artificial intelligence (AI) is of growing importance as it has the potential to improve healthcare through innovative medical devices and provide more personalized healthcare. The various domains AI plays a role is illustrated in Fig. 4. The application of new mobile devices, the development of smart home technologies, and the growing acceptance of smart health have made it possible for healthcare providers to access various form of medical media, such as X-rays and voice recordings. It is now much easier to obtain and share medical data with other healthcare providers for future care [36]. Smart home technologies have assisted healthcare providers in providing high-quality, low-cost care to patients. Structured and unstructured healthcare data can be processed using artificial intelligence. Machine learning (ML), which includes neural networks, classical support vector and deep learning, are common AI techniques used in the healthcare sector [37].

Research shows that there are various healthcare domains where AI can play a critical role. For example, wearable technologies [38] and [39] allow patients to monitor their health and AI-based clinical decision support systems are continuously used to aid and improve diagnosis [31]. Chatbots have been employed to provide mental health services that could lead to early prevention and support [7]. Surgical robots and mixed reality technology can help perform more precise surgeries [30]. To increase the effectiveness of their operations and patient management, many hospitals are installing artificial intelligence technologies that allow the information to gather from electronic medical records (EMRs) [40]. The drug industry [41], [42] is another area that has benefitted from the application of AI.

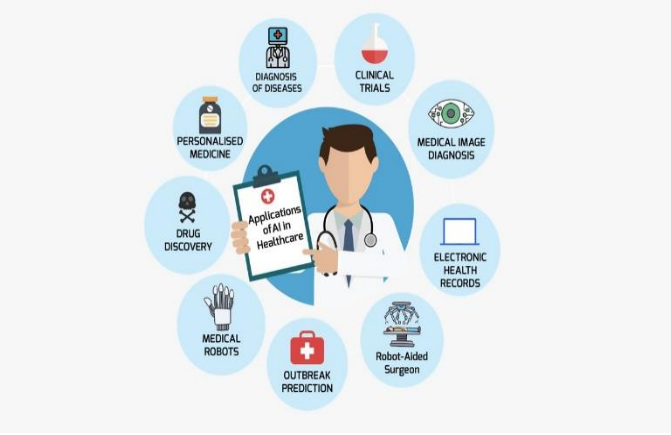


Fig. 4. Application of artificial intelligence in healthcare.

2) AI-driven monitoring system in smart healthcare: AI-driven monitoring systems have attracted researchers' attention due to their significant impact on people's lives. There have been numerous research studies [33], [43], frameworks [44], [45], [46] and services [8], [36] that focus on integrating artificial intelligence for smart healthcare. Table I summarizes the development of health monitoring systems considering the healthcare context, the invention, and AI technology.

- Remote Health Monitoring Systems

Researchers in [8] used IoT and AI to remotely monitor cardiovascular patients using a device that allows for smart monitoring of human vitals such as body temperature, heart rate, oxygen levels. In [36], the authors suggest a smart home healthcare monitoring system where installed sensors and devices can meet the needs of elderly people for continuous care at home. Researchers in [47] offer a system for remote monitoring of chronically ill or older people. The system is designed to automatically identify physiological signals [48] developed a framework for hybrid real-time remote monitoring (HRRM) for chronically ill patients. Researchers in [49] developed a home health monitoring system to track the health conditions of individuals with type 2 diabetes and hypertension. The system would diagnose and monitor patients' blood pressure and diabetes status. Researchers in [50] developed a remote monitoring system for Parkinson's disease patients using a voice signal as input.

- Mobile and Wearable Health Monitoring Systems

Researchers in [44] developed a deep learning-based voice pathology detection system for remote patient monitoring. The mobile application uses parallel CNNs to extract deep-learned features. In [51], the authors suggest a mobile and automatic system to improve patient cardiovascular health management abilities using IoT and AI technologies. A mobile application system is proposed by [45] for obesity management. The application utilizes a genetic algorithm as an AI engine to predict the meals a user will need to consume to meet their calorie, macronutrient, and micronutrient goals. The author in [46] presents a wearable health monitoring device to monitor patients' activities and vitals while they are engaged in activity to support efficient and effective health monitoring. The author in [52] offer a Smart Health Monitoring System framework built using IoT and machine learning to monitor patients affected by coronary artery disease. Research by [53] focuses on developing an AI-based patient monitoring system to observe the vital signs of Covid 19 patients remotely. Another research team in [54] have developed a smart healthcare framework for the detection and monitoring of COVID-19 using smart connectivity sensors and deep learning (DL). The application collects and analyzes patient Chest X-ray images. The system will then predict the patient's status and alert the healthcare professional to take action. In [55], the author offers a smart home healthcare framework for diabetic management using AI technologies. The proposed framework included portable sensors to detect temperature, blood sugar level, and activity of diabetic patients.

Remote diagnosis and treatment are crucial in rural areas and other regions with insufficient qualified medical professionals. Numerous healthcare applications' capacity to gather patient data remotely can help to overcome workforce constraints and process the data automatically for use in patient care as needed [5]. Further, traveling for medical care is not only time- and money-consuming, but it can also be challenging for some people, such as the elderly [56]. Automating smart healthcare monitoring systems limits the risk of human error, which can put both patients and providers at risk [57]. The information obtained is the main difference between face-to-face and remote diagnosis.

AI-based monitoring systems are showing promising results and can help to transform the healthcare industry by providing a more efficient and effective way of monitoring patients' health, reducing the risk of human error, and providing more affordable treatment options. Remote health monitoring systems employ IoT and AI-based technology to remotely monitor cardiovascular, elderly, chronically ill, and Parkinson's disease patients. Mobile and wearable health monitoring systems collect real-time data on patients through use of mobile applications, wearable devices, and IoT technologies. The use of wearable devices and mobile phones has made it easier to collect data for remote diagnosis and treatment. IoT and sensor technology are also used to handle the data collected by these devices. The use of AI in healthcare monitoring has the potential to provide affordable treatment options and improve the overall health of vulnerable citizens. The future of healthcare monitoring is expected to see an increase in the adoption of wearable devices and mobile applications, making the use of AI in healthcare monitoring more prevalent in daily life.

Modern technology, including mobile and wearable devices, has made it easier to find affordable treatment options by collecting real-time data without affecting regular activities. Wearable sensors such as watches, straps and glasses as well as mobile phones are useful methods to collect data [7]. Usually, these devices are linked to a network and establish remote communication with mobile devices. Fig. 5 shows the growing trend of wearable technology and the increasing focus on collecting and analyzing personal health data to improve overall wellness. The use of cloud-based infrastructure and edge computing devices allows for real-time data processing and analysis, which can lead to more accurate and actionable insights. Wearable devices including glasses, watches, and chest straps, can be connected to edge computing devices such as smartphones or middleware servers through Wi-Fi and Bluetooth technology. These devices are used to collect and transmit data related to the wearer's health and fitness, and the data is processed and stored on cloud-based infrastructure for further analysis. Glasses can have a display that can provide the wearer with real-time information such as heart rate and other health metrics. During exercise, the chest strap may monitor the wearer's heart rate and other vital signs, while a watch may track physical activity, calories burned, and sleep patterns. With enough data obtained, it is possible and practical to provide remote diagnosis and remote therapy. IoT and sensor technology are combined to handle the data that has been obtained [56]. The use of wearable devices will grow and become more prevalent in daily life, making them potentially useful to monitor the health of vulnerable citizens [58].

TABLE I. SUMMARY OF HEALTH MONITORING SYSTEMS

Ref.	Healthcare context	Invention	AI technology
[8]	Cardiovascular patients	A device based on IoT and AI that monitor the human vitals, including oxygen levels, heart rate, and body temperature.	K-Nearest Neighbors (KNNs)
[36]	Health conditions of elderly people	A monitoring system that includes video cameras and microphones fitted in the smart home.	Interlaced derivative pattern
[47]	Chronically ill or elderly	A physiological signal monitoring Smart-Monitor system.	Deep Neural Network
[48]	Chronically ill patients	A framework for hybrid real-time remote monitoring (HRRM) for chronically ill patients.	Naïve Bayes, Support vector
[49]	Type 2 diabetes and hypertension	A Smart home health monitoring system that analyzes blood pressure and glucose levels.	Support vector K-NN, Decision tree
[50]	Parkinson's Disease	A healthcare monitoring system for PD patients using voice signal as input	Support vector Gaussian mixture model, Random forest tree
[44]	Voice disorders	A smart mobile healthcare framework that enables voice pathology detection. The system used parallel CNNs to extract deep-learned features.	Parallel convolutional neural network, AlexNet
[51]	Cardiovascular patients.	An AI-enhanced mobile system for cardiovascular health management	Convolutional Neural Network, Recurrent Neural Network
[45]	Management of Obesity	A mobile application for self-management of obesity.	Genetic Algorithm technique
[46]	General health	An IoT and AI-based wearable monitor that tracks the patient's activities and vitals	Neural network
[52]	Coronary disease	A Smart Health Monitoring System. The application includes a pulse sensor that can be placed on a fingertip to monitor early signs of heart disease.	Decision Tree Algorithm and Random, Backwoods Classifier Algorithm, Support vector machine
[53]	Covid 19	AI-Based Patient Monitoring System to observe vital signs of patients. A webpage application is used to communicate between patients and health professionals.	Deep neural network
[54]	Covid 19	A smart healthcare framework for detection and monitoring of COVID-19.	ResNet50, convolutional neural network
[55]	Diabetic patients	A framework for automated monitoring of diabetic patients using sensors and smartphones.	Naïve Bayes, Random Forest, ZeroR, Simple logistic, sequential minimal optimization

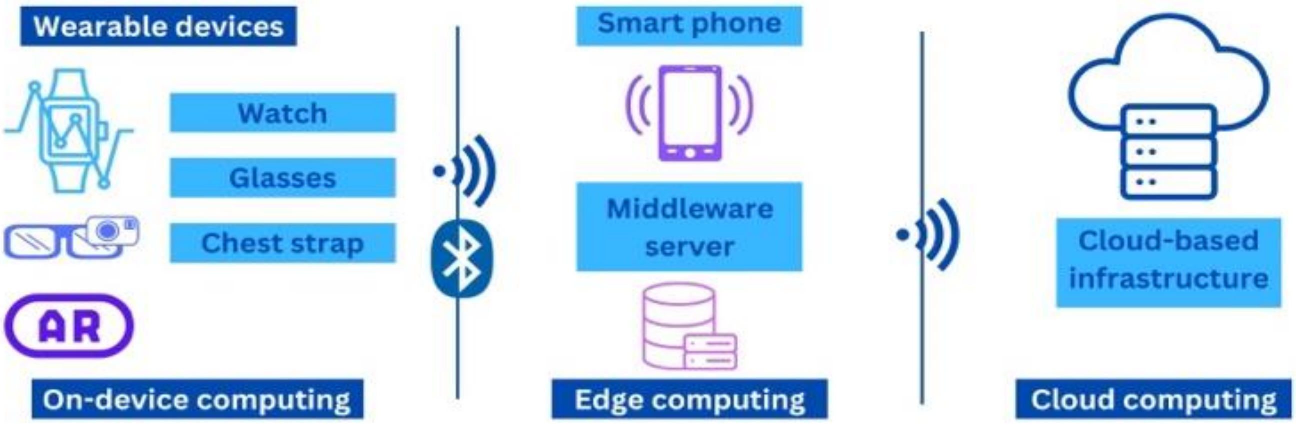


Fig. 5. Wearable device application process.

3) User-Centered Principles in AI-Driven Monitoring Systems: User-centered AI monitoring systems are health monitoring systems that are designed with the end user in mind. These systems employ artificial intelligence (AI) technology to collect, process, and analyze health-related data to improve patients' overall health and well-being. The goal of these systems is to provide personalized health monitoring and support to users without interfering with their privacy [33]. According to research, user-centered AI monitoring systems have the potential to transform the health industry by providing real-time health monitoring and support to patients [26], [59]. Using artificial intelligence technology, health-related data such as vital signs, physical activity levels, and sleep patterns can be automatically collected, analyzed, and used to detect health-related issues [8], [49].

Several studies have highlighted the importance of involving end users in the development of AI monitoring systems. Study [60] presents a comprehensive design methodology for developing mobile health (mHealth) apps for chronic pain management. The methodology focuses on incorporating user-centered design principles to improve the user experience and app adoption. The author discusses the pain management landscape, current challenges with mHealth app development, and highlights the importance of involving end-users in the design process. Also [61] provides a detailed overview of the importance of user-centered design in the development of health apps. The study emphasizes the increasing popularity of health apps and the need for designers and developers to ensure that these apps meet the needs and preferences of users. The authors discuss several key components of a user-centered design approach, such as user research, user feedback, and continuous evaluation, all of which are necessary for ensuring that health apps are effective, usable, and accessible. The authors also offer practical advice for those looking to create user-centered health apps. They emphasize the importance of taking user characteristics like age, gender, culture, and socioeconomic status into account during the design process. The authors emphasize the importance of health apps being accessible and usable for people with disabilities.

A study by [62] developed a voice monitoring system for disorder prevention using a user-centered design approach. The authors present a step-by-step process for developing a voice monitoring system that meets the needs and preferences of users, as well as a thorough overview of the importance of user-centered design in the development of healthcare technologies. The authors suggest a process for user-centered design that includes user research, prototyping, testing, and evaluation. The authors also provide a detailed discussion of the technical challenges involved in developing a voice monitoring system, such as speech recognition and data processing, as well as recommendations for overcoming these obstacles. A study led by [63] investigates dementia patients, their caregivers, and healthcare providers' physical, psychological, and social needs. The study employed a user-centered design method, which included semi structured interviews, workshops, and smart home trials. The findings of this study demonstrate how this methodology can help uncover psychosocial and contextual factors, as well as help to develop more patient-centered interventions. The authors suggest that their findings can be incorporated into clinical practice as well as public health strategy to develop patient-centered interventions. Another study by [32] outlines a user-centered strategy for creating smart home solutions that puts an emphasis on creating home automation systems that meet users' needs and expectations. The authors develop scenarios to better understand user needs and behaviors as well as to point out areas where smart home technology can benefit users in their day-to-day activities. The study found that user-centered design can produce smart home systems that are more efficient and user-friendly, and that it is essential to incorporate user feedback throughout the design process to produce solutions that satisfy their needs.

User-centered AI monitoring systems have the potential to transform the health industry by providing personalized health monitoring and support to patients without interfering with their privacy. Incorporating user-centered design principles in the development of these systems is crucial to ensure that they are effective, usable, and accessible. User-centered design methodologies have been successfully employed in various healthcare technologies, such as mobile health (mHealth) apps for chronic pain management, voice monitoring systems for disorder prevention, and smart home solutions for dementia patients. As shown in Fig. 6 these methodologies involve user research, prototyping, testing, and evaluation, which uncover psychosocial and contextual factors and help to develop more patient-centered interventions. Incorporating user feedback throughout the design process can produce solutions that satisfy user needs and preferences and are more efficient and user-friendly. Understanding the needs, preferences, and behavior of users is critical to the success of these technologies, and incorporating user feedback throughout the design process can produce solutions that satisfy user needs and preferences. As healthcare technologies continue to advance, incorporating user-centered design principles will be essential to develop patient-centered interventions that improve patient outcomes and experiences.

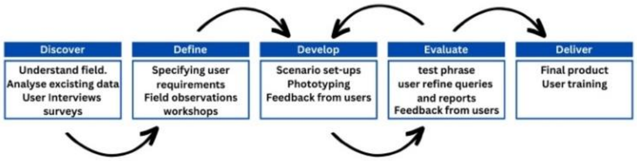


Fig. 6. Collaborative AI system build through a user-centered approach.

IV. Challenges and Opportunities of AI-Driven Monitoring Systems in Smart City Healthcare

A. Challenges

The development and deployment of user-centered AI monitoring systems in smart health is a complex process that is faced with some significant challenges. The main challenges are highlighted here including data privacy and security, data quality and availability, interoperability, clinical validity and user acceptance.

1) Data privacy and security: Health data is highly sensitive and personal information that must be protected.

Ensuring the privacy and security of health data is a major challenge in the development and deployment of user-centered AI monitoring systems [32]. AI systems have a significant and ongoing security challenge due to the complexity of the data and increasing frequency of network attacks. Security faults, for example, an area of concern since medical data, which is gathered by different types of sensors, is subject to errors. These faults can make it challenging to understand or diagnose the patient's condition. Furthermore, real-time monitoring necessitates a fast and dependable network connection, which can be difficult to achieve in rural areas and on low-power systems [33].

2) Data quality and availability: For AI monitoring systems to be effective, they must be trained on high-quality data. Health data consisting of doctors' notes, observations, results of pathology, radiology images, and signals are typically not very extensive and can be challenging to access [64], [65]. A significant challenge is ensuring that the data used to train AI systems is accurate, up-to-date and widely available.

3) Interoperability: AI monitoring systems must be able to integrate with existing health IT systems to be useful [66]. This necessitates that the systems exchange data and communicate with one another in a seamless manner. Ensure that AI monitoring systems are interoperable with existing health IT systems can be difficult, especially in environments where different systems use different data standards.

4) Clinical validity: For AI monitoring systems to be effective in clinical settings, they must be able to provide accurate and reliable information. This necessitates system validation and the accuracy and dependability of the results [64], [67]. It is crucial to ensure AI monitoring systems are clinically valid, especially in the health sector since inaccurate diagnosis and treatment can have severe consequences.

5) User acceptance: AI monitoring systems must be usable and provide transparency to end-users to be effective (Miotto et al., 2017). This includes ensuring that the systems are user-friendly and that users trust the information that they provide [60], [61]. Ensuring that AI monitoring systems are acceptable to end-users can be challenging, particularly in the context of health where users may be concerned about the reliability of the information provided.

6) Trust: The success of user-centered AI monitoring systems in smart health greatly depends on building trust with the users. Users will not accept the results of AI systems if they do not trust the system [68].

Despite of these challenges, the development and deployment of AI-based monitoring systems for smart health remains a significant area of research and development. Identifying and addressing these challenges can help develop user-centered AI monitoring systems in smart health that are effective, usable, and trustworthy, improving patient health.

B. Opportunities

AI-driven monitoring systems have made a substantial impact on people's lives, attracting the interest of researchers. Although there are several challenges in the development and deployment of AI-based monitoring systems there are also numerous advantages, such as personalized Health Monitoring, the delivery of remote diagnosis and treatment, and the collection of real-time data without interfering with regular activities. The opportunities are demonstrated in this section.

1) Personalized health monitoring: These systems use AI technologies to collect, process, and analyze health-related data, allowing for personalized health monitoring and support to the users. The automatic collection of data, such as vital signs, physical activity levels, and sleep patterns, can help detect health-related issues in real-time [8], [49], [61], [69].

2) Real-time feedback: These systems can provide real-time feedback to the user, allowing them to make informed decisions about their health. They can help identify potential health risks and provide personalized health recommendations, such as lifestyle changes or exercise regimens, to help improve overall health [3], [32], [61].

*3) Non-intrusive monitoring***:** User-centered AI monitoring systems in smart health are designed to be non-intrusive, allowing users to monitor their health without interference with their daily activities [33].

4) Improved health outcomes: These systems can help individuals manage their health more effectively and make lifestyle changes to improve overall health outcomes. By providing real-time feedback and personalized health recommendations, these systems can help individuals make informed decisions about their health and well-being [26], [59].

5) Increased access to health services: User-centered AI monitoring systems in smart health can help increase access to health services for individuals in remote or underserved areas. By allowing for remote monitoring and analysis of health-related data, these systems can help bridge the gap in access to health services for these individuals [3], [33].

6) Increased efficiency and reduced costs: The use of AI technologies can help streamline healthcare processes and reduce costs. By automating the collection and analysis of health-related data, these systems can reduce the need for manual data entry and analysis, saving time and resources [3], [26], [70]-[72].

The current research indicates that while user-centered AI monitoring systems hold significant potential for transforming the healthcare industry, there are also certain challenges related to their development and deployment. One of the key challenges is making sure the system is user-friendly and simple to use, as many patients may be intimidated by technology [61]. Furthermore, there are privacy and security concerns associated with the collection and storage of health-related data that must be addressed for these systems to be safe and secure to use [33]. Other concerns include Clinical Validity, Data Quality and Availability and Interoperability of these systems. However, there are some great opportunities in developing and deploying user-centered AI monitoring systems in smart health. One of the main advantages of these systems is their ability to provide users with real-time feedback, allowing them to make informed health decisions [3]. Furthermore, these systems can offer personalized health recommendations, such as lifestyle changes or exercise programs [69] to assist users in improving their overall health.

V. Future Directions for User-centered AI Monitoring in Smart Health

Within this section, potential future directions will be presented regarding the challenges related to developing and implementing user-centered AI monitoring systems in the context of smart health. An overview of the challenges along with the possible future directions is illustrated in Table II.

TABLE II. Overview of Challenges and Possible Future Directions

Challenges	Examples	Potential future directions
Data Privacy and Security	Health data is highly sensitive and must be protected from security faults, such as sensor errors.	Invest in strong security measures, such as encryption, secure communication, and monitoring for breaches and unauthorized access.
Data Quality and Availability	Ensuring that data used to train AI systems is accurate, up-to-date and widely available.	Invest in data infrastructure that makes obtaining, storing, and managing health data easier.
Interoperability	AI monitoring systems must integrate with existing health IT to exchange data and communicate.	Follow common data standards and invest in tools that make integrating AI monitoring systems with existing health IT systems easier.
Clinical Validity	Ensure AI monitoring systems are clinically valid.	Invest in research and validation of AI monitoring systems to ensure they continue to provide valid and trustworthy information.
User Acceptance	AI monitoring systems must be usable and provide transparency.	Involve end-users in AI monitoring systems to ensure usability and trust. Education and training are needed to promote adoption.
Trust	Users must trust the system to accept the results of AI systems.	Develop user-friendly systems that provide accurate and reliable health-related data and are transparent in their operation.

It is critical to implement strong security measures to protect health data to address the challenge of data privacy and security. This can include employing encryption and secure communication protocols, as well as monitoring for breaches and unauthorized access on a regular basis. Furthermore, it is critical to be transparent about the use of health data and to obtain patient consent before using their data for AI training or monitoring. Data Quality and Availability are other critical factors to consider when developing high-quality AI monitoring systems. To tackle these issues, it may be necessary to invest in data infrastructure that makes obtaining, storing, and managing health data easier. Furthermore, health data sharing may be necessary to encourage the development of high-quality monitoring AI systems. As a solution to the interoperability challenge, it may be necessary to adopt common data standards and invest in tools that make integrating AI monitoring systems with existing health IT systems easier. Furthermore, it is critical to foster open communication and information sharing among different organizations to ensure that AI monitoring systems can work seamlessly with existing health IT systems. The challenge of clinical validity can be addressed by investing in the research and validation of AI monitoring systems. Additionally, ongoing monitoring and evaluation of the systems is required to ensure that they continue to provide valid and trustworthy information. User acceptance is an important factor that can determine the success of user-centered AI monitoring systems in smart health. To address this issue, we must ensure that end-users are involved in the design and development of AI monitoring systems. This can include user testing and gathering user feedback to ensure that the systems are usable and that users trust the information provided. Furthermore, education and training may be required to promote widespread adoption of AI monitoring systems. Finally, building trust with the users is a critical factor in the success of user-centered AI monitoring systems in smart health. To build trust, these systems must be user-friendly and easy to use, and provide accurate and reliable health-related data. Additionally, the systems must be transparent in their operation and provide clear explanations for the health recommendations provided to the user.

User-centered AI monitoring systems in smart health hold significant potential for transforming the healthcare industry by providing real-time health monitoring and support to patients. User-centered AI monitoring systems are designed to improve the patient experience in healthcare by using technology to collect and analyze data from patients. These systems use artificial intelligence algorithms to monitor patients and provide real-time information about their health status. However, careful consideration must be given to the design and deployment of these systems to ensure that they are user-friendly, secure, and provide accurate and reliable health-related data. By addressing these challenges, it is possible to develop user-centered AI monitoring systems in smart health that are effective, usable, and trustworthy, thereby improving patient health outcomes.

VI. Discussion and Conclusion

In recent years, smart health technologies have been gaining popularity as a means of providing high-quality patient care, including user-centered AI monitoring services. These systems hold great potential to transform the health industry by providing real-time health monitoring and support to patients. Based on the results of this research paper, there appear to be several key factors that need to be addressed when developing user-centered AI monitoring systems for smart health. These include data protection and security, data quality and availability, interoperability, clinical validity, user acceptance, and building trust among users. This is consistent with the results of previous studies that have emphasized the importance of these factors in designing effective AI healthcare monitoring systems. The current paper differs from previous studies in that it emphasizes the importance of user-centered design and involving end-users in the development process. Although previous studies have also recognized the importance of user-centered design, this paper highlights the need to involve end users in the design and development of these systems. This suggests that there may be a need for greater collaboration between developers and end-users in the development of user-centered AI monitoring systems for smart health. The research also indicates the importance of user research, user feedback, and continuous evaluation in ensuring that these systems are developed in a manner that prioritizes the needs and preferences of users. User-centered design has been shown to produce more efficient and user-friendly health monitoring systems and incorporating user feedback throughout the development process is crucial to producing solutions that meet the needs of patients and caregivers. Overall, the findings suggest that, while user-centered AI monitoring systems have significant potential for transforming healthcare, they must be designed and deployed with care to ensure that they are effective, usable, and trustworthy. By addressing the challenges identified in this study, it may be possible to develop AI monitoring systems that are better able to support patients and improve health outcomes.

Future research in this field should focus on developing and deploying artificial intelligence (AI) technologies that can collect, process, and analyze health-related data in a secure and privacy-sensitive manner. Furthermore, research should be conducted to identify ways to increase trust in these systems, such as developing user-friendly interfaces, providing clear and transparent information about how data is collected and processed, and ensuring that the data used to train these systems is diverse and representative of the population it is intended to serve. In conclusion, how we treat and monitor patients is impacted by the continually shifting number of devices in a smart city. As a result, developing smart devices often results in compatibility and security issues. Artificial intelligence and deep learning approaches can be used to improve the management and coordination of device data and network models to address this challenge and ensure smooth connectivity between smart devices.

REFERENCES

[1] S. S. Raju and R. C. Kumar, “EAI Endorsed Transactions on Smart Cities Artificial Intelligence in Smart cities and Healthcare,” 2022, doi: 10.4108/eetsc.v6i3.2275.

[2] Z. Asim, “Shaping Healthcare System Under Industry 5.0: Trends and Barriers,” Sudan Journal of Medical Sciences, Sep. 2022, doi: 10.18502/sjms.v17i3.12115.

[3] S. Tian, W. Yang, J. M. le Grange, P. Wang, W. Huang, and Z. Ye, “Smart healthcare: making medical care more intelligent,” J Glob Health, vol. 3, no. 3, pp. 62–65, 2019, doi: 10.1016/J.GLOHJ.2019.07.001.

[4] M. Saranya, “A Survey on Health Monitoring System by using IOT,” Int J Res Appl Sci Eng Technol, vol. 6, no. 3, pp. 778–782, Mar. 2018, doi: 10.22214/ijraset.2018.3124.

[5] A. A. Mohammed, M. A. Burhanuddin, M. Saad Talib, M. E. Hameed, and M. F. Ali, “A Review on IoT-Based Healthcare Monitoring Systems for Patient in Remote Environments,” European Journal of Molecular & Clinical Medicine, vol. 07, no. 03, 2022.

[6] M. Raza, M. Awais, N. Singh, M. Imran, and S. Hussain, “Intelligent IoT Framework for Indoor Healthcare Monitoring of Parkinson’s Disease Patient,” IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 593–602, Feb. 2021, doi: 10.1109/JSAC.2020.3021571.

[7] V. Mody and V. Mody, “Mental Health Monitoring System using Artificial Intelligence: A Review,” in 2019 IEEE 5th International Conference for Convergence in Technology, I2CT 2019, 2019. doi: 10.1109/I2CT45611.2019.9033652.

[8] Z. Ashfaq et al., “Embedded AI-Based Digi-Healthcare,” Applied Sciences (Switzerland), vol. 12, no. 1, 2022, doi: 10.3390/app12010519.

[9] M. Tabaa, F. Monteiro, H. Bensag, and A. Dandache, “Green Industrial Internet of Things from a smart industry perspectives,” Energy Reports, vol. 6, pp. 430–446, Nov. 2020, doi: 10.1016/J.EGYR.2020.09.022.

[10] N. Zakaria and J. A., “Smart City Architecture: Vision and Challenges,” International Journal of Advanced Computer Science and Applications, vol. 6, no. 11, 2015, doi: 10.14569/ijacsa.2015.061132.

[11] J. Galih, P. Negara, and A. W. R. Emanuel, “A Conceptual Smart City Framework for Future Industrial City in Indonesia,” 2019. [Online]. Available: www.ijacsa.thesai.org.

[12] M. K. Al-Azzam and M. B. Alazzam, “Smart city and Smart-Health framework, challenges and opportunities,” International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, 2019, doi: 10.14569/ijacsa.2019.0100223.

[13] A. Gharaibeh et al., “Smart Cities: A Survey on Data Management, Security, and Enabling Technologies,” IEEE Communications Surveys and Tutorials, vol. 19, no. 4. Institute of Electrical and Electronics Engineers Inc., pp. 2456–2501, Oct. 01, 2017. doi: 10.1109/COMST.2017.2736886.

[14] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, “Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future,” J Clean Prod, vol. 274, 2020, doi: 10.1016/j.jclepro.2020.122877.

[15] H. Singh and S. J. Miah, “Smart education literature: A theoretical analysis,” Educ Inf Technol (Dordr), vol. 25, no. 4, 2020, doi: 10.1007/s10639-020-10116-4.

[16] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, “A review of machine learning and IoT in smart transportation,” Future Internet, vol. 11, no. 4. 2019. doi: 10.3390/FI11040094.

[17] A. I. Voda and L. D. Radu, “How can artificial intelligence respond to smart cities challenges?,” in Smart Cities: Issues and Challenges Mapping Political, Social and Economic Risks and Threats, 2019. doi: 10.1016/B978-0-12-816639-0.00012-0.

[18] E. M. Ouafiq, M. Raif, A. Chehri, and R. Saadane, “Data Architecture and Big Data Analytics in Smart Cities,” Procedia Comput Sci, vol. 207, pp. 4123–4131, Jan. 2022, doi: 10.1016/j.procs.2022.09.475.

[19] H. M. K. K. M. B. Herath and M. Mittal, “Adoption of artificial intelligence in smart cities: A comprehensive review,” International Journal of Information Management Data Insights, vol. 2, no. 1, p. 100076, Apr. 2022, doi: 10.1016/j.jjimei.2022.100076.

[20] P. Mishra, P. Thakur, and G. Singh, “Sustainable Smart City to Society 5.0: State-of-the-Art and Research Challenges,” SAIEE Africa Research Journal, vol. 113, no. 4, pp. 152–164, Dec. 2022, doi: 10.23919/SAIEE.2022.9945865.

[21] A. M. Shahat Osman and A. Elragal, “Smart Cities and Big Data Analytics: A Data-Driven Decision-Making Use Case,” Smart Cities, vol. 4, no. 1, pp. 286–313, Feb. 2021, doi: 10.3390/smartcities4010018.

[22] S. E. Bibri and J. Krogstie, “The emerging data–driven Smart City and its innovative applied solutions for sustainability: the cases of London and Barcelona ,” Energy Informatics, vol. 3, no. 1, p. 5, Dec. 2020, doi: 10.1186/s42162-020-00108-6.

[23] N. Gardner and L. Hespanhol, “SMLXL: Scaling the smart city, from metropolis to individual,” City, Culture and Society, vol. 12, pp. 54–61, Mar. 2018, doi: 10.1016/j.ccs.2017.06.006.

[24] S. Andreani, M. Kalchschmidt, R. Pinto, and A. Sayegh, “Reframing technologically enhanced urban scenarios: A design research model towards human centered smart cities,” Technol Forecast Soc Change, vol. 142, 2019, doi: 10.1016/j.techfore.2018.09.028.

[25] A. Abella, M. Ortiz-de-Urbina-Criado, and C. De-Pablos-Heredero, “A model for the analysis of data-driven innovation and value generation in smart cities’ ecosystems,” Cities, vol. 64, pp. 47–53, Apr. 2017, doi: 10.1016/j.cities.2017.01.011.

[26] M. Nasr, Md. M. Islam, S. Shehata, F. Karray, and Y. Quintana, "Smart Healthcare in the Age of AI: Recent Advances, Challenges, and Future Prospects," IEEE Access, vol. 9, pp. 145248–145270, 2021, doi: 10.1109/ACCESS.2021.3118960.

[27] A. Abdelaziz, A. S. Salama, and A. M. Riad, "A hybrid intelligent model for enhancing healthcare services on cloud environment," International Journal of Advanced Computer Science and Applications, vol. 9, no. 11, 2018, doi: 10.14569/IJACSA.2018.091105.

[28] T. Alam, "mHealth Communication Framework using Blockchain and IoT Technologies," no. June, 2020, doi: 10.20944/preprints202006.0180.v1.

[29] P. Valsalan, N. ul Hasan, I. Baig, and M. Zghaibeh, "Remote Healthcare Monitoring using Expert System," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, p. 2022, 2022, doi: 10.14569/IJACSA.2022.0130370.

[30] M. Bhandari, T. Zeffiro, and M. Reddiboina, "Artificial intelligence and robotic surgery: Current perspective and future directions," Current Opinion in Urology, vol. 30, no. 1. Lippincott Williams and Wilkins, pp. 48–54, Jan. 01, 2020. doi: 10.1097/MOU.0000000000000692.

[31] R. F. Mansour, A. el Amraoui, I. Nouaouri, V. G. DIaz, D. Gupta, and S. Kumar, "Artificial Intelligence and Internet of Things Enabled Disease Diagnosis Model for Smart Healthcare Systems," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3066365.

[32] M. J. Kim, M. E. Cho, and H. J. Jun, "Developing Design Solutions for Smart Homes Through User-Centered Scenarios," Front Psychol, vol. 11, 2020, doi: 10.3389/fpsyg.2020.00335.

[33] A. V. L. N. Sujith, G. S. Sajja, V. Mahalakshmi, S. Nuhmani, and B. Prasanalakshmi, "Systematic review of smart health monitoring using deep learning and Artificial intelligence," Neuroscience Informatics, vol. 2, no. 3, 2022, doi: 10.1016/j.neuri.2021.100028.

[34] K. Paranjape, M. Schinkel, R. N. Panday, J. Car, and P. Nanayakkara, "Introducing artificial intelligence training in medical education," JMIR Medical Education, vol. 5, no. 2. 2019. doi: 10.2196/16048.

[35] J. Grunhut, A. T. Wyatt, and O. Marques, "Educating Future Physicians in Artificial Intelligence (AI): An Integrative Review and Proposed Changes," J Med Educ Curric Dev, vol. 8, p. 238212052110368, Jan. 2021, doi: 10.1177/23821205211036836.

[36] M. S. Hossain, "Patient status monitoring for smart home healthcare," in 2016 IEEE International Conference on Multimedia and Expo Workshop, ICMEW 2016, 2016. doi: 10.1109/ICMEW.2016.7574719.

[37] A. N. Navaz, M. A. Serhani, H. T. el Kassabi, N. Al-Qirim, and H. Ismail, "Trends, Technologies, and Key Challenges in Smart and Connected Healthcare," IEEE Access, vol. 9, pp. 74044–74067, 2021, doi: 10.1109/ACCESS.2021.3079217.

[38] A. K. Tripathy, A. G. Mohapatra, S. P. Mohanty, E. Kougianos, A. M. Joshi, and G. Das, "EasyBand: A Wearable for Safety-Aware Mobility during Pandemic Outbreak," IEEE Consumer Electronics Magazine, vol. 2248, no. c, pp. 10–14, 2020, doi: 10.1109/MCE.2020.2992034.

[39] S. Sun et al., "Using smartphones and wearable devices to monitor behavioural changes during COVID-19," vol. 44, no. 0, 2020, [Online]. Available: http://arxiv.org/abs/2004.14331

[40] S. Lee and H. S. Kim, "Prospect of artificial intelligence based on electronic medical record," Journal of Lipid and Atherosclerosis, vol. 10, no. 3. Korean Society of Lipid and Atherosclerosis, pp. 282–290, Sep. 01, 2021. doi: 10.12997/JLA.2021.10.3.282.

[41] K. K. Mak and M. R. Pichika, "Artificial intelligence in drug development: present status and future prospects," Drug Discovery Today, vol. 24, no. 3. 2019. doi: 10.1016/j.drudis.2018.11.014.

[42] B. R. Beck, B. Shin, Y. Choi, S. Park, and K. Kang, "Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model," Comput Struct Biotechnol J, vol. 18, pp. 784–790, 2020, doi: 10.1016/j.csbj.2020.03.025.

[43] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," Mechanical Systems and Signal Processing, vol. 115. 2019. doi: 10.1016/j.ymssp.2018.05.050.

[44] M. Alhussein and G. Muhammad, "Automatic Voice Pathology Monitoring Using Parallel Deep Models for Smart Healthcare," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2905597.

[45] S. M. Sefa-Yeboah, K. Osei Annor, V. J. Koomson, F. K. Saalia, M. Steiner-Asiedu, and G. A. Mills, "Development of a Mobile Application Platform for Self-Management of Obesity Using Artificial Intelligence Techniques," Int J Telemed Appl, vol. 2021, 2021, doi: 10.1155/2021/6624057.

[46] T. Malche et al., "Artificial Intelligence of Things- (AIoT-) Based Patient Activity Tracking System for Remote Patient Monitoring," J Healthc Eng, vol. 2022, 2022, doi: 10.1155/2022/8732213.

[47] P. Rajan Jeyaraj and E. R. S. Nadar, "Smart-Monitor: Patient Monitoring System for IoT-Based Healthcare System Using Deep Learning," IETE J Res, vol. 68, no. 2, 2022, doi: 10.1080/03772063.2019.1649215.

[48] M. K. Hassan, A. I. el Desouky, S. M. Elghamrawy, and A. M. Sarhan, "A Hybrid Real-time remote monitoring framework with NB-WOA algorithm for patients with chronic diseases," Future Generation Computer Systems, vol. 93, 2019, doi: 10.1016/j.future.2018.10.021.

[49] S. P. Chatrati et al., "Smart home health monitoring system for predicting type 2 diabetes and hypertension," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 3, 2022, doi: 10.1016/j.jksuci.2020.01.010.

[50] M. Alhussein, "Monitoring Parkinson's Disease in Smart Cities," IEEE Access, vol. 5, 2017, doi: 10.1109/ACCESS.2017.2748561.

[51] Z. Fu, S. Hong, R. Zhang, and S. Du, "Artificial-intelligence-enhanced mobile system for cardiovascular health management," Sensors (Switzerland), vol. 21, no. 3, 2021, doi: 10.3390/s21030773.

[52] H. Pandey and S. Prabha, "Smart Health Monitoring System using IOT and Machine Learning Techniques," in 2020 6th International Conference on Bio Signals, Images, and Instrumentation, ICBSII 2020, 2020. doi: 10.1109/ICBSII49132.2020.9167660.

[53] M. Zia Ur Rahman et al., "Real-time artificial intelligence based health monitoring, diagnosing and environmental control system for COVID-19 patients," Mathematical Biosciences and Engineering, vol. 19, no. 8, pp. 7586–7605, 2022, doi: 10.3934/mbe.2022357.

[54] N. Nasser, Q. Emad-ul-Haq, M. Imran, A. Ali, I. Razzak, and A. Al-Helali, "A smart healthcare framework for detection and monitoring of COVID-19 using IoT and cloud computing," Neural Comput Appl, Sep. 2021, doi: 10.1007/s00521-021-06396-7.

[55] A. Rghioui, J. Lloret, S. Sendra, and A. Oumnad, "A smart architecture for diabetic patient monitoring using machine learning algorithms," Healthcare (Switzerland), vol. 8, no. 3, 2020, doi: 10.3390/healthcare8030348.

[56] G. Huang, Y. Fang, X. Wang, Y. Pei, and B. Horn, "A Survey on the Status of Smart Healthcare from the Universal Village Perspective," in 4th IEEE International Conference on Universal Village 2018, UV 2018, 2019. doi: 10.1109/UV.2018.8642125.

[57] Naveen, R. K. Sharma, and A. R. Nair, "IoT-based Secure Healthcare Monitoring System," in Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019, 2019. doi: 10.1109/ICECCT.2019.8868984.

[58] B. Quispe-Lavalle, F. Sierra-Liñan, M. Cabanillas-Carbonell, and N. Wiener, "Mobile Applications for the Implementation of Health Control against Covid-19 in Educational Centers, a Systematic Review of the Literature," 2022. [Online]. Available: www.ijacsa.thesai.org

[59] B. Marent et al., "Development of an mHealth platform for HIV care: Gathering user perspectives through co-design workshops and interviews," JMIR Mhealth Uhealth, vol. 6, no. 10, 2018, doi: 10.2196/mhealth.9856.

[60] Y. Koumpouros, "User-Centric Design Methodology for mHealth Apps: The PainApp Paradigm for Chronic Pain," Technologies (Basel), vol. 10, no. 1, 2022, doi: 10.3390/technologies10010025.

[61] G. Molina-Recio, R. Molina-Luque, A. M. Jiménez-García, P. E. Ventura-Puertos, A. Hernández-Reyes, and M. Romero-Saldaña, "Proposal for the User-Centered Design Approach for Health Apps Based on Successful Experiences: Integrative Review," JMIR Mhealth Uhealth, vol. 8, no. 4, p. e14376, Apr. 2020, doi: 10.2196/14376.

[62] L. M. Kopf and J. Huh-Yoo, "A User-Centered Design Approach to Developing a Voice Monitoring System for Disorder Prevention,"

Journal of Voice, vol. 37, no. 1, pp. 48–59, Jan. 2023, doi: 10.1016/j.jvoice.2020.10.015.

[63] F. Tiersen et al., “Smart Home Sensing and Monitoring in Households With Dementia: User-Centered Design Approach,” JMIR Aging, vol. 4, no. 3, p. e27047, Aug. 2021, doi: 10.2196/27047.

[64] S. S. R. Abidi and S. R. Abidi, “Intelligent health data analytics: A convergence of artificial intelligence and big data,” Healthc Manage Forum, vol. 32, no. 4, pp. 178–182, 2019, doi: 10.1177/0840470419846134.

[65] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: Review, opportunities and challenges,” Brief Bioinform, vol. 19, no. 6, pp. 1236–1246, 2017, doi: 10.1093/bib/bbx044.

[66] T. Davenport and Ravi Kalakota, “The Potential for Artificial Intelligence in Healthcare,” Future Healthc J, vol. 6, no. 2, pp. 94–98, 2019.

[67] S. H. Park, J. Choi, and J. S. Byeon, “Key principles of clinical validation, device approval, and insurance coverage decisions of artificial intelligence,” Korean Journal of Radiology, vol. 22, no. 3. 2021. doi: 10.3348/kjr.2021.0048.

[68] A. Hassan, M. A. A. Abdulhak, R. bin Sulaiman, and H. Kahtan, “User centric explanations: A breakthrough for explainable models,” 2021 International Conference on Information Technology, ICIT 2021 - Proceedings, vol. 0, pp. 702–707, 2021, doi: 10.1109/ICIT52682.2021.9491641.

[69] M. Kim, Y. Kim, and M. Choi, “Mobile health platform based on user-centered design to promote exercise for patients with peripheral artery disease,” BMC Med Inform Decis Mak, vol. 22, no. 1, p. 206, Dec. 2022, doi: 10.1186/s12911-022-01945-z.

[70] S. Saba Raoof and M. A. S. Durai, “A Comprehensive Review on Smart Health Care: Applications, Paradigms, and Challenges with Case Studies,” Contrast Media Mol Imaging, vol. 2022, 2022, doi: 10.1155/2022/4822235.

[71] A. Hassan, R. Sulaiman, M. A. Abdulgabber, and H. Kahtan, "Towards User-Centric Explanations for Explainable Models: A Review," Journal of Information System and Technology Management, vol. 6, no. 22, 2021, doi: DOI: 10.35631/JISTM.622004.

[72] H. Kahtan, K. Z. Zamli, W. N. A. W. A. Fatthi, A. Abdullah, M. Abdulleteef, and N. S. Kamarulzaman, "Heart Disease Diagnosis System Using Fuzzy Logic," presented at the Proceedings of the 2018 7th International Conference on Software and Computer Applications, Kuantan, Malaysia, 2018. [Online]. Available: https://doi.org/10.1145/3185089.3185118.

Investigation of Combining Deep Learning Object Recognition with Drones for Forest Fire Detection and Monitoring

Mimoun YANDOUZI[1], Mounir GRARI[2], Mohammed BERRAHAL[3], Idriss IDRISSI[4],
Omar MOUSSAOUI[5], Mostafa AZIZI[6], Kamal GHOUMID[7], Aissa KERKOUR ELMIAD[8]
Lab. LSI, ENSAO, Mohammed First University, Oujda, Morocco[1, 7]
Lab. MATSI, ESTO, Mohammed First University, Oujda, Morocco[2, 3, 4, 5, 6]
Lab. LARI, FSO, Mohammed First University, Oujda, Morocco[8]

***Abstract*—Forest fires are a global environmental problem that can cause significant damage to natural resources and human lives. The increasing frequency and severity of forest fires have resulted in substantial losses of natural resources. To mitigate this, an effective fire detection and monitoring system is crucial. This work aims to explore and review the current advancement in the field of forest fire detection and monitoring using both drones or unmanned aerial vehicles (UAVs), and deep learning techniques. The utilization of drones fully equipped with specific sensors and cameras provides a cost-effective and efficient solution for real-time monitoring and early fire detection. In this paper, we conduct a comprehensive analysis of the latest developments in deep learning object detection, such as YOLO (You Only Look Once), R-CNN (Region-based Convolutional Neural Network), and their variants, with a focus on their potential application in the field of forest fire monitoring. The performed experiments show promising results in multiple metrics, making it a valuable tool for fire detection and monitoring.**

Keywords—Forest fire; deep learning; drones; unmanned aerial vehicles; object detection; YOLO; Faster R-CNN

I. Introduction

Forests are critical for our planet. They regulate our climate, purify our air and water, and are home to countless plants and animals. Sadly, forests around the world are under real threats from climate change, deforestation, and other human activities. One of the most devastating impacts on forests are wildfires. More recently, forest fires have become an annual phenomenon across the world. Statistics show that millions of acres of forests are yearly burnt. This has caused tremendous loss of forest resources, major economic damages to forest organizations, and lives of humans and animals. The rise in wildfires is largely attributed to climate change. Warmer temperatures and drier conditions make it easier for fires to start and spread. When a wildfire starts, it can spread quickly through the entire forest. Wildfires can have a significant impact on the environment and the economy. They can permanently damage forests, which can take years to be restored. In addition, wildfires can result in substantial financial damage as companies and industries relying on forests are compelled to shut down or move [1].

A wide range of techniques are used to detect and monitor forest fires, mainly we retain two main techniques based on sensors and imagery. The first one is based on deployed sensors that can detect environmental measurements such as temperature, humidity, gas levels, etc. These sensors, which may be strategically positioned throughout the forest, will notify authorities if they detect any fire. The second technique is imagery-based; it uses images coming from fixed cameras, satellites, or drones. It provides authorities with a bird-eye view of the fire and its precise location [2, 3].

Recently, with the great advances in deep learning (DL) and its applications, new opportunities are available to the problem-solving of computer vision in the field of forest fires monitoring. We remind that the traditional approach uses only visual analysis of images taken by satellites or aerial cameras. The existing system only detects wildfires within the camera view; it cannot identify the exact location of the fire [4]. This gap led us to propose, in this paper, an automated system that helps identify potential forest fires using object detection algorithms, such as YOLO (You Only Look Once) and R-CNN (Region-based Convolutional Neural Net-work).

The rest of this paper is organized as follows: The Section II presents the background of our research; the Section III gives a detailed overview of related works; the Section IV describes our proposed method; and before concluding this work, the Section V presents the obtained results and the discussion of the efficiency of our proposed system.

II. Background

A. Computer Vision (CV)

The biological vision is an inspiring model for computer vision. The mammalian visual system can decipher a complex scene in an instant, sophisticated enough to distinguish, ripe fruit from a poisonous berry, or fire from the sun. Similarly, computer vision is an emerging field that is rapidly evolving, making significant progress in recent years.

In general, computer vision is concerned with the automatic extraction, analysis, and understanding of information from images. This can be a difficult task, as images are often cluttered and contain complex noises. However, recent advances in ma-chine learning have allowed for significant progress in this area. For example, deep learning (a subset of machine learning) is a powerful technique that has been used to achieve amazing results in computer vision [5].

There are many different applications of computer vision, including object recognition, face detection, and scene understanding [6].

B. Deep Learning (DL)

Deep learning, inspired by the brain, allows a computer to deeply learn from data by creating relevant large neuronal network models [7].

Deep learning is a relatively new field of machine learning, and it is already having a major impact. It is used in a variety of applications, including image recognition, speech recognition, and natural language processing [8].

Deep learning is a powerful tool that can be used to solve many complex problems. However, requires an important processing power and a large amount of data in order to train fast its models and learn effectively [9].

C. Object Detection

Object detection is the task of detecting instances of objects in an image, regardless of their position or orientation. This can be a very difficult task, as there can be a great deal of variation in the appearance of objects. For example, two different people can have very different opinions on what constitutes a "tree". Despite this challenge, object detection is a very important task in many applications, such as security, car self-driving, and robotics. In addition to detecting objects in an image, CV can also be used in following their movement. This is very helpful in keeping track of people or vehicles in a security system, or in avoiding obstacles for autonomous vehicles. There are many different types of techniques for object detection, but one of the most popular ones are those DL-based. DL is able to learn from both unstructured and unlabeled data. This makes it ideal for object detection, as it can learn to identify objects from a variety of different sources [10]. Furthermore, one of the advantages of deep learning for object detection is that it can learn to identify objects that are not easily detectable by humans. For example, deep learning can be used to detect objects in images that are blurry or have low contrast. Additionally, it can be used to detect objects that are occluded or partially hidden. Another advantage of deep learning for object detection is that it can learn to identify objects from a variety of different views. This is helpful because it means that the object detector will be more robust and will be able to identify objects even when they are not perfectly visible.

There are a few different object detection models that are popular among developers and researchers [11], including YOLO, and Faster R-CNN models. Each of these models has its own strengths and weaknesses. Object detection algorithms can be broadly classified into two categories: one-stage and two-stage. One-stage algorithms detect objects in a single step, whereas two-stage algorithms divide the process into two phases. The first phase uses a classifier to determine potential object positions, and the second phase employs a region proposal method to pinpoint the objects' most probable locations, as depicted in Fig. 1. One-stage algorithms are faster but less accurate, while two-stage algorithms are slower, but more accurate. One-stage algorithms are favored in real-time scenarios where processing speed is more prioritized than accuracy, whereas two-stage ones are utilized when accuracy is of utmost importance.

The components of object detection architecture typically comprise five parts: the input, backbone, neck, dense layer, and sparse layer. The input is the image fed into the network, which is usually pre-processed for standardization such as resizing or normalization. The backbone, which is typically a pre-trained convolutional neural network, extract features from the input image. The neck combines the extracted features from the backbone. The dense layer is a fully connected layer that generates the final object detection results using the combined features from the neck. The Sparse layer performs similarly to the dense layer, with one key difference: its connections are sparse, meaning that not every neuron in the current layer is linked to every neuron in the prior layer. This architectural design is often used to optimize neural network performance by reducing the number of parameters and improving computational efficiency. The Sparse layer is commonly employed in feature extraction and object detection tasks, and it also outputs the final object detection results.

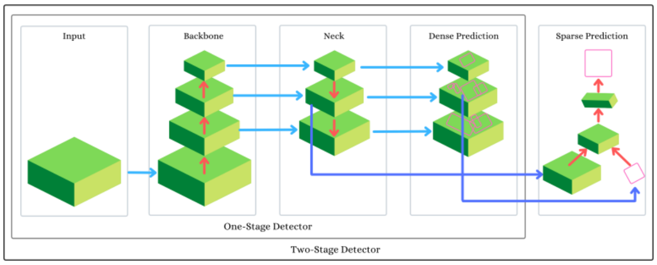


Fig. 1. Object detection architectures: one-stage and two-stage approaches.

1) R-CNN (Region-based CNNs) and its variants: R-CNN (Region-based Convolutional Neural Network) and its variants are a family of deep neural network architectures for image classification and object detection [12, 13]. R-CNN was originally proposed for object detection in natural images. The main idea of R-CNN is to use a CNN to process a region of an image, extract features from it, and then classify it. The R-CNN architecture has been successful in many object detection tasks, including detecting objects in both natural images and video.

Variants of R-CNN include Fast R-CNN [14] and Faster R-CNN [15]. Fast R-CNN is an improvement over R-CNN that uses a Region Proposal Network (RPN) to propose regions, rather than using a sliding window. Faster R-CNN is an even further improvement that shares convolutional layers between the RPN and the classifier, resulting in even faster performance [16].

2) YOLO (You Only Look Once): The YOLO object detection system is a widely used method for detecting objects in images and videos, originally created by Joseph Redmon and Ali Farhadi [17, 18]. YOLO is a real-time object detection system that is fast and accurate. YOLO has been used in a variety of applications and is considered to be the best of object detection systems of the literature.

YOLO is a deep learning model that is able to effectively identify objects in images and video frames. This is done by first partitioning an image into a set of grid cells, and then using a specially designed neural network to predict the bounding box coordinates and class probabilities for each cell.

YOLO is constantly being improved and its new versions are being released. The first version, YOLOv1, was released in 2016. YOLOv2 was released in 2016. YOLOv3 was released in 2018. YOLOv4 was released in 2020. YOLOv5 was released in 2021. And the latest YOLOv6 [19], YOLOv7 [20], and YOLOv8 [21] (v6 and v7 released in 2022, and v8 released in 2023), included further improvements in terms of accuracy, speed, and they introduced a new segmentation pipeline.

III. Related Works

Zheng et al. [22] worked on the classification of dynamic scenes to facilitate the process of detection and tracking of objects and thus improve the performance of visual surveillance. The proposed model, Bi-heterogeneous convolutional neural network (Bi-CNN), extracts spatial and temporal information from the video sequences to categorize them. The model was trained and tested on a dataset composed of drone videos. They achieved a mean accuracy of 93%.

Jiao et al. [23] proposed a model based on YOLOv3 that can be deployed in architecture using UAVs. The developed platform is presented with all the technical choices of the UAV and the analysis station. The DL model is therefore deployed on a ground station. They were able to achieve a speed of photo transmission of 3.2 images per second with a fire recognition rate of 83%. The same authors, in a second work [24], upgraded the equipment used to reach a transmission speed of 30 frames per second and an accuracy of 91%.

Lohit et al. [25] used object detection to solve a post-fire problem related to reforestation. The authors use a drone equipped with a Raspberry Pi board on which deep learning models are deployed. A comparative study is performed between the models DenseNet121, Resnet152, and MobileNetv2. The dataset used is composed of UAV images, UAV Dataset (from Kaggle), and Open-source photos. The best results were obtained when using the model DenseNet121 with an accuracy of 93.1%.

Wang et al [26] initially chose the YOLOv4 object detection architecture as the neural network's backbone. Due to the large number of parameters, heavy computational load, and significant memory requirements, this model is not suitable for implementation on embedded development kits with limited computational power. As a result, they replaced the YOLOv4 model's backbone with a MobileNetV3 model to create an initial lightweight YOLO + MobileNet model and reduce the number of parameters in the model as well as the computational load. The model was then further compressed by removing redundant parts of the proposed network structure. Finally, using knowledge distillation, they improved the detection accuracy of the compressed model and obtained the final model.

Yanık et al. [27] presented a new drone-based architecture for smoke and fire recognition tasks in low-cost forests equipped with image processing and object detection capabilities. To do so, they used a drone equipped with a Raspberry on which a lightweight deep learning model based on MobileNet is deployed. The study focuses on the issue of battery consumption in order to increase the number of flight hours of the UAV. The proposed model "ssdlite mobilenet" is tested on four variants of parameters related to the number of images in the training and testing phases on the COCO dataset.

However, most of the time, object detection systems tend to be inaccurate and inefficient for detecting potential forest fires when image quality is bad or when the fire area is relatively small. As such, an accurate and efficient object detection system is required to detect potential forest fires from images or videos immediately. In this paper, we propose an automated system that uses an object detection technique to detect potential forest fires and quickly alert the appropriate authorities. In this way, they can rapidly respond to such crises and reduce the impact of the fire on forests.

IV. Proposed Method

According to the above research works, computer vision-based methods provide improvements over traditional methods. This is where our research comes in, we propose an automated system to detect fire in forests by using drones. The drone is equipped with a high-resolution camera, which films the area to be inspected for fire hazards. Video from the Drone's camera is transferred to an object detection system. The object detection system uses different techniques to automatically detect potential forest fires from a video. The coordinates of the detected fires are then transferred to the Geographic Information System (GIS). The GIS then creates a map of the detected fires and sends the map to the fire department. The map shows the location of the fire and the drone's current location. The fire department then sends a fire patrol to this location. The drone continues to film the area and continues to send videos to our system. The object detection system keeps following the fire and sending the updated fire location to the GIS. The GIS then updates the map and sends it to the fire department. The Fire patrol is then able to follow the fire and put it out (see Fig. 2).

The methodology proposed consists of five key steps, outlined in Fig. 3.

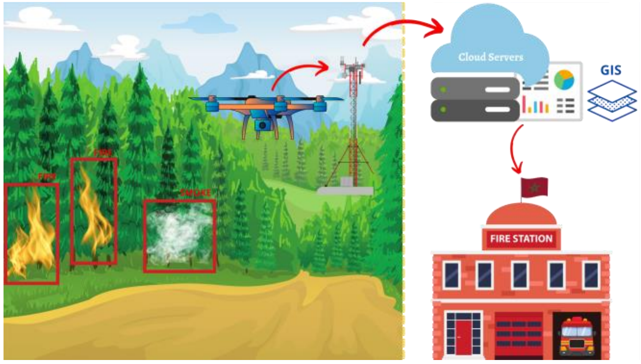


Fig. 2. Proposed architecture.

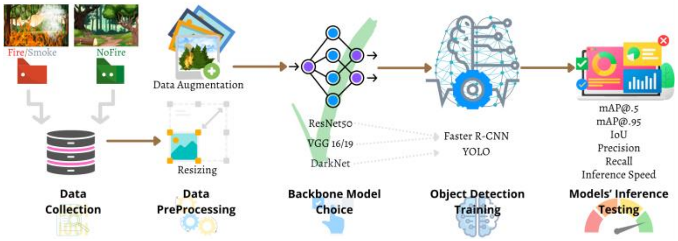


Fig. 3. Proposed method.

A. Dataset Collection

The first step in our proposed method is to collect a dataset of images containing forest fires. Our dataset (after data augmentation) reaches a total of 4236 images with the labels Fire and Smoke. These photos were taken with both ground-level cameras and aerial drones. The ground cameras were used to capture detailed images of forest fires in real-time, providing accurate representations of the fires (some images were shot by us on the campus, see Fig. 4). Aerial drone images, on the other hand, were used to provide a broader perspective by capturing larger areas of an entire forest, allowing us to monitor the extent and size of the fire's affected area. Additionally, photos were gathered from publicly accessible datasets such as online image libraries and websites. These images, which depict real-world scenarios of forest fires, were used to further expand the dataset and provide a more diverse dataset for an effective learning.

To facilitate the labeling process, we leveraged the open-source software Label-Studio, which required significant time and effort from our project team. To streamline the task, we divided the dataset among all members, enabling us to complete the process efficiently.

It is worth noting that we have taken the necessary precautions to ensure that the images are diverse in terms of the various types of fires, fire intensities, and environments in which they occurred. This was done to ensure that the model could detect fires accurately in a variety of conditions, thus improving its performance and generalization ability.

Fig. 4. Images of the man-made supervised fires.

B. Data Preprocessing

The next step in our proposed method is to preprocess and augment the collected dataset. Preprocessing includes cropping and resizing the images to the required size (640x640), as well as converting the images to a standard format such as JPG or PNG (JPG in our case). Data preprocessing plays a crucial role in object detection and can determine the success or failure of an object detection system. By resizing the images to a standard size, we can ensure that all the data is of the same size, making it easier to work with and compare.

Augmentation involves applying various distortions and transformations to the images, such as rotations, horizontal flips, vertical flips, and random crops, to increase the variety of the dataset and make it more robust. Data preprocessing is performed to increase the available training data and improve the ability of the object detection system to recognize objects from various perspectives. Resizing and augmenting the data help to increase the chances of success in object detection.

C. Backbone Model Choice

When selecting a backbone model for object detection, we have several options to consider, including VGG-16, VGG-19, and ResNet50, which have been found to be effective in our previous research [4]. It is important to carefully evaluate the strengths and weaknesses of each model in our object detection system to determine which one performs best on our specific dataset. To do this, we can benchmark each model and compare its performance for the Faster R-CNN model. DarkNet is specifically used as the backbone model for YOLOv6, v7, and v8.

ResNet (Residual Network) is a deep learning model introduced by Microsoft Research in 2015 [28]. It is known for its ability to train very deep neural networks with hundreds or even thousands of layers, using a technique called skip connections or shortcut connections. These connections allow the model to learn residual functions, or the difference between the input and the desired output, rather than trying to learn the entire mapping from scratch. This helps to alleviate the vanishing gradient problem and enables ResNet to achieve very good performance on a variety of tasks. In this work, we will utilize three backbone combinations with the faster R-CNN architecture, including:

- C4 feature extractor: a type of feature extractor used to extract relevant information from the feature maps produced by a convolutional neural network for object detection.
- Feature Pyramid Network (FPN): a type of neural network architecture used for object detection that combines high-resolution and semantically strong features to produce a multi-scale feature representation.
- 5 levels of down-sampling (DC5): a design choice in a feature extractor where the image is down-sampled five times to produce a lower resolution version while preserving important features, making object detection easier and faster to process.

VGG was developed by the Visual Geometry Group at the University of Oxford [29], it is a convolutional neural network architecture known for its simplicity and good performance on image classification tasks. It consists of a series of convolutional and max pooling layers, followed by a few fully-connected layers. VGG-16 and VGG-19 are two variations of the VGG model that differ in the number of layers and the number of parameters.

DarkNet is a neural network framework developed by Joseph Redmon [17]. It is the basis for the YOLO object detection algorithm, which is known for its speed and real-time performance. DarkNet consists of a series of convolutional and max pooling layers, followed by multiple fully-connected layers. It is designed to be simple and easy to extend, making it a popular choice for researchers and practitioners working on object detection and other computer vision tasks [30].

D. Object Detection Training

The fourth step in our proposed method involves fine-tuning and training our object detection models on the preprocessed and augmented dataset. The dataset is split into train (70%), validation (20%), and test (10%) sets. We use a variety of models including YOLOv6, v7, and v8, and faster R-CNN (RS50 and VGG16/19). The backbone model serves as the foundation for our models. We have selected faster R-CNN as it is a more efficient and precise object detection algorithm compared to R-CNN and Fast R-CNN, and it has the capability to process complex images and learn high-level features with fast inference speed [16]. YOLO, particularly its newer versions, is also well-known for its speed and accurate results.

We train our models using labeled data, which typically consists of bounding boxes around objects in the image and information about the class of objects contained within the box. The input data for faster R-CNN is in the form of TensorFlow record (TFRecord) files, while YOLO uses TXT annotations and YAML config files. The goal of this step is to create a highly accurate and reliable model for detecting forest fires.

E. Models Evaluation

Finally, we evaluate the models' performance on the collected dataset by using the test set (10%) to assess the mean average precision and inference speed of each model. The testing phase is crucial as it enables us to measure the models' performance on previously unseen data and helps us to determine the overall efficiency of the models in detecting different types of forest fires.

V. Results and Discussions

In this section, we describe the results and discussion of our proposed method for object detection using a drone-mounted camera. To assess the accuracy of the proposed system, we have evaluated the results of object detection against a dataset of real forest fires. Moreover, the system was tested on fake and real forest fires and smokes to study the robustness of the proposed system.

A. Hardware Characteristics

The experimental setup used in this work consisted of a drone (DJI Mavic Air) equipped with a high-resolution camera, a computer, and the proposed object detection system. To run the proposed object detection model, we used a high-performance computing machine with the following hardware specifications:

- Two Intel Gold 6148 (2.4GHz/20-core) processors.
- Two NVIDIA Tesla V100 graphics cards, each having 32GB of RAM.

B. Evaluation Metrics

There are a variety of different metrics that can be used to evaluate the performance of an object detection algorithm, including:

1) Average Precision (AP): It is a fairly straightforward metric that simply measures the average precision of the detector across all classes. This is a good metric to get a general idea of how well the detector is performing. However, it doesn't give any insight into how well the detector is performing in specific classes. AP is calculated by first computing the precision-recall curve for a given set of detections, then taking the average of the precision values at regularly spaced recall levels. Given a set of detections [31], the formula for average precision (AP) is:

$$AP = \sum_{k=0}^{k=n-1}[Recalls(k) - Recalls(k+1) * Precisions(k)] \quad (1)$$

Where Recalls(n)=0, Precisions(n)=1, and n=Number of thresholds.

2) Mean Average Precision (mAP): It is a more sophisticated metric that takes into account the precision of the detector in each class. This is a good metric to get a more detailed picture of how well the detector is performing. However, it can be more difficult to interpret than AP. The mAP metric is usually reported at several confidence thresholds (e.g., 0.5, 0.95). The formula for mean average precision (mAP) is:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} APi \quad (2)$$

Where APi is the Average Precision of class i and N is the number of classes.

3) Intersection over Union (IoU): It is a metric that measures the amount of overlap between the detected object and the ground truth object. This is a good metric to make sure that the detected object is a good match for the ground truth object. However, it can be more difficult to be interpreted than AP or mAP. Given two rectangles, with coordinates (x1, y1, x2, y2) and (x3, y3, x4, y4), the formula for Intersection over Union (IoU) is:

$$IoU = \frac{(Area\ of\ Intersection)}{(Area\ of\ Union)} \quad (3)$$

$$Area\ of\ Intersection = (min(x2, x4) - max(x1, x3)) * (min(y2, y4) - max(y1, y3)) \quad (4)$$

$$Area\ of\ Union = (x2 - x1) * (y2 - y1) + (x4 - x3) * (y4 - y3) - Area\ of\ Intersection \quad (5)$$

In general, AP is the primary metric used to measure the performance of an object detection model. However, mAP is also commonly used as it provides a more thorough overview of the model's performance. IoU is used as a complementary metric to provide insights into how well the model is doing in terms of localization.

C. Evaluating the Results

In this study, several object detection models were evaluated for their performance in detecting forest fires and smoke. The models included Faster R-CNN (with different backbone networks) and YOLO models (v6, v7, and v8 with different architectures and computational requirements) trained on a dataset of forest fire and smoke images; we trained the Faster RCNN models over 500000 iterations and 1000 epochs for the YOLO models. The models were trained and evaluated using several metrics, including mAP at 0.5 and 0.95 IoU thresholds, recall, and precision. The inference time (s/image) was also measured on two Nvidia Graphics cards V100.

The YOLO models v6, v7 and v8 although they appeared successively in time, they are not necessarily progressively improved versions, and the meaning of (n) is nano, (s) is small, (l) is large model, and (x) is extra-large model (in the case of YOLOv7; this model does not provide a large version). The nano (n), small (s), and large (l) variations of the YOLO models have different numbers of layers and parameters, which can affect their accuracy and inference time.

Table I summarizes the achieved results for the implemented models on the testing set. The Faster R-CNN models with the ResNet-50 (RS50) and Feature Pyramid Network (FPN) backbones showed the best performance, with the Faster R-CNN (RS50) FPN achieving a mAP@0.5 of 90.57% and a mAP@0.95 of 80.34%. This model also had the lowest inference time among the Faster R-CNN models, with an average of 0.0281 seconds per image. On the other hand, the YOLO models showed slightly lower performance compared to the Faster R-CNN models, with YOLOv8n achieving a mAP@0.5 of 89.45% and a mAP@0.95 of 79.28%. However, the YOLO models had a much lower inference time, with YOLOv8n having an average of 0.0011 seconds per image.

Fig. 5 to 10 shows the performance of Faster R-CNN (RS50) FPN and YOLOv8n over iterations on the validation set. These figures show that both models achieved a relatively stable performance over iterations, with Faster R-CNN (RS50) FPN achieving a higher mAP@0.5 and mAP@0.95, and YOLOv8n having a lower loss.

However, the YOLO models performed well in terms of inference time, with YOLOv6n, YOLOv8n, YOLOv8s, and YOLOv8l having an inference time of fewer than 0.0011 seconds per image. This makes YOLO a good choice for real-time applications such as drone data, where fast processing speed is essential. The Fast-RCNN (RS50) C4, Fast-RCNN (RS50) DC5, and Fast-RCNN (VGG19) models also showed good results, however, the processing speed was higher compared to YOLO models.

In sum, the choice between YOLO and Faster RCNN models for the task of forest fire detection would depend on the desired trade-off between accuracy and processing speed. For applications that prioritize high accuracy, the Faster R-CNN models, particularly the Faster R-CNN (RS50) FPN model would be the best choice. On the other hand, for real-time applications that require fast processing speeds, the YOLO models, particularly YOLOv8n, would be the best option (Fig. 11).

In conclusion, the choice between YOLO and Faster RCNN models for the task of forest fire detection would depend on the desired trade-off between accuracy and processing speed. For applications that prioritize high accuracy, the Faster RCNN models, particularly the Faster RCNN (RS50) FPN model, would be the best choice. On the other hand, for real-time applications that require fast processing speeds, the YOLO models, particularly YOLOv8n, would be the best option (Fig. 10).

TABLE I. ACHIEVED RESULTS FOR THE IMPLEMENTED MODELS (ON THE TESTING SET)

Model Name	**mAP@0.5 %**	**mAP@0.95 %**	**IoU %**	**Recall %**	**Precision %**	**Inference time (s/image)**
Faster R-CNN (RS50) C4	89.32	79.12	89.36	90.31	89.17	~0.0553
Faster R-CNN (RS50) DC5	89.16	78.96	88.15	89.74	88.96	~0.1374
Faster R-CNN (RS50) FPN	90.57	80.34	91.02	90.83	90.61	~0.0281
Faster R-CNN (VGG19)	89.75	79.65	90.44	89.74	89.41	~0.0753
Faster R-CNN (VGG16)	89.62	79.52	89.23	89.74	89.21	~0.0675
YOLOv6n	89.12	78.96	88.06	89.04	88.82	~0.0009
YOLOv7	89.29	79.12	89.17	89.24	89.02	~0.0027
YOLOv8n	89.45	79.28	89.36	89.61	89.44	~0.0011
YOLOv6s	88.98	78.82	88.03	88.57	88.42	~0.0022
YOLOv8s	89.31	79.16	89.25	89.40	89.24	~0.0015
YOLOv6l	88.84	78.68	88.06	88.19	88.01	~0.0086
YOLOv7x	89.01	78.85	89.12	87.79	87.62	~0.0051
YOLOv8l	89.17	79.01	89.24	89.19	89.02	~0.0025

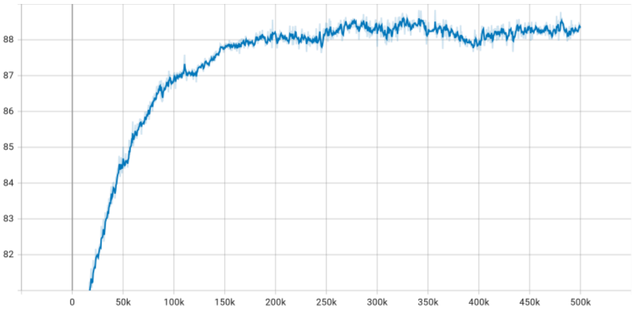


Fig. 5. Achieved mAP@0.5 over iterations for faster R-CNN (RS50) FPN (on the validation set).

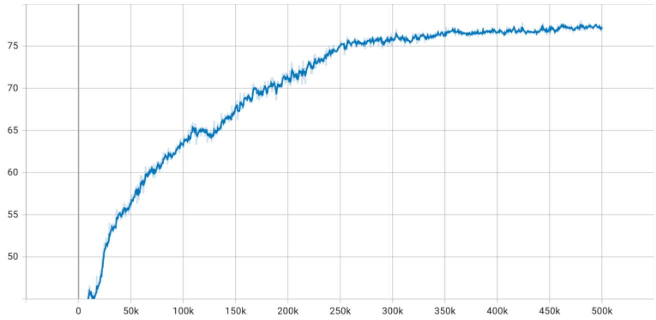


Fig. 6. Achieved mAP@0.95 over iterations for faster R-CNN (RS50) FPN (on the validation set).

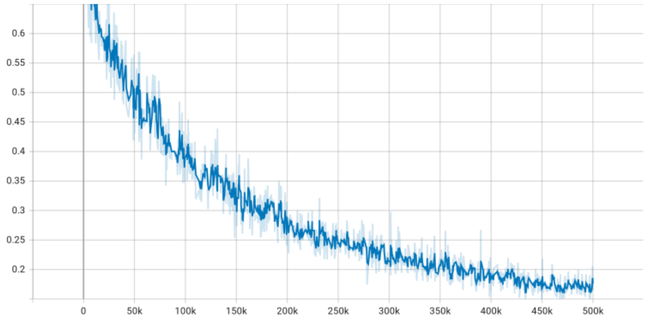


Fig. 7. Achieved loss over iterations for faster R-CNN (RS50) FPN (on the validation set).

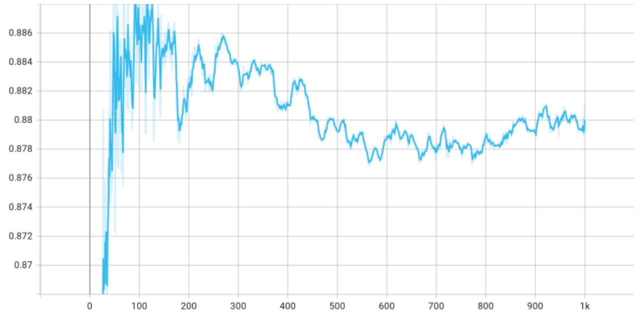


Fig. 8. Achieved mAP@0.5 over iterations for YOLOv8n (on the validation set).

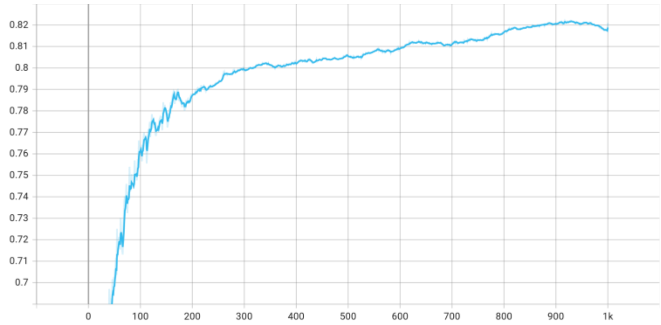


Fig. 9. Achieved mAP@0.95 over iterations for YOLOv8n (on the validation set).

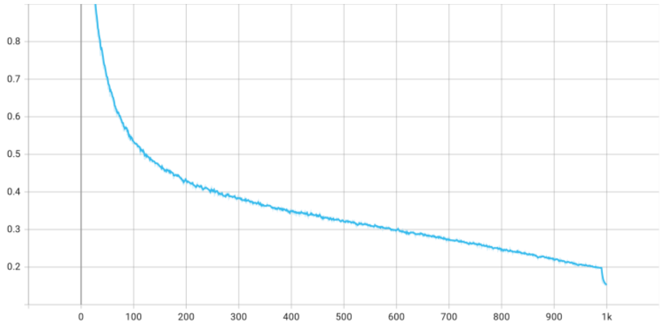


Fig. 10. Achieved loss over iterations for YOLOv8n (on the validation set).

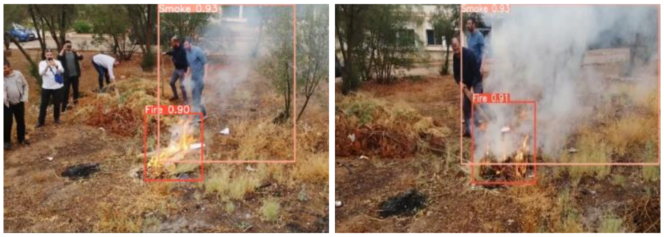


Fig. 11. Forest fire and smoke detection by drone - examples using YOLOv8n.

VI. Conclusion

Forests are of utmost importance to maintain the balance of the ecosystem and provide various ecological, social and economic benefits. However, the increasing frequency and severity of forest fires pose a significant threat to the sustainability of forests and their functions, making early detection and prompt actions critical for limiting the damages. The use of drones fitted with sensors and cameras presents a cost-effective and efficient solution for detecting fires in real-time. The proposed method consists of four major steps, including video recording, object detection, GIS mapping, and fire department notification, to provide an efficient and cost-effective solution for real-time monitoring and early fire detection. This study conducts an extensive evaluation of the recent advancements in deep learning object detection techniques, including YOLO, Faster R-CNN, and their variants, with a specific emphasis on their suitability for forest fire monitoring. Based on the experimental findings, these techniques exhibit positive outcomes in several metrics, thereby presenting a promising tool for detecting and monitoring fires. To select the appropriate model for detecting forest fires and smoke based on drone images, it is important to find a balance between accuracy and processing speed. For higher accuracy, the Faster RCNN model is recommended, whereas for real-time applications that prioritize speed, the YOLO model, particularly the YOLOv8n version, is the better choice with a mAP@0.5 of 89.45%, a mAP@0.95 of 79.28% and an inference time of almost 0.0011 seconds per image.

As part of our upcoming tasks, we are currently exploring the utilization of thermal images captured by UAVs. Additionally, we are examining the individual contributions of each RGB layer during model training to effectively decrease the overall number of parameters.

ACKNOWLEDGMENT

This work is supported by the Mohammed First University under the PARA1 Program (Low-cost, real-time Forest Fire Detection System based on Wireless Sensor Networks - SDF-RCSF). And the computational resources of HPC-MARWAN are provided by the National Center for Scientific and Technical Research (CNRST), Rabat, Morocco.

REFERENCES

[1] M. Grari, I. Idrissi, M. Boukabous, O. Moussaoui, M. Azizi, and M. Moussaoui, “Early wildfire detection using machine learning model deployed in the fog/edge layers of IoT,” Indones. J. Electr. Eng. Comput. Sci., vol. 27, no. 2, pp. 1062–1073, Aug. 2022, doi: 10.11591/IJEECS.V27.I2.PP1062-1073.

[2] M. Grari et al., “Using IoT and ML for Forest Fire Detection, Monitoring, and Prediction: a Literature Review,” J. Theor. Appl. Inf. Technol., vol. 100, pp. 5445–5461, 2022.

[3] M. Yandouzi et al., “Review on forest fires detection and prediction using deep learning and drones,” J. Theor. Appl. Inf. Technol., vol. 100, no. 12, pp. 4565–4576, 2022.

[4] M. Yandouzi et al., “Forest Fires Detection using Deep Transfer Learning,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 8, pp. 268–275, Oct. 2022, doi: 10.14569/IJACSA.2022.0130832.

[5] M. Berrahal and M. Azizi, “Augmented Binary Multi-Labeled CNN for Practical Facial Attribute Classification,” Indones. J. Electr. Eng. Comput. Sci., vol. 23, no. 2, pp. 973–979, Aug. 2021.

[6] A. Kherraki and R. El Ouazzani, “Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring,” IAES Int. J. Artif. Intell., vol. 11, no. 1, pp. 110–120, Mar. 2022.

[7] I. Idrissi, M. Azizi, and O. Moussaoui, “A Stratified IoT Deep Learning based Intrusion Detection System,” in 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Mar. 2022, pp. 1–8, doi: 10.1109/IRASET52964.2022.9738045.

[8] M. Boukabous and M. Azizi, “Review of Learning-Based Techniques of Sentiment Analysis for Security Purposes,” in Innovations in Smart Cities Applications Volume 4, Springer, Cham, 2021, pp. 96–109.

[9] Y. Hammoudi, I. Idrissi, M. Boukabous, Y. Zerguit, and H. Bouali, “Review on maintenance of photovoltaic systems based on deep learning and internet of things,” Indones. J. Electr. Eng. Comput. Sci., vol. 26, no. 2, May 2022.

[10] M. Boukabous and M. Azizi, “Crime prediction using a hybrid sentiment analysis approach based on the bidirectional encoder representations from transformers,” Indones. J. Electr. Eng. Comput. Sci., vol. 25, no. 2, pp. 1131–1139, Feb. 2022, doi: 10.11591/IJEECS.V25.I2.PP1131-1139.

[11] M. Berrahal and M. Azizi, “Review of DL-Based Generation Techniques of Augmented Images using Portraits Specification,” in 4th International Conference on Intelligent Computing in Data Sciences, ICDS 2020, Nov. 2020, pp. 1–8, doi: 10.1109/ICDS50568.2020.9268710.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 580–587, Nov. 2013, doi: 10.48550/arxiv.1311.2524.

[14] R. Girshick, “Fast R-CNN,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, vol. 2015 Inter, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[15] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2015, doi: 10.48550/arxiv.1506.01497.

[16] M. Boukabous and M. Azizi, “Image and video-based crime prediction using object detection and deep learning,” Bull. Electr. Eng. Informatics, vol. 12, no. 3, pp. 1630–1638, Jun. 2023, doi: 10.11591/EEI.V12I3.5157.

[17] J. Redmon and A. Farhadi, “ YOLO: Real-Time Object Detection,” 2018. https://pjreddie.com/darknet/yolo/ (accessed Jan. 30, 2023).

[18] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Apr. 2018, Accessed: Mar. 22, 2023. [Online]. Available: https://arxiv.org/abs/1804.02767v1.

[19] C. Li et al., “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” Sep. 2022, doi: 10.48550/ARXIV.2209.02976.

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” arXiv Prepr. arXiv2207.02696, 2022.

[21] G. Jocher, A. Chaurasia, and J. Qiu, “YOLOv8,” 2023. https://github.com/ultralytics/ultralytics (accessed Jan. 30, 2023).

[22] J. Zheng, C. Xianbin, Z. Baochang, Y. Huang, and Y. Hu, “Bi-heterogeneous Convolutional Neural Network for UAV-based dynamic scene classification,” ICNS 2017 - ICNS CNS/ATM Challenges UAS Integr., Aug. 2017, doi: 10.1109/ICNSURV.2017.8011932.

[23] Z. Jiao et al., “A Deep learning based forest fire detection approach using uav and yolov3,” 1st Int. Conf. Ind. Artif. Intell. IAI 2019, Jul. 2019, doi: 10.1109/ICIAI.2019.8850815.

[24] Z. Jiao et al., “A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection,” Proc. 32nd Chinese Control Decis. Conf. CCDC 2020, pp. 4963–4967, Aug. 2020, doi: 10.1109/CCDC49329.2020.9163816.

[25] G. V. S. Lohit and D. Bisht, “Reforestation Using Drones and Deep Learning Techniques,” 2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021, pp. 847–852, Mar. 2021, doi: 10.1109/ICACCS51430.2021.9442053.

[26] S. Wang, J. Zhao, N. Ta, X. Zhao, M. Xiao, and H. Wei, “A real-time deep learning forest fire monitoring algorithm based on an improved Pruned + KD model,” J. Real-Time Image Process. 2021 186, vol. 18, no. 6, pp. 2319–2329, May 2021, doi: 10.1007/S11554-021-01124-9.

[27] A. Yanık, M. Yanık, M. S. Güzel, and G. E. Bostancı, “Machine Learning–Based Early Fire Detection System Using a Low-Cost Drone,” Adv. Sens. Image Process. IoT, pp. 1–18, Feb. 2022, doi: 10.1201/9781003221333-1.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, Dec. 2015, doi: 10.1109/CVPR.2016.90.

[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015.

[30] Redmon J Darknet: Open Source Neural Networks in C. https://pjreddie.com/darknet/. (accessed Jan. 23, 2023).

[31] Average Precision - Hasty.ai. https://hasty.ai/docs/mp-wiki/metrics/average-precision. (accessed Jan. 30, 2023).

Method for Frequent High Resolution of Optical Sensor Image Acquisition using Satellite-Based SAR Image for Disaster Mitigation

Kohei Arai, Yushin Nakaoka, Osamu Fukuda, Nobuhiko Yamaguchi, Wen Liang Yeoh, Hiroshi Okumura
Information Science Department, Saga University, Saga City, Japan

Abstract—**Method for frequent high resolution of optical sensor imagery data acquisition from satellite-based SAR (Synthetic Aperture Radar) image for disaster mitigation is proposed. The proposed method is based on Generative Adversarial Network: GAN-based super resolution and conversion method from a SAR imagery data to the corresponding optical sensor imagery data in order to increase observation frequency. Through experiments, it is found that it is possible to convert SAR imagery data to the corresponding optical sensor imagery data and also found that the spatial resolution of SAR imagery data is improved remarkably. Thus, initial stage of disaster (small scale of disaster) can be detected with resolution enhanced optical sensor imagery data derived from the corresponding SAR imagery data which results in prevention of secondary occurrence of relatively large scale of disaster. It is also found that 2.5 m of spatial resolution of optical sensor imagery data can be acquired every 2.5 days in the case that only Sentinel-1/SAR and Sentinel-2/MSI (Multi Spectral Imager) are used, for instance.**

Keywords—Frequent observation; Synthetic Aperture Radar: SAR; super resolution; Generative Adversarial Network: GAN; GAN-based conversion of images

I. Introduction

The purpose of this research is to detect relatively small-scale weather caused disaster (an initial stage of disaster) such as landslide disaster, slope failure, flood, road submergence, sediment disaster, etc. for mitigation of a secondary occurrence of relatively large scale of disaster. For this purpose, observation frequency and spatial resolution are key issues.

In case of weather caused disaster, optical sensor data cannot be acquired due to weather conditions, rainy and cloudy nevertheless it is easy to detect disaster areas in the optical sensor images. On the other hand, SAR imagery data can be acquired in such weather conditions it is not so easy to detect disaster areas in the SAR imagery data though. Also, SAR imagery data can be acquired in nighttime. Therefore, observation frequency of SAR sensors is much higher than that of optical sensors. Furthermore, there are a variety of spatial resolutions of optical and SAR sensor data. High spatial resolution of sensor data is required for an initial stage of the disaster detection.

The method proposed here allows enhancement of the acquired SAR and optical sensor imageries using the well-known GAN[1] [1]-based Single Image Super Resolution: SISR[2] [2] and Very Deep Super Resolution: VDSR[3] [3] as well as improved super resolution [4] and also conversion of a SAR imagery data to the corresponding optical sensor imagery data [5], [6]. By combining between the optical sensor imagery data derived from the SAR imagery data and the actual optical sensor data, observation frequency is increased remarkably. Thus, frequent high resolution of optical sensor imagery data can be obtained from the SAR imagery data. This is a basic idea of the proposed method.

The proposed method is validated with satellite-based SAR and optical sensor imagery data derived from the Sentinel-1/SAR and Sentinel-2/MSI sensors. There is well reported research works on the conversion of Sentinel-1/SAR imagery data to optical sensor of imagery data through learning processes based on GAN with training samples of Sentinel-1/SAR and Sentinel-2/ MSI imagery data [7]. Before the conversion, super resolution is applied to Sentinel-1/SAR and Sentinel-2/MSI imagery data then learning processes of GAN-based conversion of SAR to optical sensor imagery data. Therefore, 10 m of spatial resolutions of Sentinel-1/SAR and Sentinel-2/MSI are improved to 2.5 m of spatial resolution with the spatial resolution enhancing factor of 4.

Meanwhile, SAR imagery data is geometrically affected by the well-known fore shortening, layover, and shadowing. Therefore, some treatments, orthographic transformation and terrain correction are required to avoid such influences. In this paper, effects of these treatments are investigated.

The following section describes the related research works followed by the proposed method. Then, experiments conducted are described followed by conclusion with some discussions.

II. Related Research Works

As for the disaster mitigation from space related previous research works, there are the following published papers,

Visualization of 5D assimilation data for meteorological forecasting and its related disaster mitigation utilizing VIS5D[4]

[1] https://ja.wikipedia.org/wiki/%E6%95%B5%E5%AF%BE%E7%9A%84%E7%94%9F%E6%88%90%E3%83%8D%E3%83%83%E3%83%88%E3%83%AF%E3%83%BC%E3%82%AF

[2] https://github.com/ServiceNow/SISR

[3] https://github.com/twtygqyy/pytorch-vdsr

[4] https://www.ssec.wisc.edu/~billh/vis5d.html

of software tool is demonstrated for disaster mitigation with remote sensing satellite imagery data [8]. Flooding and oil spill disaster relief using Sentinel of remote sensing satellite data is well reported [9]. Convolutional neural network considering physical processes and its application to disaster detection is proposed and validated with remote sensing satellite imagery data [10].

Present Status for Disaster Observation Systems Working Group is reported for research working group of Japan-US Space Research Cooperation [11]. Four Dimensional GIS and Its Application to Disaster Monitoring with Satellite Remote Sensing Data is proposed [12]. An Expectation to Remote Sensing for Disaster Management is announced for the United Nation and Japan-US Science/Technology and Space Application Program [13].

The Conference on GIS and Application of Remote Sensing to Disaster Management Four Dimensional GIS and Its Application to Disaster Monitoring with Satellite Remote Sensing Data is reported [14] together with the Current Status on Disaster Monitoring with Satellites in Japan [15]. Opening Remarks of Satellite Based Disaster Management is made for the Disaster Management Working Group [16].

Disaster related activities are introduced for Japan US Science and Technology as well as Asian Disaster Reducing Center [17]. Internet GIS and Disaster Information Clearing House is proposed [18]. Opening address of the disaster management symposium, United Nations Center for Regional Development is made [19] together with the Virtual center for disaster management for United Nations Center for Regional Development as well [20].

Joint Research on Disaster Management is also proposed and accepted in the United Nations, Center for Regional Development, UNCRD Headquarter [21]. Visualization of disaster information derived from Earth observation data is investigated [22]. Disaster monitoring with ASTER[5] onboard Terra satellite is proposed and demonstrated its usefulness [23].

ICT technology is for disaster mitigation in particular for Tsunami warning [24]. On the other hand, cellular automata-based approach for prediction of hot mudflow disaster area is investigated [25]. Simulation of hot mudflow disaster with cellular automata and verification with satellite imagery data is conducted [26].

Backup communication routing through Internet Satellite, WINDS[6], for transmission of disaster relief data is attempted [27]. Meanwhile, deceleration in the micro traffic model and its application to simulation for evacuation from disaster area is proposed and validated [28] together with cellular automata approach for disaster propagation prediction and required data system in GIS representations [29].

Cell based GIS as Cellular Automata for disaster spreading prediction and required data systems is proposed [30]. On the other hand, disaster relief with satellite based Synthetic Aperture Radar data is attempted [31]. Sentinel 1A SAR Data Analysis for Disaster Mitigation in Kyushu is well reported and demonstrated its usefulness [32]. Convolutional neural network considering physical processes and its application to disaster detection is proposed and validated [33].

Flood damage area detection method by means of coherency derived from interferometric SAR analysis with Sentinel-1A SAR is proposed [34]. Change detection method with multi-temporal satellite images based on wavelet decomposition and tiling is proposed in particular for disaster mitigations [35]. A comparative study of flooding area detection with SAR images based on thresholding and difference images acquired before and after the flooding is conducted and evaluated its usefulness [36].

III. Proposed Method

The proposed method uses GAN-based super resolution and conversion method from a SAR imagery data to the corresponding optical sensor imagery data in order to increase observation frequency.

The method proposed here allows enhancement of the acquired SAR and optical sensor imageries using the well-known GAN-based SISR and VDSR as well as improved super resolution and also conversion of a SAR imagery data to the corresponding optical sensor imagery data. By combining between the optical sensor imagery data derived from the SAR imagery data and the actual optical sensor data, observation frequency is increased remarkably. Thus, frequent high resolution of optical sensor imagery data can be obtained from the SAR imagery data. This is a basic idea of the proposed method.

On the other hand, VDSR is a convolutional neural network architecture designed for performing single image super-resolution processing. A VDSR network learns the mapping between low-resolution and high-resolution images. This mapping is possible because the low-resolution and high-resolution images have similar image content, differing mainly in fine high-frequency content.

VDSR uses residual learning method. This is what trains the network to estimate the residual image. In the context of super-resolution processing, the residual image is the difference between a high-resolution reference image and a low-resolution image upscaled using bicubic interpolation to match the size of the reference image. A residual image contains information about the detailed high-frequency content of the image.

The VDSR network detects the residual image from the luminance of the color image. The luminance channel Y of an image is the brightness of each pixel as a linear combination of red, green, and blue pixel values. On the other hand, the two chrominance channels Cb and Cr of the image represent chrominance information in different linear combinations of red, green and blue pixel values. VDSR is trained using only the luma channel. This is because human perception is more sensitive to changes in brightness than to changes in color. Train the VDSR network to estimate the residual image. Then you can reconstruct the high-resolution image by adding the estimated residual image to the up-sampled low-resolution image and converting the image back to the RGB color space.

[5] https://asterweb.jpl.nasa.gov/eos.asp

[6]https://ja.wikipedia.org/wiki/%E3%81%8D%E3%81%9A%E3%81%AA_(%E4%BA%BA%E5%B7%A5%E8%A1%9B%E6%98%9F)

Magnification is relative to the size of the low-resolution image of the size of the reference image. Low-resolution images especially lose information about the high-frequency content of the image, so SISR becomes even worse at higher magnifications. VDSR uses a large receptive field to solve this problem. This example trains a VDSR network using expansion by scaling by multiple factors. Augmentation by scaling allows the network to take advantage of the low-magnification image context, which improves results at high magnifications. Furthermore, the VDSR network can be generalized by accepting images with non-integer magnifications.

The proposed method is validated with satellite-based SAR and optical sensor imagery data derived from the Sentinel-1/SAR and Sentinel-2/MSI sensors. There are well reported research works on the conversion of Sentinel-1/SAR imagery data to optical sensor of imagery data through learning processes based on GAN with training samples of Sentinel-1/SAR and Sentinel-2/ MSI imagery data.

In accordance with study [7] (M. Schmitt1, L. H. Hughes1, X. X. Zhu, THE SEN1-2 DATASET FOR DEEP LEARNING IN SAR-OPTICAL DATA FUSION[7], DOI:10.5194/isprs-annals-IV-1-141-2018, License CC BY 4.0, Conference: ISPRS TCI Symposium 2018)[8], the following conversion can be done from Sentinel-1/SAR to Sentinel-2/MSI,

1) Training datasets: SEN1-2 is dataset composed of 282, 384 pairs of corresponding Synthetic Aperture Radar and optical image patches acquired by the Sentinel-1 and Sentinel-2 remote sensing satellites, respectively.

2) Methods of data evaluation: Semi-automatic download and image preparation using Google Earth Engine and MATLAB[9].

3) A dataset [10] containing 282, 384 pairs of images acquired from Sentinel-1 (SAR satellite) and Sentinel-2 (optical satellite) and matched with the surrounding topography using Google Earth Engine and MATLAB.

4) Image generation by “pix2pix”- by extracting 3964 pairs of SAR and optical images from the dataset.

5) SAR image of Sentinel-1 in the dataset is only VV polarization. Then the result of this analysis is conversion from VV polarization SAR image to RGB optical image: Image size is unified to 256 by 256.

6) Dataset Name: SEN1-2.

7) Developed in 2018.

8) Dataset structure: Top level: ROIs1158_spring, ROIs1868_summer, ROIs1970_fall, ROIs2017_winter - folders corresponding to 4 different random ROI distributions and four weather seasons Second level: s1_i, s2_i - Folders corresponding to scenes from which patches were cut. s1 indicates the Sentinel-1 SAR image and s2 indicates the Sentinel-2 optical image.

9) Dataset size: Number of files: 564,768 Storage: 44,742MB.

10) Description: SEN1-2 dataset includes Sentinel-1 and Sentinel-2. It contains 282,384 pairs of corresponding SAR and optical image patches acquired by two satellites respectively. The patches are distributed over the Earth's landmass and span all four weather seasons. This is reflected in the structure of the dataset. The SAR patch is provided with an 8-bit single-channel image representing the sigma note backscatter values in dB scale. The optical patch uses 8-bit color images representing bands 4, 3, and 2.

11) License Terms apply: The CC-BY dataset contains Copernicus data (2017)[11].

12) Data Size: as shown in Fig. 1.

ROI of region of interest and seasons are listed here in Fig. 1. There is huge volume of the satellite imagery data. Fig. 2 shows just 40 image pair of a small portion of SEN1-2 image dataset, Sentinel-1/SAR (greyscale) on the right and the corresponding areas of Sentinel-2/MSI (color) on the left side out of 842 datasets. These images are mostly farm areas, forested areas, rural areas, and the others and are used for training for GAN.

Install the following to use NVIDIA's GPU board in deep learning.

CUDA Toolkit,

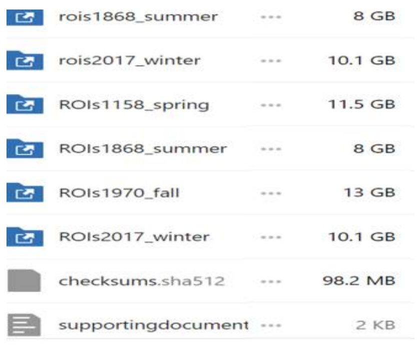
Driver software for GPU board,

Libraries such as cuDNN,

Build environment for Anaconda,

Install PyTorch,

Perform processing to convert a grayscale VV polarization SAR image into an RGB optical image. The number of epochs is 100 and the mini-batch size is 128.



rois1868_summer	···	8 GB
rois2017_winter	···	10.1 GB
ROIs1158_spring	···	11.5 GB
ROIs1868_summer	···	8 GB
ROIs1970_fall	···	13 GB
ROIs2017_winter	···	10.1 GB
checksums.sha512	···	98.2 MB
supportingdocument	···	2 KB

Fig. 1. Data size of SEN1-2 dataset.

[7] https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-1/141/2018/

[8] https://www.researchgate.net/publication/327896050_The_SEN1-2_dataset_for_deep_learning_in_SAR-optical_data_fusion?enrichId=rgreq-563078bee4732752f41a9a0fe2e0d323-XXX&enrichSource=Y292ZXJQYWdlOzMyNzg5NjA1MDtBUzo2NDQzMTg2NDU3MjcyMzJAMTUzMDYyODk0NDIzNA%3D%3D&el=1_x_3&_esc=publicationCoverPdf

[9] https://ja.wikipedia.org › wiki › MATLAB

[10] https://mediatum.ub.tum.de/1436631

[11] https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/TermConditions/TC_Sentinel_Data_31072014.pdf

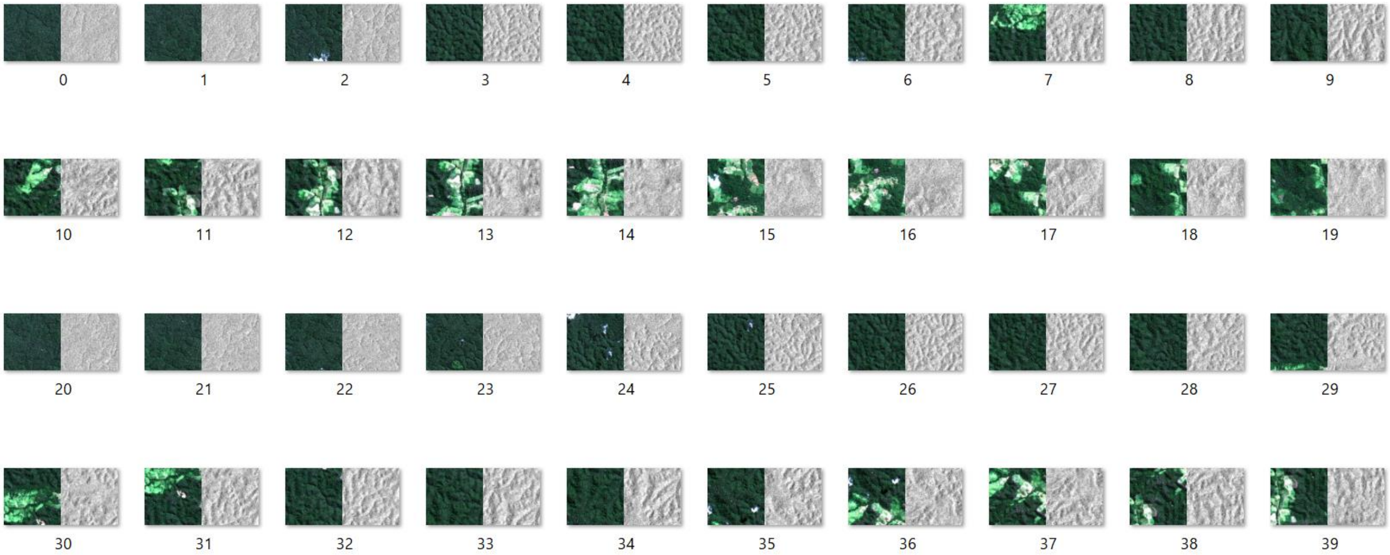


Fig. 2. Small portion of the SEN1-2 image datasets.

After the necessary datasets are read, then create a ConcatDataset class so that both SAR images and optical data can be called during training and learned. Build pix2pix Generator and Discriminator respectively. Train using loaded datasets and constructed generators and classifiers.

Before the conversion, super resolution is applied to Sentinel-1/SAR and Sentinel-2/MSI imagery data then learning processes of GAN-based conversion of SAR to optical sensor imagery data. Therefore, 10 m of spatial resolutions of Sentinel-1/SAR and Sentinel-2/MSI are improved to 2.5 m of spatial resolution with the spatial resolution enhancing factor of 4.

Without high-frequency information, the image quality of high-resolution images is limited. Furthermore, SISR is an ill-posed problem because a single low-resolution image can generate multiple candidate high-resolution images.

The residual image is the difference between the high resolution reference image and the low resolution image upscaled using bicubic interpolation to match the size of the reference image. Then CNN training the residual image is performed.

Software that rewrites only the conversion function of the image conversion software "waifu2x"[12] using Caffe and builds it for Windows using CUDA (or cuDNN) can convert faster than CPU. From the download site[13], it is possible to download the zip file of waifu2x-caffe.zip (650 MB, Source code (zip), Source code (tar.gz)).

The pix2pix is a machine learning model that generates a fake image based on the content of a certain image, whereas ordinary GANs generate fake images from random noise, in other words, image-to-image conversion. It is a model that learns the features. For learning, it is necessary to prepare a pair of an input image and a correct image.

Meanwhile, there is a learning model called pix2pixHD derived from pix2pix that learns image-to-image conversion. The differences between pix2pix and pix2pixHD include improved generators, classifiers, and loss functions. By generating a fake image with a magnification, it becomes easier to capture the local and global features of the image. Also, pix2pix used a loss function called the “L1 loss function” that learns low-frequency components, but by improving it, it became possible to generate high-resolution images.

IV. Experiment

A. Spatial Resolution Enhancement based on Super Resolution

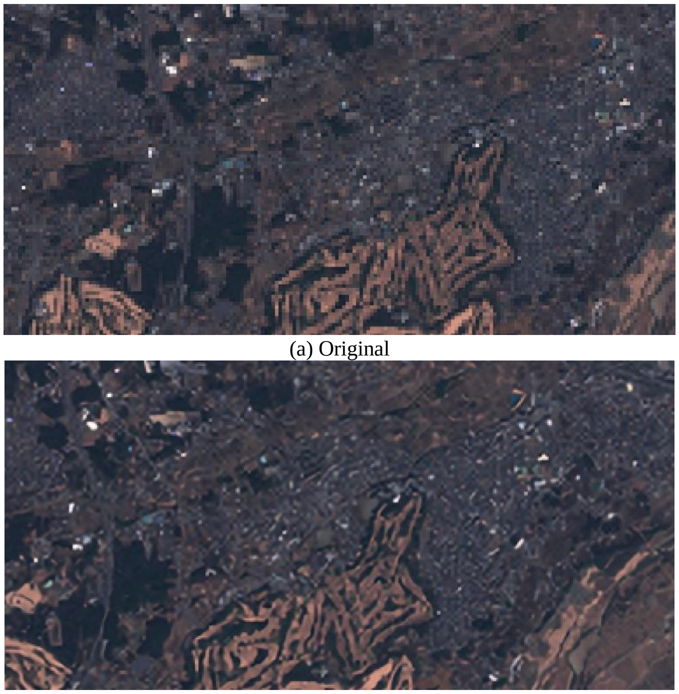
Spatial resolution of Sentinel-1/SAR and Sentinel-2/MSI images in the SEN1-2 image dataset can be enhanced with the super resolution by the enhancing factor of 4. Fig. 3 shows just an example of the enhancement based on waif2x-caffe. Also, Fig. 4 shows the frequency components of the original and the spatial resolution enhanced image. It is obvious that the enhanced image has much high frequency components rather than the original image. Therefore, the spatial resolution of Sentinel-1/SAR and Sentinel-2/MSI can be improved by the factor of 4 which results in 10 m resolution of Sentinel-1 and 2 are to be 2.5 m of spatial resolution.

B. Generation of Optical Sensor of Imagery Data from the Corresponding SAR Imagery Data

Through the learning processes with SEN1-2 of 842 datasets of the Sentinel-1/SAR and Sentinel-2/MSI imagery data, the learning models of pix2pix and pix2pixHD are created. Therefore, optical sensor imagery data is to be created with an unknown SAR image is input. Fig. 5 shows the input SAR image and the created optical sensor images with pix2pix and pix2pixHD. For a reference, original optical sensor image is shown in Fig. 5. In Fig. 5(c), the created optical sensor images by pix2pix and pix2pixHD are shown upper and lower parts, respectively. Also, four cases of image creations are shown in Fig. 5(c) with the different epochs. 670 of training

[12] https://github.com/lltcggie/waifu2x-caffe/releases
[13] https://www.aozoraapps.net/waifu2x-caffe/

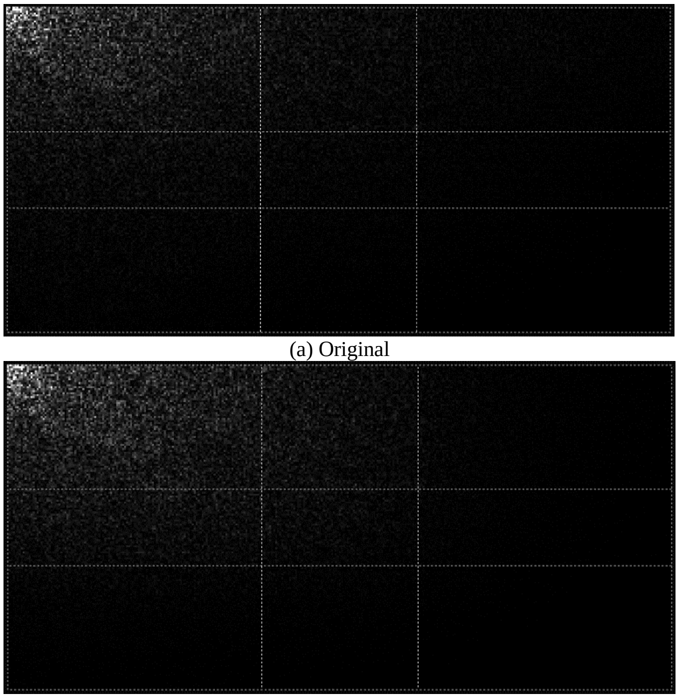
datasets are used for the learning processes of pix2pix and pix2pixHD.



(a) Original

(b) Enhanced

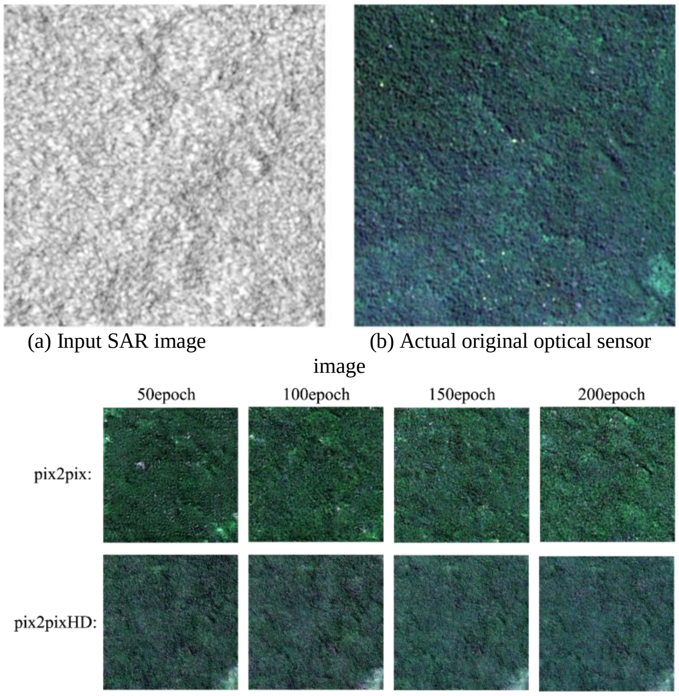
Fig. 3. Example of the spatial resolution enhancement with waif2x-caffe.



(a) Original

(b) Enhanced

Fig. 4. Frequency components of the original and the spatial resolution enhanced images.



(a) Input SAR image (b) Actual original optical sensor image

(c) Created optical sensor images

Fig. 5. Example of images of the input SAR, actual original and created optical sensor images with pix2pix and pix2pixHD.

For the object is a forest area, there is no big difference from the correct image. In addition, when pix2pix and pix2pixHD are compared, it can be seen that pix2pixHD can obtain high resolution even with a small number of epochs.

C. *Example of Disaster Area Detection with the Trained Models of pix2pix and pix2pixHD*

Sentinel-2/MSI images were generated from Sentinel1/SAR to verify the possibility of detecting a slope failure that occurred in Sakae Village, Nagano Prefecture, using a trained model with pi2pix and pix2pixHD. Information on the slope failure area is shown below. The location is latitude, longitude 36.858234 N, 138.610001E.

Fig. 6 shows a photograph of the slope failure area. In addition, other research papers have confirmed that the slope failure caused a maximum surface displacement of 13 cm from August 2020 to May 2022 in an area of about 400,000 m^2 in the southeastern part of the landslide area.

Fig. 7 shows Google map of the slope and the location of the slope failure area and the elevation of the slope. Also, Fig. 7(c) and Fig. 7(d) show Sentinel-2/MSI and Sentinel-1/SAR images, respectively.

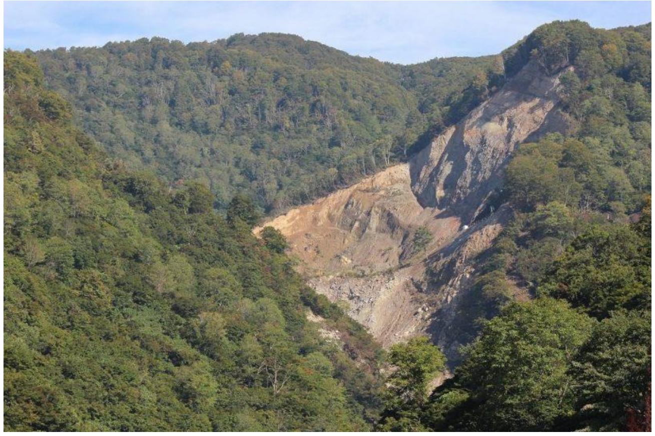
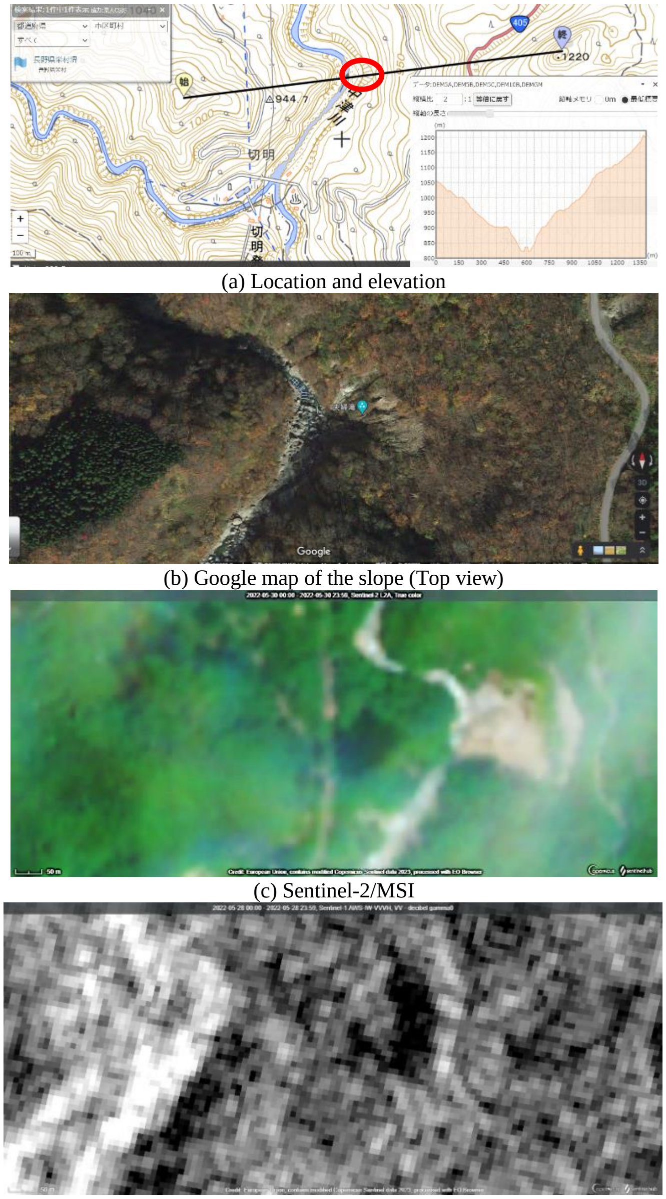


Fig. 6. Photograph of the slope failure area.



(a) Location and elevation

(b) Google map of the slope (Top view)

(c) Sentinel-2/MSI

(d) Sentinel-1/SAR

Fig. 7. Location of the slope failure area and the elevation of the slope and Sentinel-1/SAR as well as Sentinel-2/MSI images.

It is not so easy to identify the slope failure area using SAR image. Therefore, if the optical sensor image can be created from SAR image, then the slope failure area can be detected easily. The Sentinel-1/SAR of 2022/08/28 was input to the trained model with 842 image data of Sentinel-1/SAR and Sentinel-2/MSI and converted into an optical image (VV polarization) by GAN. Fig. 8 shows Sentinel-1/SAR and the created quasi-Sentinel-2/MSI images (optical sensor image).

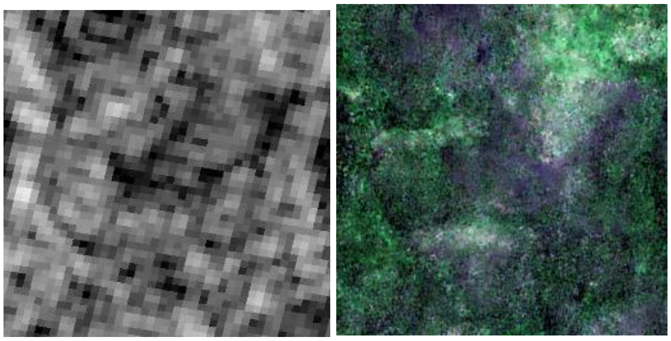
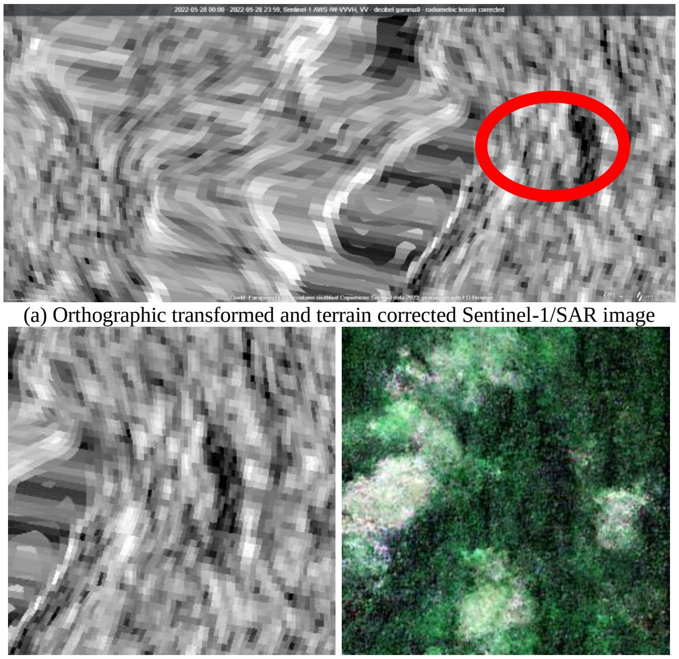


Fig. 8. Sentinel-1/SAR and the created quasi Sentinel-2/MSI images (optical sensor image) of the slope failure area.

Essentially, SAR image is acquired with oblique view and has geometric distortions due to the fore shortening, layover, and shadowing. Therefore, some treatments are required to eliminate the influences of geometric distortions. That is, orthographic transformation and terrain correction are required for Sentinl-1/SAR image. Fig. 9(a) and Fig. 9(b) show the orthographic transformed and terrain corrected Sentinel-1/SAR image. By using this corrected SAR image, optical sensor image is created. The created image with pix2pixHD is shown in Fig. 9 (c).



(a) Orthographic transformed and terrain corrected Sentinel-1/SAR image

(b) Portion of Sentinel-1/SAR (c) Created optical sensor image

Fig. 9. Orthogonal transformed and terrain corrected Sentinel-1/SAR and the created optical sensor image with pix2pixHD.

From Fig. 9, the location and the area of slope failure is getting much clear in particular, Fig. 9(c) of the created optical sensor image from the SAR image.

Sentinel-1/SAR is C-band SAR and has VH and VV polarization of receiving signal (where VH stands for that transmit with vertical polarization and received with horizontal polarization), the experiments conducted for not only VV but also VH polarizations. From these experiments, the following is concluded.

1) In comparing VV and VH, the VV backscattering intensity of the slope failure part is larger than that of VH.

2) In the comparison of optical images by pix2pixHD, the VV optical image of the slope failure part is more like an optical image than that of VH.

3) Comparing the effects of orthographic transformation and terrain correction, the correction effect of VV backscattering intensity at the slope failure part is larger than that of VH.

Sentinel-1/SAR is C-band SAR and does not have HH polarization, so it cannot be compared with QPS/SAR-2[14] data of HH polarization in X band with 70 cm of spatial resolution, but it can detect slope failure parts to the same extent as VV and the effects of orthographic transformation and terrain correction are considered to be about the same.

V. Conclusion

Method for frequent high resolution of optical sensor image acquisition from satellite-based SAR image for disaster mitigation is proposed. The proposed method uses GAN-based super resolution and conversion method from a SAR imagery data to the corresponding optical sensor imagery data in order for increasing observation frequency.

Through experiments, it is found that it is possible to convert SAR imagery data to the corresponding optical sensor imagery data and also found that the spatial resolution of SAR imagery data is improved remarkably. Thus, initial stage of disaster (small scale of disaster) can be detected with resolution enhanced optical sensor imagery data derived from the corresponding SAR imagery data which results in prevention of secondary occurrence of relatively large scale of disaster.

Furthermore, it is found:

1) In comparing VV and VH, the VV backscattering intensity of the slope failure part is larger than that of VH,

2) In the comparison of optical images by pix2pixHD, the VV optical image of the slope failure part is more like an optical image than that of VH.

3) Comparing the effects of orthographic transformation and terrain correction, the correction effect of VV backscattering intensity at the slope failure part is larger than that of VH.

4) Sentinel-1/SAR is C-band SAR and does not have HH polarization, so it cannot be compared with QPS/SAR-2 data of HH polarization in X band, but it can detect slope failure parts to the same extent as VV and the effects of orthographic transformation and terrain correction are considered to be about the same.

VI. Future Research Works

Further investigation needs to be conducted on the creation of optical sensor image with the different parameters of pix2pixHD with more training datasets of spatial resolution enhanced Sentinel-1/SAR and Sentinal-2/MSI images.

References

[1] lanJ.Goodfellow, Jean,Pouget-Abadie, Mehdi, Mirza, Bing,Xu, David, Warde-Farley, Sherjil, Ozair, Aaron, Courville, Yoshua,Bengio , "Generative Adversarial Networks" , Jun. 2014.

[2] Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. “Learning a Deep Convolutional Network for Image Super-Resolution.” In Computer Vision – ECCV 2014, 184–99. Springer International Publishing, 2014.

[3] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee. 2016. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks.” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem: 1646–54, 2016.

[4] Dong, Chao, Chen Change Loy, and Xiaoou Tang. 2016. “Accelerating the Super-Resolution Convolutional Neural Network.” In Computer Vision – ECCV 2016, 391–407. Springer International Publishing, 2016.

[5] Phillip Isola,Jun-Yan Zhu,Tinghui Zhou,Alexei A.Efros , "Image-to-Image Translation with Conditional Adversarial Networks", Nov. 2016.

[6] Ting-Chun Wang , Ming-Yu Liu , Jun-Yan Zhu , Andrew Tao , Jan Kautz , Bryan Catanzaro , NVIDIA Coorporation , UC Berkeley , “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs” , in CVPR , 2018.

[7] Michael Schmitt , Lloyd Haydn Hughes , Xiao Xiang Zhu , “The SEN1-2 Dataset for Deep Learning in SAR-Optical Data Fusion” , Jul.2018.

[8] Kohei Arai, Visualization of 5D assimilation data for meteorological forecasting and its related disaster mitigation utilizing VIS5D of software tool, International Journal of Advanced Research in Artificial Intelligence, 2, 9, 24-29, 2013.

[9] Kohei Arai, Flooding and oil spill disaster relief using Sentinel of remote sensing satellite data, International Journal of Advanced Computer Science and Applications IJACSA, 10, 12, 290-297, 2019.

[10] Kohei Arai, Convolutional neural network considering physical processes and its application to disaster detection, International Journal of Advanced Computer Science and Applications IJACSA, 10, 12, 105-111, 2019.

[11] Kohei Arai, Present Status for Disaster Observation Systems Working Group, Proceedings of the 5th Japan-US Space Research Cooperation Conference, Hawaii, Nov. 1995.

[12] Kohei Arai, Four Dimensional GIS and Its Application to Disaster Monitoring with Satellite Remote Sensing Data, Proceedings of the Conference on GIS and Application of Remote Sensing to Disaster Management, 132-137(1997).

[13] Kohei Arai, An Expectation to Remote Sensing for Disaster Management, Proceedings of the United nation and Japan-US Science/Technology and Space Application Program Joint Symposium on Disaster Management, (1997).

[14] Kohei Arai, The Conference on GIS and Application of Remote Sensing to Disaster Management Four Dimensional GIS and Its Application to Disaster Monitoring with Satellite Remote Sensing Data, Proceedings of the Conference on GIS and Application of Remote Sensing to Disaster Management, 132-137 Greenbelt, Maryland, U.S.A., 1997.

[15] Kohei Arai, The Current Status on Disaster Monitoring with Satellites in Japan, Proc. of the Committee on Earth Observation Satellites/Working Group on Information Systems and Services/Task Team 19 Meeting, Greenbelt, Maryland, U.S.A., 1997.

[14] https://i-qps.net/

[16] Kohei Arai, Opening Remarks of Satellite Based Disaster Management, Proc. of the Disaster Management Workshop in Ihilani Hotel, Hawaii USA, Invited Speech, 1998.

[17] Kohei Arai, Disaster related activities, Proceedings of the 1st JUSTSAP-ADRC Joint Symposium on Disaster Management, (1999).

[18] Kohei Arai, Internet GIS and Disaster Information Clearing House, Proceedings of the 1st JUSTSAP-ADRC Joint Symposium on Disaster Management, (1999).

[19] Kohei Arai, Opening address of the disaster management symposium, United nations Center for Regional Development Proceedings, No.34, pp.9-12, (1999).

[20] Kohei Arai, Virtual center for disaster management, United nations Center for Regional Development Proceedings, No.34, pp.33-38, (1999).

[21] Kohei Arai, Joint Research on Disaster Management, Proceedings of the United Nations, Center for Regional Development, UNCRD Headquarter, Nagoya, 7 Jan., 1999.

[22] Kohei Arai, Visualization of disaster information derived from Earth observation data, Proceedings of the Asian Disaster Reduction Center R&D Project Workshop, Aug.31, (2000).

[23] Kohei Arai, Disaster monitoring with ASTER onboard Terra satellite, Proceedings of the Japan-US Science, Technology and Space Application Program Workshop, Hiro, Hawaii, (2000).

[24] Kohei Arai, ICT technology for disaster mitigation, -Tsunami warning system-, Proceedings of the 1st International Workshop on Knowledge Cluster Systems, 2007.

[25] Kohei Arai and Achmad Basuki, Cellular automata-based approach for prediction of hot mudflow disaster area, Proceedings of the International Conference on Computational Science and Its Applications (ICCSA2010), LNCS part-II, 87-98, 2010.

[26] Kohei Arai and Achmad Basuki, Simulation of hot mudflow disaster with cellular automata and verification with satellite imagery data, Proceedings of the ISPRS WG VIII/1 TS-19, 2010.

[27] Kohei Arai, Kiyotaka Fujisaki, Hiroaki Ikemi, Masato Masuya, Terumasa Miyahara, Backup communication routing through Internet Satellite, WINDS, for transmission of disaster relief data, Proceedings of the International Symposium on WINDS Application Experiments, 2010.

[28] Tri Harsono, Kohei Arai, Deceleration in the micro traffic model and its application to simulation for evacuation from disaster area, Proceedings of the IES: Industrial Electronics Seminar, at EEPIS, 1-8, 2011.

[29] Kohei Arai, Cellular automata approach for disaster propagation prediction and required data system in GIS representations, Proceedings of the 1st ICSU/WDS Conference - Global Data for Global Science, 2011.

[30] Kohei Arai, Cell based GIS as Cellular Automata for disaster spreading prediction and required data systems, CODATA Data Science Journal, 137-141, 2012.

[31] Kohei Arai, Hiroshi Okumura, Shogo Kajiki, Disaster relief with satellite based synthetic aperture radar data, Proceedings of the SAI Future Technology Conference 2017, No.521, 1026-1029, in Vancouver, 2017.

[32] Kohei Arai, Sentinel 1A SAR Data Analysis for Disaster Mitigation in Kyushu, Kyushu Brunch of the Japanese Society on Remote Sensing, Special Lecture for Young Engineers on Remote Sensing, Nagasaki University, 2018.

[33] Kohei Arai, Convolutional neural network considering physical processes and its application to disaster detection, International Journal of Advanced Computer Science and Applications IJACSA, 10, 12, 105-111, 2019.

[34] Kohei Arai, Hiroshi Okumura, Shogo Kajiki, Flood Damage Area Detection Method by Means of Coherency Derived from Interferometric SAR Analysis with Sentinel-1A SAR, International Journal of Advanced Computer Science and Applications IJACSA, 11, 7, 88-94, 2020.

[35] Kohei Arai, Change Detection Method with Multi-temporal Satellite Images based on Wavelet Decomposition and Tiling, Journal of Advanced Computer Science and Applications, Vol. 12, No. 3, 56-61, 2021.

[36] Kohei Arai, Comparative Study of Flooding Area Detection with SAR Images based on Thresholding and Difference Images Acquired Before and After the Flooding, International Journal of Advanced Computer Science and Applications, Vol. 12, No. 11, 70-78, 2021.

AUTHOR'S PROFILE

Kohei Arai, He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 also was with National Space Development Agency of Japan from January, 1979 to March, 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He moved to Saga University as a Professor in Department of Information Science on April 1990. He was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was a councilor of Saga University for 2002 and 2003. He also was an executive councilor for the Remote Sensing Society of Japan for 2003 to 2005. He is a Science Council of Japan Special Member since 2012. He is an Adjunct Professor of University of Arizona, USA since 1998. He also is Vice Chairman of the Science Commission "A" of ICSU/COSPAR since 2008 then he is now award committee member of ICSU/COSPAR. He wrote 77 books and published 670 journal papers as well as 500 conference papers. He received 66 of awards including ICSU/COSPAR Vikram Sarabhai Medal in 2016, and Science award of Ministry of Mister of Education of Japan in 2015. He is now Editor-in-Chief of IJACSA and IJISA. http://teagis.ip.is.saga-u.ac.jp/index.html.

A Comparative Study of Twofish, Blowfish, and Advanced Encryption Standard for Secured Data Transmission

Kwame Assa-Agyei, Funminiyi Olajide
Department of Computer Science, Nottingham Trent University, Nottingham, United Kingdom

***Abstract*—Now-a-days, network security is becoming an increasingly significant and demanding research area of interest. Threats and attacks on information and Internet security are getting increasingly difficult to detect. As a result, encryption has emerged as a solution and now plays a critical role in information security systems. Many techniques are required to safeguard shared data. In this work, the encryption, decryption times, and throughput (speed) of the three most commonly used block cipher algorithms: Twofish, Blowfish, and AES were investigated using different file types. Comparison of symmetric encryption techniques of experiments on these types of algorithms uses a lot of computer resources including CPU time, memory, and battery power. Previous research has yielded diverse results in terms of time complexity, speed, space complexity, power consumption, and security. However, this research evaluated the effectiveness of each algorithm based on the following parameters: process time and speed. An application was developed for data simulation to test different file formats and for the encryption process and speed using Python 3.10.**

Keywords—Cryptography; twofish; blowfish; advanced encryption standard; throughput; data encryption; decryption

I. Introduction

Due to the increasing number of incidents in which personal data between two parties is taken by intruders, it is critical to protect data communicated over the Internet nowadays[1]. People spend so much time connected to a network that network security has become an extremely important part of data communication. These are vulnerable to security attacks such as unauthorized access to a file or alterations to its contents. One of the main reasons invaders succeed is that most of the information obtained from a system is in a form that can be read and comprehended. The solution to this dilemma is to utilize Cryptography. This is the art and science of securing information from unwanted individuals by changing it into an indiscernible form to its attackers while it is stored and transported [2]. There are numerous encryption methods that are widely available and utilized in information security. They are classified as Symmetric (private) or Asymmetric (public) Key Encryption. Only one key is needed to encrypt and decrypt data in symmetric keys encryption or secret key encryption. Asymmetric keys employ two keys: private and public keys. The public key is used to encrypt data, while the private key is used to decrypt it (e.g. RSA and ECC) [3]. A block cipher algorithm is a symmetric key cryptosystem whose security is based on sophisticated non-linear transformations and whose encryption speed is quite fast. As a result, the block cipher algorithm has evolved into a vital encryption technique that is widely utilized in applications such as secure data transfer, storage encryption, digital signing, and entity certification [4]. The primary purpose of the security mechanism is to give message privacy while also ensuring data confidentiality, integrity, and non-repetition. The primary function of network security is to enable efficient data authentication and authorization through the use of cryptographic algorithms [5]. A cryptographic algorithm is typically computationally heavy and thereby, consumes a lot of computing power such as CPU time, memory usage, and power consumption [6].

Previous research has revealed inconsistencies in the efficacy of various encryption methods. The current work analyzed symmetric (AES, Twofish, and Blowfish) cryptographic algorithms using multiple file types such as binary, text, and image files with a unique key bit size of 128. These encryption methods were compared based on three different parameters: encryption time, decryption time, and throughput. The effectiveness of each technique is demonstrated using simulation data. This study addresses the following research questions.

RQ 1: What is the performance difference between the various algorithms using a constant key bit size of 128?

RQ2: Which block cipher technique works better in the context of process time and throughput using different file types? Hence, the current study makes the following key contributions.

1) To perform an extensive evaluation of the encryption, decryption times, and speed using a unique key bit size of 128.

2) To analyze the performance using different file types.

3) To perform an extensive analysis of the performance of selected algorithms, namely: AES (Rijndael), Twofish, and Blowfish

The rest of the paper is organized as follows: Section II presents the related work. The experimental analysis and setup is presented in Section III. Section IV and V present the performance results and discussion of this research. Finally, the conclusion is drawn in Section VI.

II. Related Work

In recent years, several surveys based on various cryptographic techniques, such as the Blowfish, Twofish, and

AES algorithms, have been published. Various researchers discuss network security and cryptography challenges. This research explains and analyses earlier work in the field of data encryption to provide a broader perspective on the performance of the encryption methods.

Nema and Rizvi [7] conducted a critical analysis of various Symmetric Key Cryptographic algorithms. The objective is to identify the strengths and weaknesses of cryptographic algorithms. During the analysis, the research work observed that Blowfish was the best among all in terms of Security, Flexibility, Memory usage, and Encryption performance. Tyagi and Ganpati [8] evaluated the performance of Symmetric Key Encryption Algorithms to have a deeper understanding of the cryptography process and to perform a comparative analysis of symmetric encryption algorithms of cryptography. Blowfish algorithm runs faster than other popular symmetric key encryption algorithms: DES, 3DES, and AES. It also concluded that Blowfish gives better performance than DES, 3DES, and AES in terms of encryption time, decryption time, and throughput. 3DES has the least performance among all mentioned algorithms. The authors in [9] analysed an approach to identifying cryptographic algorithms from the cipher text. The focus of the study is to identify the performance of the cryptographic algorithm on cipher text only. The unique research work concluded that the identification rate can obtain around 90% if keys are the same for training and testing cipher texts. When they use different keys for training and testing cipher texts, it identifies AES from any one of the other four cryptographic algorithms with a high identification rate in one-to-one identification. A study conducted to assess the performance of encryption algorithms based on execution time, memory required for implementation, and throughput across two different operating systems. Based on the simulation results, AES and Salsa20 are preferable to Blowfish for plain text data encryption [6]. Singh et al., [10] presented a fair comparison between the most common four encryption algorithms namely; AES, DES, 3DES, and Blowfish in terms of security and power consumption. The simulation results showed that AES had a better performance than other common algorithms. Singh and Supriya [11] reviewed in-depth the well-known encryption methods such as RSA, DES, 3DES, and AES. They added that a variety of encryption techniques are available and that the advantages and disadvantages of each algorithm will determine which method is optimal for encrypting plain text. Each method is effective for real-time encryption. Each technique is distinctive in its own way, may be appropriate for various purposes, and has advantages and disadvantages of its own. The AES algorithm has been shown to be the most effective in terms of speed, time, throughput, and the avalanche effect, according to studies and a literature review. Ramesh and Suruliandi [12] evaluated the efficacy of some few particular symmetric algorithms in 2013. The experimental findings and input text file size led to the conclusion that the Blowfish method generates higher throughput while requiring less execution time and memory. In comparison to AES and DES, B1owfish performed around four times faster. Comparing Blowfish to AES and DES, memory usage is lower. Since AES required more computing resources than other algorithms, its performance results were subpar. Blowfish is not only the quickest encryption algorithm, but it also offers excellent security because of its large key size, making it suitable for usage in a wide range of applications, including packet encryption, random bit generation, internet-based security, and many more. Gautam et al., [13] conducted an experiment on cryptographic algorithms to analyze their performance and usage. The outcomes of the research on AES and TWOFISH are regarded as the two top candidates for achieving the aims of the study focus. These two outperform the other encryption methods in terms of speed, entropy, and optimal encoding, however, AES still has an advantage over TWOFISH due to its higher efficiency. The authors in [14] evaluated the performance of DES and Blowfish using different memory sizes. Both algorithms have high security to resist differential cryptanalysis and linear cryptanalysis attacks. They evaluated encryption function speed based on different memory sizes. The experimental results showed Blowfish is much faster than DES but as the speed increase for Blowfish, it is slower compared to DES. This was because of the needs to have more memory for sub-key and S boxes initialization. Kuma and Karthikeyan [15] conducted a comparison study on the effectiveness of the Blowfish and Rejindael (AES) algorithms for the chosen cryptographic algorithms in terms of energy consumption, changing data types like text or documents and images, power consumption, changing packet size, and changing key size. The simulation findings revealed that Blowfish surpasses AES in almost all of the test scenarios. The study found that while AES is better for image encryption, blowfish is better for text-based encryption. It is also shown that performance changes when the AES algorithm's key size is altered. Overall, the study found that AES can be used in circumstances needing a high level of security. Blowfish, however, is a performance-wise viable option. Suresh and Neema [16] explored hardware implementation of Blowfish algorithm for the secure data transmission in Internet of Things. It concluded that of all the cryptographic algorithms, the Blowfish algorithm is the best in terms of execution time, memory usage, throughput, power consumption, and security, and thus, well suited for IoT. The authors in [17] analysed the parameters of various cryptographic techniques, including AES and Blowfish, for performance, including encryption speed, CPU usage over time, and battery consumption. The outcomes showed that in terms of processing speed and throughput, the Blowfish approach performed better than the AES algorithm. The algorithm has a higher throughput while running more quickly and with less energy. According to the study, blowfish is the best option. AES, 3DES, Blowfish, and Twofish were the focus of an empirical investigation by Dibas and Sabri. The outcome demonstrated that, in terms of execution time, AES is the most effective encryption and decryption algorithm. In terms of encryption and decryption, Blowfish performed far better than 3DES. The findings obtained by Twofish were the worst. The authors found that, in terms of memory usage for encryption, AES and 3DES used less memory whereas Blowfish and Twofish used more memory and had the largest ciphertext sizes [18]. In 2020, Gosh conducted a side-by-side comparison of the three algorithms AES, Blowfish, and Twofish while taking into account various factors like speed and computation time. Conclusion: In terms of the evaluated evaluation measures, such as encryption time, decryption time, and throughput, Twofish clearly outperformed AES and Blowfish [19]. Raigoza and Jituri [20] evaluated the performance of symmetric encryption algorithms. The aim of this paper is to assess and contrast the performance of the Blowfish algorithm and the widely used Advanced Encryption Standard (AES). The AES algorithm outperformed Blowfish in terms of speed, with a difference of around 200 to 300 milliseconds. And, when the data size was altered, there were minor changes between the methods evaluated, such that the encrypted data for the AES and Blowfish algorithms tended to be roughly the same length. When the authors changed the ASCII value range, both the AES and the Blowfish algorithms increased overall execution time as the ASCII value increased, but the regression line slope for the Blowfish was more than the AES. Given the same rising ASCII values, the encrypted data from the Blowfish algorithm tended to be greater in size than the AES-encrypted data. The authors in [21] conducted an experiment to evaluate the effectiveness of the most widely used symmetric algorithms in terms of Security, Architecture, Limitations, and Efficiency and to draw attention to the shortcomings of various algorithms. AES was discovered to be the best algorithm in terms of security, efficiency, and architecture. The authors in [22] examined AES and Twofish encryption schemes. The simulation's findings were as follows: (1) for text encryption, AES is faster than Twofish, but as RAM is increased, Twofish overtakes AES. (2) AES is faster for image encryption, although Twofish performs equally well with more RAM. (3) Twofish works better for sound encryption, and its speed increases even more with more RAM. The authors in [23] evaluated the various encryption methods for secure data transmission. The study came to the conclusion that Blowfish outperformed AES, DES, and 3DES in terms of encryption and decryption times, power use, memory utilisation, latency, jitter, and security level. The authors in [24] investigated performance of selected security algorithms in cloud computing to evaluate and contrast the effectiveness of AES (Rijndael), Blowfish, and RSA. Result of the simulated outcomes, indicated that Blowfish performed better than the AES and RSA algorithms. According to Yegireddi and Kumar [25] conducted a survey to assess the efficiency of well-known conventional encryption techniques. It concluded that AES and Blowfish are the only algorithms that give speed and security due to their variable key.

III. Experimental Analysis

We have implemented the various symmetric encryptions in Python. Our performance evaluation is based on the implementation of three symmetric algorithms AES, Twofish and blowfish for encryption and decryption, and throughput. The following criteria were used: a) encryption and decryption time; b) throughput; and c) 128 key bit size for AES, Twofish, and Blowfish. To show the outcomes for the conclusion, the values for each criterion were logged and graphically plotted. The simulation was run on a laptop with an Intel® CoreTM i5-10210U CPU running at 2.40 GHz and 16 GB of RAM. Version 21H2 of Windows 11 Pro for Workstations was used. A key size of 128 bits was utilised as the benchmark in this experiment to acquire trustworthy values for evaluating the efficiency of AES, Blowfish, and Twofish cryptographic algorithms. The experiment was run three times and the mean execution time was recorded. The three block-cipher methods—AES, Blowfish, and Twofish—are also listed in Table I as a summary.

TABLE I. Key and Block Size

Factors	AES	Blowfish	Twofish
Key sizes	128	128	128
Block size	128 bits	64 bits	128 bits

IV. Performance Evaluation

A. Process Time (Encryption and Decryption Time)

Tables II to VII show the comparison of results. It is worth noting that AES-128 key bit size has the quickest encryption and decryption time on average.

TABLE II. AES – Encryption (128 Key Bit)

File format	File size (in kb)	MEAN
file_example_TXT	9	0.037459
file-example_PDF_1MB	1,018	0.368214
file_example_MP3_5MG	5,166	1.182518
file_example_MP4_1280_10MG	9,610	2.218504
file-sample_1MB_DOCX	1,003	0.247889
file_example_XLS_5000	657	0.171259
file_example_PPT_250kB	243	0.105667
file_example_JPG_2500kB	2,446	0.569065

TABLE III. Blowfish – Encryption (128 Key Bit)

File format	File size (in kb)	MEAN
file_example_TXT	9	0.010809
file-example_PDF_1MB	1,018	0.304263
file_example_MP3_5MG	5,166	1.436277
file_example_MP4_1280_10MG	9,610	2.874504
file-sample_1MB_DOCX	1,003	0.306568
file_example_XLS_5000	657	0.218169
file_example_PPT_250kB	243	0.111124
file_example_JPG_2500kB	2,446	0.718191

TABLE IV. Twofish – Encryption (128 Key Bit)

File format	File size (in kb)	MEAN
file_example_TXT	9	0.25
file-example_PDF_1MB	1,018	26.34
file_example_MP3_5MG	5,166	131.50
file_example_MP4_1280_10MG	9,610	244.11
file-sample_1MB_DOCX	1,003	25.39
file_example_XLS_5000	657	16.59
file_example_PPT_250kB	243	6.20
file_example_JPG_2500kB	2,446	63.02

TABLE V. AES – DECRYPTION (128 KEY BIT)

File format	File size (in kb)	MEAN
file_example_TXT	9	0.01238
file-example_PDF_1MB	1,018	0.288262
file_example_MP3_5MG	5,166	1.213215
file_example_MP4_1280_10MG	9,610	2.238994
file-sample_1MB_DOCX	1,003	0.298465
file_example_XLS_5000	657	0.200942
file_example_PPT_250kB	243	0.076602
file_example_JPG_2500kB	2,446	0.615148

TABLE VI. BLOWFISH – DECRYPTION (128 KEY BIT)

File format	File size (in kb)	MEAN
file_example_TXT	9	0.016267
file-example_PDF_1MB	1,018	0.298124
file_example_MP3_5MG	5,166	1.504732
file_example_MP4_1280_10MG	9,610	2.703225
file-sample_1MB_DOCX	1,003	0.298391
file_example_XLS_5000	657	0.21287
file_example_PPT_250kB	243	0.092704
file_example_JPG_2500kB	2,446	0.735764

TABLE VII. TWOFISH – DECRYPTION (128 KEY BIT)

File format	File size (in kb)	MEAN
file_example_TXT	9	0.244043
file-example_PDF_1MB	1,018	25.60037
file_example_MP3_5MG	5,166	135.4913
file_example_MP4_1280_10MG	9,610	245.67
file-sample_1MB_DOCX	1,003	25.33303
file_example_XLS_5000	657	16.62755
file_example_PPT_250kB	243	6.150904
file_example_JPG_2500kB	2,446	62.01268

B. *Throughput*

The throughput of an encryption scheme defines the speed of encryption. The encryption scheme's throughput is calculated by dividing the total plaintext in bytes encrypted by the encryption time [14]. In this experiment, the throughput is derived from calculated as the total plaintext in Kilobytes encrypted/encryption time (KB/sec) divided by their mean time generated. AES has the highest throughput making it the fastest of the three followed by blowfish. The results are shown in Tables VIII to X.

TABLE VIII. AES THROUGHPUT IN KILOBYTES/SECONDS (128 KEY BIT)

File Name	File Size (in kb)	Encryption Throughput	Decryption Throughput
	KB	KB/Sec	KB/Sec
file_example_TXT	9	240.2626872	726.9789984
file_example_PDF_1MB	1,018	2764.696617	3531.50953
file_example_MP3_5MG	5,166	4368.64386	4258.10759
file_example_MP4_1280_10MG	9,610	4331.747881	4292.106187
file_example_1MB_DOCX	1,003	4046.165824	3360.528035
file_example_XLS_5000	657	3836.294735	3269.600183
file_example_PPT_250kB	243	2299.677288	3172.240934
file_example_JPG_2500kB	2,446	4298.278756	3976.278879

TABLE IX. BLOWFISH THROUGHPUT IN KILOBYTES/SECONDS 128 KEY BIT)

File Name	File Size (in kb)	Encryption Throughput	Decryption Throughput
		KB/Sec	KB/Sec
file_example_TXT	9	832.6394671	553.2673511
file_example_PDF_1MB	1,018	3345.789662	3414.686506
file_example_MP3_5MG	5,166	3596.799225	3433.169495
file_example_MP4_1280_10MG	9,610	3343.185468	3555.012994
file_example_1MB_DOCX	1,003	3271.704809	3361.361435
file_example_XLS_5000	657	3011.426921	3086.390755
file_example_PPT_250kB	243	2186.746337	2621.246117
file_example_JPG_2500kB	2,446	3405.779243	3324.435553

TABLE X. TWOFISH THROUGHPUT IN KILOBYTES/SECONDS (128 KEY BIT)

File Name	File Size (in kb)	Encryption Throughput	Decryption Throughput
		KB/Sec	KB/Sec
file_example_TXT	9	36	36.87874678
file_example_PDF_1MB	1,018	38.64844343	39.76505027
file_example_MP3_5MG	5,166	39.2851711	38.12790932
file_example_MP4_1280_10MG	9,610	39.36749826	39.11751537
file_example_1MB_DOCX	1,003	39.50374163	39.59257933
file_example_XLS_5000	657	39.60216998	39.51273639
file_example_PPT_250kB	243	39.19354839	39.506388
file_example_JPG_2500kB	2,446	38.81307521	39.44354606

V. DISCUSSION OF RESULTS

Tables I to IX show the encryption time, decryption time, and throughput. Performance analysis varies based on a particular file type, but on average, AES outperforms Blowfish and Twofish in terms of speed and process time. Furthermore, the figures in Fig. 1 and Fig. 2 are based on the average of total encryption/decryption and throughput of AES, Blowfish, and Twofish. An overview of all the comparisons can be summarized into the following Table XI. The summary in Table XI is based on values from Fig. 1 and Fig. 2. AES-128 produced fast encryption, decryption times and speed than Blowfish and Twofish. The results show that Blowfish can match the encryption and decryption speeds of AES.

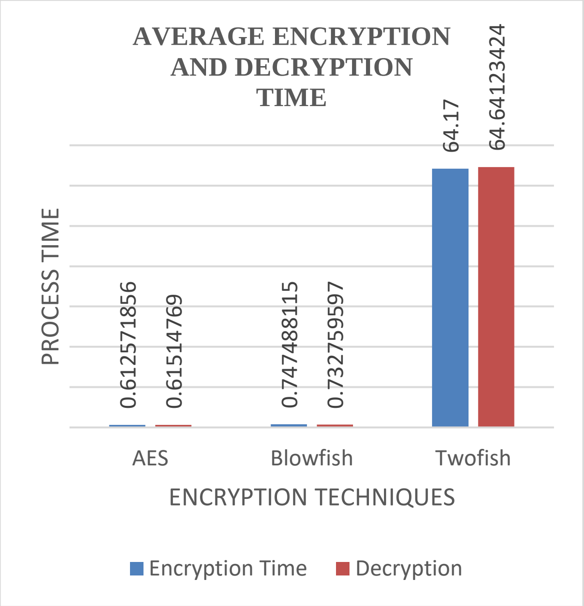


Fig. 1. Average process for AES, blowfish and twofish.

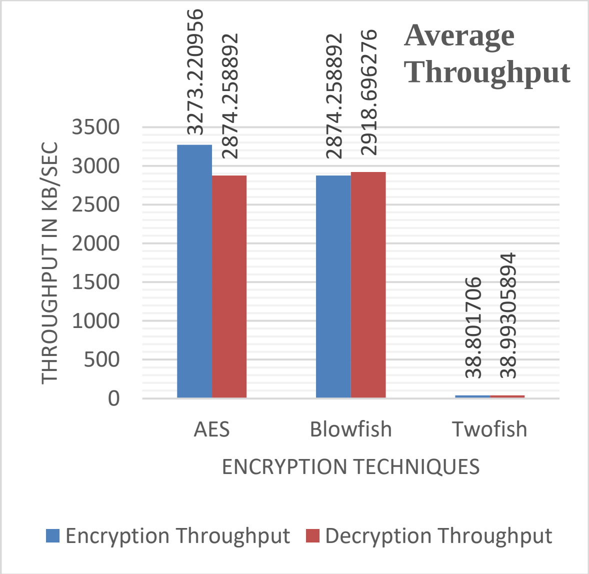


Fig. 2. Average throughput for AES, blowfish, and twofish.

TABLE XI. AES, BLOWFISH, AND TWOFISH AN OVERALL COMPARISON

Parameters	AES	Blowfish	Twofish
Key bit size	128	128	128
Encryption	Very fast	Fast	Too slow
Decryption	Very fast	Fast	Too slow
Throughput (Speed)	Very high	High	Low

VI. CONCLUSION

In today's rapidly expanding Internet and network applications, encryption algorithms play a critical role in ensuring information security. Based on a key bit size of 128 in this study, we evaluated three symmetric key encryption algorithms: AES, Twofish, and Blowfish. Based on the experimental results, the 128 key bit of AES algorithm has the shortest process time and runs quicker than Twofish and Blowfish. Overall results proved that the AES algorithm is more suitable for secure data transfer.

REFERENCES

[1] P. K. Ghosh, S. K. Ghosh, and L. M. Khan, “Current trend of bank selection criteria of retail customers in Bangladesh: An investigation,” Glob. Bus. Financ. Rev., vol. 20, no. 2, pp. 27–34, 2015, doi: 10.17549/gbfr.2015.20.2.27.

[2] M. Panda, “Performance analysis of encryption algorithms for security,” in International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016 - Proceedings, 2017, pp. 278–284, doi: 10.1109/SCOPES.2016.7955835.

[3] D. S. Abd Elminaam, H. M. A. Kader, and M. M. Hadhoud, “Evaluating the performance of symmetric encryption algorithms,” Int. J. Netw. Secur., vol. 10, no. 3, pp. 213–219, 2010.

[4] B. Xing, D. D. Wang, Y. Yang, Z. Wei, J. Wu, and C. He, “Accelerating DES and AES Algorithms for a Heterogeneous Many-core Processor,” Int. J. Parallel Program., vol. 49, no. 3, pp. 463–486, 2021, doi: 10.1007/s10766-021-00692-4.

[5] S. N. Karale, K. Pendke, and P. Dahiwale, “The survey of various techniques & algorithms for SMS security,” ICIIECS 2015 - 2015 IEEE Int. Conf. Innov. Information, Embed. Commun. Syst., 2015, doi: 10.1109/ICIIECS.2015.7192943.

[6] M. Panda and A. Nag, “Plain Text Encryption Using AES, DES and SALSA20 by Java Based Bouncy Castle API on Windows and Linux,” Proc. - 2015 2nd IEEE Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2015, pp. 541–548, 2015, doi: 10.1109/ICACCE.2015.130.

[7] P. Nema and M. A. Rizvi, “Critical Analysis of Various Symmetric Key Cryptographic Algorithms,” Int. J. Recent Innov. Trends Comput. Commun., vol. 3, no. 6, pp. 4301–4306, 2015.

[8] N. Tyagi and A. Ganpati, “Comparative Analysis of Symmetric Key Encryption Algorithms,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 4, no. 6, pp. 94–99, 2014.

[9] C. Tan and Q. Ji, “An approach to identifying cryptographic algorithm from ciphertext,” in Proceedings of 2016 8th IEEE International Conference on Communication Software and Networks, ICCSN 2016, 2016, pp. 19–23, doi: 10.1109/ICCSN.2016.7586649.

[10] G. Singh, A. Kumar, and K. S. Sandha, “A Study of New Trends in Blowfish Algorithm,” Int. J. Eng. Res. Appl. www.ijera.com, vol. 1, no. 2, pp. 321–326, 2015, [Online]. Available: www.ijera.com.

[11] G. Singh and S. Supriya, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security,” Int. J. Comput. Appl., vol. 67, no. 19, pp. 33–38, 2013, doi: 10.5120/11507-7224.

[12] A. Ramesh and A. Suruliandi, “Performance analysis of encryption algorithms for information security,” Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2013, pp. 840–844, 2013, doi: 10.1109/ICCPCT.2013.6528957.

[13] S. Gautam, S. Singh, and H. Singh, “A Comparative Study and Analysis of Cryptographic Algorithms: RSA, DES, AES, BLOWFISH, 3-DES, and TWOFISH,” Int. J. Res. Electron. Comput. Eng., vol. 7, no. 1, 2019, [Online]. Available: https://www. researchgate.net/publication/ 334724160.

[14] T. Nie, C. Song, and X. Zhi, “Performance evaluation of DES and Blowfish algorithms,” 2010 Int. Conf. Biomed. Eng. Comput. Sci. ICBECS 2010, pp. 16–19, 2010, doi: 10.1109/ICBECS.2010.5462398.

[15] M. Anand Kumar and S. Karthikeyan, “Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms,” Int. J. Comput. Netw. Inf. Secur., vol. 4, no. 2, pp. 22–28, 2012, doi: 10.5815/ijcnis.2012.02.04.

[16] M. Suresh and M. Neema, "Hardware Implementation of Blowfish Algorithm for the Secure Data Transmission in Internet of Things," Procedia Technol., vol. 25, no. Raerest, pp. 248–255, 2016, doi: 10.1016/j.protcy.2016.08.104.

[17] C. Haldankar and S. Kuwelkar, "Implementation of Aes and Blowfish Algorithm," Int. J. Res. Eng. Technol., vol. 03, no. 15, pp. 143–146, 2014, doi: 10.15623/ijret.2014.0315026.

[18] H. Dibas and K. E. Sabri, "A comprehensive performance empirical study of the symmetric algorithms:AES, 3DES, Blowfish and Twofish," 2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc., pp. 344–349, 2021, doi: 10.1109/ICIT52682.2021.9491644.

[19] A. Ghosh, "Comparison of Encryption Algorithms : AES , Blowfish and Twofish for Security of Wireless Networks," Int. Res. J. Eng. Technol., no. June, pp. 4656–4659, 2020, doi: 10.13140/RG.2.2.31024.38401.

[20] J. Raigoza and K. Jituri, "Evaluating Performance of Symmetric Encryption Algorithms," Proc. - 2016 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2016, pp. 1378–1379, 2017, doi: 10.1109/CSCI.2016.0258.

[21] S. S. Ghosh, H. Parmar, P. Shah, and K. Samdani, "A Comprehensive Analysis between Popular Symmetric Encryption Algorithms," 1st Int. Conf. Data Sci. Anal. PuneCon 2018 - Proc., 2018, doi: 10.1109/PUNECON.2018.8745324.

[22] S. A. M. Rizvi, S. Z. Hussain, and N. Wadhwa, "Performance analysis of AES and Twofish encryption schemes," Proc. - 2011 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2011, pp. 76–79, 2011, doi: 10.1109/CSNT.2011.160.

[23] A. V. Mota, A. Sami, K. C. Shanmugam, Bharanidharan Yeo, and K. Krishnan, "Comparative Analysis of Different Techniques of Encryption for Secured Data Transmission," IEEE Int. Conf. Power, Control. Signals Instrum. Eng., vol. 54, no. 4, pp. 847–860, 2017.

[24] R. S. Cordova, R. L. R. Maata, A. S. Halibas, and R. Al-Azawi, "Comparative analysis on the performance of selected security algorithms in cloud computing," 2017 Int. Conf. Electr. Comput. Technol. Appl. ICECTA 2017, vol. 2018-Janua, pp. 1–4, 2017, doi: 10.1109/ICECTA.2017.8252030.

[25] R. Yegireddi and R. K. Kumar, "A survey on conventional encryption algorithms of Cryptography," Proc. 2016 Int. Conf. ICT Business, Ind. Gov. ICTBIG 2016, pp. 6–9, 2017, doi: 10.1109/ICTBIG.2016.7892684.

Elitist Animal Migration Optimization for Protein Structure Prediction based on 3D Off-Lattice Model

Ezgi Deniz Ülker
Department of Software Engineering, European University of Lefke, Mersin-10, Turkey

***Abstract*—Predicting the structure of protein has been the center of attraction for the researchers. The aim is to make a reliable prediction of the protein structure by obtaining the minimum energy values among amino acids interactions. According to the generated shape of amino acids, the functionality of the proteins can be determined. However, it is known as one of the most challenging tasks in the field of bioinformatics considering its high computation complexity. Metaheuristic algorithms are mainly preferred by researchers from various fields, since their performances are quite satisfactory in solving such complex problems. Animal Migration Optimization (AMO) algorithm is a metaheuristic approach which mimics the behavior of animals during the migration process. However, in this research to reach a high solution quality, an elitist version of Animal Migration Optimization (ELAMO) algorithm is considered and in particular it is applied to Protein Structure Prediction (PSP) problem. The performance of ELAMO is tested on some well-studied artificial and real protein sequences, and then compared with powerful optimization algorithms which are specially designed for solving PSP problem. The results show that ELAMO is quite capable in solving this problem. Hence, it can be used as an efficient optimizer for solving complex problems that require better solution quality in the field of bioinformatics.**

Keywords*—*Animal migration optimization; bioinformatics; elitism; metaheuristics; protein structure prediction

I. Introduction

In molecular biology, comprehending the structure of a protein sequence reveals the hidden functionalities of the life [1]. When the proteins are folded in different ways, the information necessary for understanding their functionalities will arise. Proteins are formed by the combination of amino acids which are connected by peptide bonds [2]. According to Christen Anfinsen's leading work, proteins can be found in the lowest energy levels which are called Gibbs energy level, when they are in three dimensional states [3]. Protein Structure Prediction (PSP) problem is located on finding this state by seeking the minimum Gibbs energy level. As the amino acid sequence becomes large, predicting the structure of a protein sequence becomes complex.

Researchers developed an approach called 'HP model' for protein folding prediction [4]. In the HP model, a protein description is made up of smaller pieces called monomers and which are either represented on 2D or 3D surface. 'H' and 'P' letters are used to define each of the monomers which are hydrophobic or polar, respectively. It is aimed to find the optimal structure of a given H-P chain that is defined as the maximum number of H-H bondings. Although the HP model is specifically designed for solving protein folding with its simplicity, it does not provide satisfactory solutions for PSP. The problem is proved to be an NP-hard problem due to large number of amino acids sequences and requires quite efficient algorithms to solve them [5–7].

One of the biggest limitations of protein folding is having the multiple local optimum points in the free-energy space and the global optimum is located in between these points which is quite challenging to obtain [8]. In order to design a scheme by avoiding the large computational cost, such models with eliminated properties in protein folding have been preferred [8–10]. An accurate example of these kinds of models is the off-lattice model which is presented by Stilinger et al. [8]. The model is employed to simplify the protein folding.

Animal Migration Algorithm (AMO) is a bioinspired metaheuristic approach proposed by Li et al. [11]. It is founded on an animal's instinct to follow their close neighbors during the migration and has quite validated performances on many optimization problems. Despite of the noticeable properties of AMO, there may be some disadvantages such as low convergence rate by choosing the next possible solutions only among the current animal's neighborhood or having a less chance by finding the global optimum because of following the wrong neighbors. In order to avoid these limitations and make it guarantee that the algorithm converges to the global optimum in less number of iterations, Elitist Animal Migration Optimization (ELAMO) is proposed on the basis of an animal's instinct to follow their leaders, not only their closest neighbors [12]. ELAMO has a validated performance in solving combinatorial NP-hard problem. However, to the best of our knowledge neither AMO nor ELAMO have been proposed for solving bioinformatics problems, particularly, for the PSP problem. In this paper, ELAMO algorithm is adapted to bring another aspect in solving Protein Structure Prediction (PSP) problem by using 3D AB Off-Lattice Model.

The rest of the paper is designed as follows; in Section II, some important studies for solving the PSP over the years have been given. In Section III, three dimensional AB off lattice model and the adaptation of Elitist Migration Algorithm to PSP problem with the model equations are given. In Section IV, ELAMO algorithm's performance is compared on both synthetic and real protein sequences with some powerful optimizers. The obtained results with the visual representations of minimum energy configurations and discussions are given in detail in this section. Lastly, in Section V, the concluding remarks are given.

II. Related Work

It is known that many metaheuristic algorithms are verified to be quite efficient in solving complex and even NP-hard problems. In specific, it is found that Bee colony optimization algorithms and its variants are used to solve PSP problem by Li et al. [13]. Kalegari and Lopes proposed an improved Differential Evolution algorithm for solving PSP using 2D and 3D off-lattice models efficiently [14]. Another Differential Evolution algorithm variant is proposed by Rakhshani et al. for solving complex protein structure prediction problems [15]. Deep learning practices are also tested for the PSP by Senior et al. [16] and achieved promising results. Schauperl and Denny performed an AI based protein structure prediction in drug discovery [17], Chowdhurry et al. [18] solved protein prediction problem using a deep learning model and Weißenow et al. [19] have solved PSP problem using AI model accurately. Multi-meme algorithms are also adapted for solving PSP by Krasnogor et al. [20]. Lin and Zhang introduced a novel-hybrid global optimization method by forming Genetic Algorithm and Particle Swarm Optimization to solve PSP in which aiming to produce lower energy conformation levels [21]. Boiani and Parpinelli proposed a hybrid algorithm called cuHjDE-3D which is formed by self-adaptive Differential Evolution that uses jDE and Hooke-Jeeves Direct Search (HJDS) [22].

The literature review revealed that the standard metaheuristic algorithms have limited performance in solving PSP problem. Before attempting to solve the problem, researchers either modify or hybridize the standard algorithms. By introducing such boosted algorithms, the researchers aimed to use the strengths of the algorithms on PSP problem.

In this study, none of the machine learning methods are implemented. Instead, a modified version of AMO algorithm has been studied to observe how it evaluates the problem by using its parameters. One of the main contributions of this study is to adapt animal migration algorithm by enhancing its diversity using elitist approach for protein sequence prediction problem and achieving satisfactory results.

III. Materials and Methods

A. Three Dimensional AB Off-Lattice Model

The AB off-lattice model is stimulated by HP model and considered as one of the useful models for solving Protein Structure Prediction Problem. When the AB off-lattice model was introduced, it was initially designed for 2-D protein structures. However, then the model was upgraded for solving 3-D models as well [8, 23].

In a protein sequence 20 types of amino acids exist which are categorized in two; hydrophobic and hydrophilic. This is simply performed according to their affinity to water. The amino acids then translated to two specialized monomers ‘A’ and ‘B’. As K-D method proposes I, V, L, P, C, M, A, G are hydrophobic amino acids represented by letter A and D, E, F, H, K, N, Q, R, S, T, W, Y are hydrophilic amino acids represented by letter B [24]. The name AB of AB off-lattice model is because of these specialized monomers A and B.

The amino acids are bonded with each other by chemical bonds and can be placed anywhere in the 3D space. The reason of that is called off-lattice is the positions of the amino acids which are not restricted by a lattice. In the AB off-lattice model, bondings are formed by set of angles; folding (θ) and rotation (ϕ). In a protein sequence which contains n monomers also contains n-2 folding angles and n-3 rotation angles. The optimal structure of AB off-lattice model produces the free energy which gives general information about the physical and chemical concept of protein sequences. Fig. 1 shows the representation of an artificial protein sequence ABAA with folding (θ) and rotation (ϕ) angles where ‘A’ and ‘B’ are hydrophobic and hydrophilic amino acids, respectively. Folding angles [θ1, θ2] and rotation angle [ϕ1] are needed to be optimized for having the minimum free energy level.

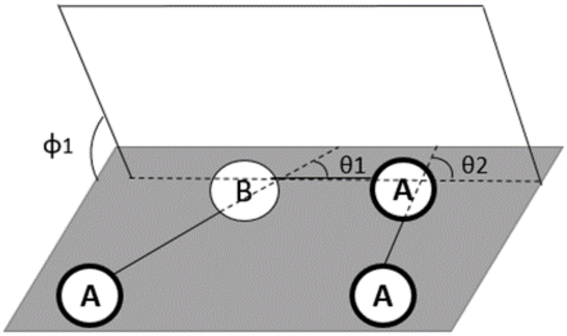


Fig. 1. 3D AB-off lattice model representation for ABAA artificial protein sequence.

Adaptation of protein structure prediction to a numerical optimization problem by 3D AB-off lattice model is done as follows,

$$Energy = \sum_{i=2}^{n-1} \frac{1-\cos\theta_i}{4} + 4\sum_{i=1}^{n-2}\sum_{j=i+2}^{n}\left[r_{ij}^{-12} - C(\varepsilon_i,\varepsilon_j) r_{ij}^{-6}\right] \quad (1)$$

where ε_i is the characteristic of the i^{th} amino acid. If $\varepsilon_i = 1$, then *i* is a hydrophilic amino acid. If $\varepsilon_i = -1$, then *i* is a hydrophobic one. The folding angles (θ) are bounded [-180°, 180°].

To obtain the distance between amino acids *i* and *j*, the following equation is used.

$$r_{ij=}\sqrt{\left[1+\sum_{k=i+1}^{j-1}\cos\left(\sum_{l=i+1}^{k}\theta_l\right)\right]^2 + \left[\sum_{k=i+1}^{j-1}\sin\left(\sum_{l=i+1}^{k}\theta_l\right)\right]} \quad (2)$$

The following equation shows the basic rotations for two amino acids.

$$C(\varepsilon_i,\varepsilon_j) = \frac{1}{8}\left(1+\varepsilon_i+\varepsilon_j+5\varepsilon_i\varepsilon_j\right) \quad (3)$$

In AB-off lattice model, it is assumpted that strong correlations between AA pairs result with the value of $C(\varepsilon_i,\varepsilon_j) = 1,$ relatively weaker correlations between BB pairs result with the value of $C(\varepsilon_i,\varepsilon_j) = 0.5$ and different pairs BA or AB pairs result with the value of $C(\varepsilon_i,\varepsilon_j) = -0.5$. Using the assumptions obtained through AB-off lattice model, the protein structure problem is converted into a numerical optimization problem that can be handled by evolutionary optimization techniques. By having various ordering of distances and rotations, different energy levels of amino acids are obtained and as the algorithm iterates the optimum energy level is obtained.

B. Adaptation of ELAMO to PSP Problem

Animal migration is a common behavior which belongs to animal herds to be used in discovering better places to live and reproduce. Animal Migration Optimization (AMO) algorithm based on this behavior and proved to be a validated optimizer in solving optimization problems [11]. Our approach is simply based on the main steps of AMO by including the elitism behavior in it. In the Elitist Animal Migration approach, the neighborhood structure of the standard AMO is reconstructed. Thus, the animals in the herd follow their leaders not only their close neighbors.

During the migration process, an animal's position depends on its neighbor. In standard AMO, migration is done by following five closest neighbors of each animal. However, in our elitist approach, an animal's instinct to follow the leader of the herd is essential. In a typical animal herd, there are three kinds of animal; Alpha (α), Beta (β) and Omega (ω).

Alpha (α) is responsible from all animals in the herd such as finding the preys or discovering new life areas. If the alpha dies, a new leader is selected among the beta (β) animals who are in charge after the alpha (α). The rest of the animals are considered as omegas (ω) who obey the rules of the herd. In the algorithm ELAMO, only α and β animals are in charge of migration. The number of α is 1 and the number of β is fixed to 4. Thus, all of the animals in the herd move towards to new life areas by following these leaders. As the algorithm iterates, new α and β animals are selected with respect to their positions among the rest of the animals. The following figures; Fig. 2 and Fig. 3 demonstrate the neighborhood structure in AMO and ELAMO, respectively where each animal represented by a circle theoretically.

Elitist animal migration algorithm is built up on two fundamental steps; Animal migration and population updating.

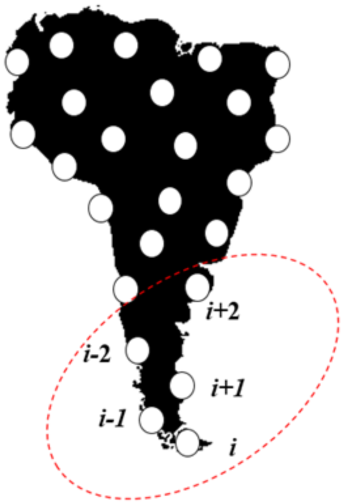


Fig. 2. The neighborhood structure of AMO algorithm where *i* is an animal and *i* ± 1,2 are the closest neighbors.

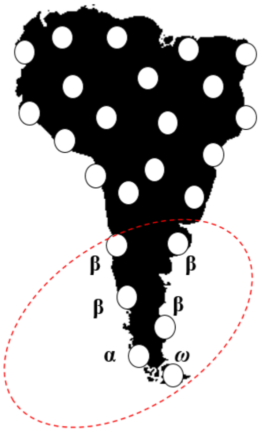


Fig. 3. The neighborhood structure of ELAMO algorithm.

In animal migration step, animals change their positions towards their α and β animals as given in (4). In population updating step, displacement of animals is introduced. Some animals may be eliminated due to death or they may compete for their positions and the losers are discarded from the population. The new positions are updated according to their fitness values as it is shown in (5).

Animal migration step;

$$X_{i,G+1} = X_{i,G} + \delta\left(X_{neighborLeader,G} - X_{i,G}\right) \quad (4)$$

where δ is a random number produced by Gaussian distribution and G is the dimension for each animal ∈ [1... D] and $X_{\mathrm{neighborLeader}}$ is the leader's position randomly selected from the neighborhood structure of an animal X_i.

Population updating step;

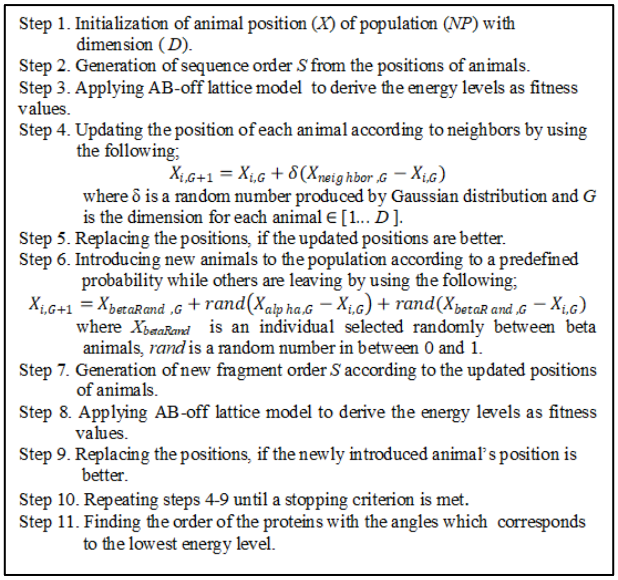
$$X_{i,G+1} = X_{betaRand_a,G} + rand\left(X_{alpha,G} - X_{i,G}\right) + rand\left(X_{betaRand_b,G} - X_{i,G}\right) \quad (5)$$

where X_{betaRand} is an animal selected randomly between beta animals, X_{alpha} is the position of alpha, rand is a random number in between 0, 1 and a ≠ b.

The main control parameters of ELAMO are δ and $rand$ which influence the population by having equilibrium between the diversity; exploring new possible areas in the search space and intensity; focusing the search area around the leaders α and β.

The adaptation of ELAMO to PSP problem is given in the Fig. 4. One of the main motivations is to reach the higher optimization level by changing the neighborhood structure. In standard AMO, the closest neighbor's position might be used just because they are considered as the closest neighbors and even if their positions are relatively worse than the others. However, in ELAMO, the best positions are chosen at each iteration and are followed by all of the animals.

As it is explained clearly in no-free lunch theorem [25], when an algorithm's performance is sufficient in some aspects, the performance may not reach to that level as it is expected for the other aspects. In ELAMO, individuals are discarded from the population as the elitism feature requires even in the beginning of the iterations. Therefore, it may be considered in ELAMO the balance of diversification and intensification may be negatively affected by the loss of individuals in the earlier stages of optimization.



Step 1. Initialization of animal position (X) of population (NP) with dimension (D).
Step 2. Generation of sequence order S from the positions of animals.
Step 3. Applying AB-off lattice model to derive the energy levels as fitness values.
Step 4. Updating the position of each animal according to neighbors by using the following;

$$X_{i,G+1} = X_{i,G} + \delta(X_{neighbor,G} - X_{i,G})$$

where δ is a random number produced by Gaussian distribution and G is the dimension for each animal ∈ [1... D].
Step 5. Replacing the positions, if the updated positions are better.
Step 6. Introducing new animals to the population according to a predefined probability while others are leaving by using the following;

$$X_{i,G+1} = X_{betaRand,G} + rand(X_{alpha,G} - X_{i,G}) + rand(X_{betaRand,G} - X_{i,G})$$

where $X_{betaRand}$ is an individual selected randomly between beta animals, $rand$ is a random number in between 0 and 1.
Step 7. Generation of new fragment order S according to the updated positions of animals.
Step 8. Applying AB-off lattice model to derive the energy levels as fitness values.
Step 9. Replacing the positions, if the newly introduced animal's position is better.
Step 10. Repeating steps 4-9 until a stopping criterion is met.
Step 11. Finding the order of the proteins with the angles which corresponds to the lowest energy level.

Fig. 4. The main steps of ELAMO algorithm for PSP problem.

A herd with α, β and ω animals correspond an AB-off lattice model for a set of sequences. All of the animals in the herd are the potential solutions in solving PSP by corresponding distances, angles and interactions between particles. According to these values, energy levels are derived. In PSP, the objective function is the energy level function and the optimal solution refers to the lowest energy value.

IV. Results and Discussion

In this section, two sets of analyses were used. First, a set of artificial Fibonacci protein sequences which have been used as benchmarks commonly in the literature is studied [8, 26] and then real protein sequences were analyzed. The data sets are experimentally examined structures for testing the efficiency of the methods for PSP problem.

A list of benchmark sequences with the 'A' and 'B' monomers are given in Table I where N is the sum of the monomers. A comparative study is performed in Table II to observe the performance of ELAMO with respect to some powerful optimization algorithms; Improved Particle Swarm Optimization (EPSO) [27], Internal Feedback strategy based on Artificial Bee Colony Algorithm (IF-ABC) [28], Combination of Genetic Algorithm and Particle Swarm Optimization (GAPSO) [21] and standard AMO which ELAMO is originated from. It is also important to note that the results of the compared algorithms included to the comparison table as they appeared in their original studies.

All simulations are implemented on an Intel Core i5 CPU with 4 GB RAM running at 3.10 GHz by C++ language. All benchmark sequences are evaluated for 30 independent runs with random initial points. All protein sequences are optimized by AMO and ELAMO for 50,000 number of iterations.

TABLE I. Details of the Artificial Protein Sequences

Coded Sequence	N
ABBABBABABBAB	13
BABABBABABBABBABABBAB	21
ABBABBABBBABBABABBABABBABBABABBAB	34

Fig. 5 is the representation of the changing lowest free energy values for three artificial protein sequences by the algorithms chosen for comparison. It can be clearly seen from both Table II and Fig. 5 that ELAMO has reached better energy values than the other algorithms for all artificial sequences.

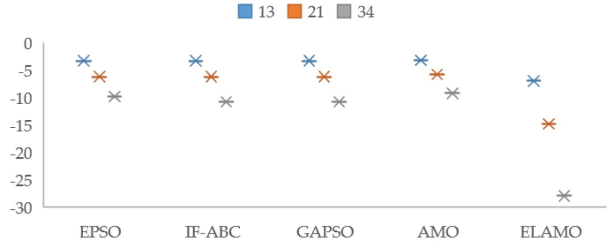


Fig. 5. A comparative free energy values obtained by different algorithms for Fibonacci sequences.

As shown in Fig. 5, almost all of the algorithms converge in similar rate in the first level of iterations; however as the iteration goes on convergence rate of ELAMO stands out for all benchmarks than the others. When the original algorithm AMO compared with the others, it is seen that all of the selected algorithms converge better than AMO.

All of the algorithms selected for comparison are either the hybrid version of one or more algorithms or the improved versions of the originals. AMO algorithm is the only one its performance was not boosted. It might be the main reason of the observation above which is about the lower convergence rate of AMO. On the other hand, this is a good indication that AMO has been improved efficiently to handle such kind of problems with high convergence rates.

In order to observe the effect of the added elitism feature on AMO, a detailed comparison between AMO and ELAMO is studied and shown in the Table III, where *Best* is for the lowest free-energy value obtained after 50,000 number of iterations, *Avg* is the averaged free-energy values of 30 independent runs and *Stdev* is for the standard deviation value.

TABLE II. The Lowest Free-Energy Values of Benchmark Sequences Obtained by the Algorithms

List of sequences	N	EPSO	IF-ABC	GAPSO	AMO	ELAMO
ABBABBABABBAB	13	-3.294	-3.294	-3.294	-3.216	-6.982
BABABBABABBABBABABBAB	21	-6.198	-6.198	-6.210	-5.751	-14.811
ABBABBABBBABBABABBABABBABBABABBAB	34	-9.834	-10.806	-10.789	-9.219	-27.989

The results denote the improved solution quality of ELAMO over AMO as well as robustness. However, for all of the artificial sequences the *Best* and the *Avg* values of ELAMO are superior than the values AMO. However, only for the sequence with the length 34, the *Stdev* value is not satisfactory as it is expected. ELAMO algorithm produces encouraging results for the lowest-free energy values and the average values, but does not achieve the development of standard deviation for this sequence. When the *Stdev* values are compared for the sequence with the length 21, it is seen that AMO is quite close to ELAMO. The reason of this unexpected performance might be because of α and β animals' convergence pace in the search space, while some of the ω animals are still in the optimization process. As the iterations progress, the majority of ω animals find their way towards the optimum.

TABLE III. COMPARATIVE RESULTS BEST, AVERAGE AND STANDARD DEVIATIONS ACHIEVED BY AMO AND ELAMO

	AMO			ELAMO		
N	**Best**	**Avg**	**Stdev**	**Best**	**Avg**	**Stdev**
13	-3.21	-2.44	0.52	-6.98	-6.38	0.39
21	-5.75	-4.71	1.17	-14.81	-14.31	0.99
34	-9.21	-8.02	1.20	-27.98	-26.30	1.83

Researchers dealt with well-preferred real protein benchmark functions for a better analyzing of their algorithms and employed to their works [15, 28–31]. The same protein sequences were used in this study as benchmark functions and rewritten according to the K-D method where I, V, L, P, C, M, A, G are hydrophobic amino acids represented by letter 'A' and D, E, F, H, K, N, Q, R, S, T, W, Y are hydrophilic amino acids represented by letter 'B'. The list of the amino acid sequences were selected from the widely used Protein Data Bank (PDB) database with different lengths to make a more efficient comparison with the other algorithms. The IDs, lenghts and contained amino acids are given in the Table IV.

In the study with real protein sequences, only one standard algorithm chosen for the thorough check. Instead, specially designed hybrid algorithms with significant performances are selected. Table V lists the lowest free energy values obtained by ELAMO as well as other competitive algorithms to make an extensive comparison. Convergence values can be seen even in earlier iterations in the Fig. 6.

TABLE IV. DETAILS OF THE REAL AMINO ACID SEQUENCES

PDB ID	N	Sequence
1AGT	38	AAAABABABABABAABAABBAAABBABAABBB ABABAB
1HVV	75	BAABBABBBBBBAABABBBABBABBABABAAAA ABBBABAABBABBBABBAABBABBAABBBBBAA BBBBBABBB
1GK4	84	BABAABABBBBABBBABBABBBBAABAABBBBB AABABBBABBABBBAABBABBBBBAABABAAAB ABAABBBBAABABBBBA
2EWH	98	AABABAAAAAAABBBAAAAAABAABAABBAAB ABAAABBBAAAABABAAABABBAAABAAABAA ABAABBAABAAAAABAAABA BBBABBAAABAABA

TABLE V. COMPARISON OF THE LOWEST FREE-ENERGY VALUES FOR THE SEQUENCES OBTAINED BY THE ALGORITHMS

PDB ID	N	CMAES	L-SHADE	E-MASA-PAMS	ABC	ELAMO
1AGT	38	-34.45	-39.31	-41.28	-25.65	-50.82
1HVV	75	-27.35	-28.77	-	-27.35	-46.98
1GK4	84	-32.72	-40.26	-	-32.72	-49.89
2EWH	98	-	-	-125.53	-	-146.81

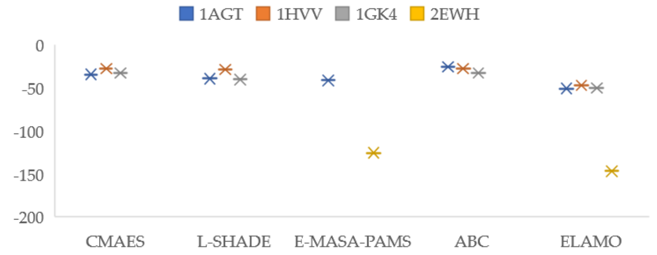


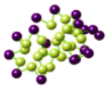
Fig. 6. A free energy values obtained by different algorithms for real protein sequences.

It can be seen from the figure that at the very first level of optimization all of the algorithms' convergence rates are very similar, but in the latter iterations elitism feature begins to appear in ELAMO and this affects the convergence rate in a very desirable way. The test parameters and results of the compared algorithms are accepted as they appear in the references [15, 28–31]. The best results for AMO and ELAMO are obtained over 30 runs. The stopping condition is set 200,000 iterations.

It can be acquired from both Table V and Fig. 6 that the performance of ELAMO can be distinguished from the others by having more precise solutions of the problems. Considering all of the results, it is possible to say that the algorithms' performances are very competitive but in the view of high solution quality, the performance of ELAMO is quite noteworthy.

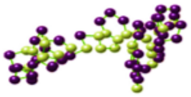
In the comparison of ELAMO with one of the standard algorithms ABC, it is observable that ELAMO produces better results. For the other competitive algorithms, we can say that CMAES and L-SHADE performances are similar to each other, while E-MASA-PAMS superior than both and produces comparable results with ELAMO. However, in both analysing, we can see that ELAMO always contributes with the lowest free energy values and it can be a distinct evident by applying a set of modifications on the right steps of the AMO algorithm, influence the performance of ELAMO in a quite remarkable way. It is note to point that the compared algorithms can not be further analzyed since there is no *Stdev* or *Avg* reported in the literature.

In the light of findings, the visual representation of the folded protein structures by the best run of ELAMO are shown in the Fig. 7 **(a)** to **(d)**. In the figures, green dots represent the hydrophobic monomer; 'A' and the purple represent the hydrophilic monomer; 'B'. As the figures reveal, the hydrophobic monomers are frequently enclosed by the hydrophilic monomers in the folding structures. This is a natural phenomena for avoiding contact with water molecules and is verified by the following figures as well.



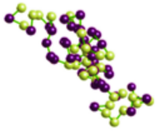
223.66, 32.18, 48.43, 76.89, 212.00, 146.29, 0.46, 139.28, 43.33, 3.16, 33.03, 35.24, 0.92, 184.10, 39.47, 49.33, 13.00, 5.42, 206.19, 33.23, 153.04, 331.12, 56.75, 23.29, 33.01, 33.84, 95.07, 149.35, 137.13, 229.69, 125.15, 1.37, 66.11, 132.91, 26.19, 32.75, 36.34, 24.63, 35.19, 2.01, 112.56, 93.59, 128.79, 15.64, 47.25, 46.16, 156.03, 74.60, 172.23, 124.18, 21.63, 81.45, 10.71, 12.28, 127.12, 207.54, 82.80, 45.45, 126.11, 39.98, 4.48, 9.16, 12.47, 32.54, 59.84, 96.30, 112.60, 128.79, 147.19, 267.96

(a) 1AGT



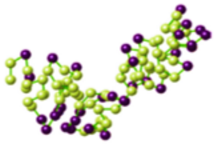
−78.08, 7.28, −14.45, −132.87, −167.25, 54.00, 21.14, −34.93, 11.80, −57.25, 18.88, 11.20, −39.26, 17.52, −154.37, 36.32, 3.58, 19.50, −30.39, 35.72, 12.98, −68.02, 25.28, −50.52, 6.12, 80.69, 30.41, −59.34, 14.24, −50.80, 0.49, 68.66, −51.06, −174.82, 9.51, −33.82, 28.14, −54.12, 175.91, 55.70, 177.17, 97.09, 166.90, 170.78, 25.60, −49.67, −51.94, −9.79, 35.65, −117.91, 161.06, −26.50, 62.46, 33.48, −76.72, 162.59, 58.15, −58.83, 67.78, 174.56, −16.23, 67.62, −73.03, 155.92, −29.30, −83.80, −8.46, −54.44, 144.51, 3.52, 2.14, 174.95, −78.80, 24.54, 33.24, −6.4086, 38.46, −100.67, 65.29, 28.51, 15.07, 159.77, 15.45, −90.47, −1.64, −6.8137, 22.80, 26.08, −9.87, −3.42, −26.46, −24.71, −162.20, −117.34, 162.15, 152.75, 117.53, 74.25, 12.50, 83.76, 61.42, 129.36, −41.02, 128.19, −94.87, −24.05, 85.90, −12.5094, 8.13, 111.77, −1.04, −137.84, 121.88, 154.04, −117.64, −172.42, −54.19, 9.07, −114.3532, −133.10, −104.39, 5.02, 108.16, 23.18, −2.51, 53.69, 57.68, −36.98, 88.47, −15.72, −30.12, 149.22, −97.60, 23.84, 14.65, −107.39, 110.41, 74.87, −36.92, 4.87, −1.1012, −123.57, 162.80

(b) 1HVV



−75.98, −19.89, 152.27, 1.22, −124.68, −49.11, 178.80, −23.15, 5.85, 89.17, 57.98, −45.97, −20.26, 13.95, −0.31, 27.52, 9.37, −90.31, −6.02, 6.73, 4.50, −96.57
−23.12, −66.86, 0.51, 102.37, 44.03, −62.46, 12.44, 3.79, 6.77, 30.54, 3.24, −33.32, 22.21, 43.47, 8.94, −58.82, 7.59, 55.36, −86.26, 13.84, −13.54, −13.86
−91.17, −8.67, −63.36, −16.58, −153.32, 65.33, 39.87, 30.10, 16.00, 12.89, 2.29, 59.19, −30.87, −155.11, 170.02, 109.21, 50.73, −32.88, 61.77, 36.36, 58.27, −47.112, −28.37, 26.82, 171.15, −59.22, −161.79, 43.37, 76.11, 102.12, 80.47, 143.49, 156.24, −13.57, −115.15, −158.24, 19.39, 77.60, −24.96, −30.31, 102.26, 44.31, −139.51, 138.27, 122.82, 76.57, 44.45, −163.63, 73.17, 17.55, 66.05, 112.66, −142.48, 143.86
72.45, 13.21, 174.81, −91.18, −56.72, −86.21, −54.81, 146.23, 124.02, 105.35, 103.77,−54.77, 36.85, −27.89, −10.96, 16.89, 121.51, 30.25, 0.81, 172.67, 74.53, −48.81,87.16, −7.18, 24.08, −29.12, −7.43, −12.15, −27.19, 11.71, −10.91, −6.66, −55.84, −1.06, −45.45, 5.52, 48.82, −14.32, −21.68, −27.25, 66.38, −124.22, 37.98, 22.97, −53.73, 72.34, −133.22, 111.17, −72.73, 0.40, −42.13, 81.12, −19.26, 2.67, −40.61,−69.59, 34.25, 6.18, −70.87, 4.77, 78.10, 103.67, −25.62, 0.06, 124.56

(c) 1GK4



17.20, 22.31, 130.91, 43.17, −65.86, 45.22, 45.39, 66.39, −38.17, −13.29, −11.77, 162.28, −1.36, 41.76, 118.16, 84.47, −107.93, −21.17, 1.56, −81.47, 53.75, 127.24, −36.24, 77.45, 40.91, 40.84, 15.31, −46.99, 148.05, 6.73, −73.33, 4.79, −18.51, 47.86, −57.72, 126.61, 42.80, 45.03, −4.94, −25.23, 7.72, 75.97, −26.34, −23.47, −162.3809, −85.06, 79.88, −73.71, −83.19, −54.78, 23.58, 73.63, 31.03, −152.61, −52.67
−12.56, 27.24, 101.89, 75.58, 19.99, 42.89, −94.54, 118.91, −106.11, −81.41, −30.06, 73.28, −15.15, 58.54, 63.97, −120.46, 40.53, −85.81, −94.97, 93.78, 39.12, 30.53, −103.37, −73.25, −106.91, 51.89, −0.0666, 134.15, 11.27, 57.54, −159.43, −15.24, −65.16, −9.36, 14.69, −18.38, −104.09, −129.91, 33.44, 45.09, −142.51, −7.95, −92.21, 6.39, −9.22, −70.47, 108.70, −101.62, 4.13, 149.12, −29.43, 121.76, 39.69, −101.02, 161.77, −70.84, −3.5138, −27.43, −25.42, −159.27, −157.34, 131.01, 97.66, 75.76, 14.23, 2.08, 0.57, −39.68, −143.96, 161.57, 109.11, −20.83, −60.65, 174.55, −10.67, −3.21, 172.61, 7.96, −44.35, 110.75, 42.41, 158.99, 55.84, −142.47, −6.7307, 158.02, 177.35, 63.57, −71.64, −10.23, −30.08, 74.24, 59.42, −23.46, −59.76, −90.7065, 113.90, −109.11, −47.12

(d) 2EWH

Fig. 7. Visual representations of minimum energy configurations for each protein sequence by ELAMO.

V. Conclusions

In this paper, an Elitist Animal Migration Optimization (ELAMO) is fitted to optimize the structure of protein sequences with 3D AB off-lattice model. According to the reformed structure of ELAMO, rather than following the neighbors, only the group leaders are followed. This movement results with high solution quality and the elimination of animals during the migration process makes the algorithm not to trap in local optimum points. To enable a more accurate comparison, standard AMO and effective algorithms are included in our experiment. Even though ELAMO has not been specially designed for solving PSP problem, its effectiveness in modelling of real protein sequences is successful.

ELAMO eliminates the animals whose positions are not adequate even in the early stages of optimization and the rest of the animals in the herd only can follow their leaders. This elimination brings faster convergence rate without trapping into local optimum points in the process. It is known that the optimization is built on consecutive iterative processes until the termination criterion is obtained. During this process, some individuals may be eliminated due to their undesired characteristics which may be improved in the latter iterations. In ELAMO, the elimination is performed even at the first level of iterations and the desired characteristics are also eliminated. It is important to keep in mind that this may bring low solution quality as the length of the protein sequence increases.

Also, when the problem complexity increases in terms of large number of amino acids, the algorithm's performance may not be as efficient as it is expected. It is because of the lack of proper design for PSP problem. In the future, ELAMO may need to be strengthened by adding some boosting steps to be efficient in solving more complex protein sequences.

References

[1] Sahu, S. S., Panda, G. A. Novel Feature Representation Method Based on Chou's Pseudo Amino Acid Composition for Protein Structural Class Prediction. Comput. Biol. Chem. 2010, 34 (5–6), 320–327. https://doi.org/10.1016/j.compbiolchem.2010.09.002.

[2] Dehzangi, A., Paliwal, K., Lyons, J., Sharma, A., Sattar, A. A. Segmentation-Based Method to Extract Structural and Evolutionary Features for Protein Fold Recognition. IEEE/ACM Trans. Comput. Biol. Bioinform. 2014, 11 (3), 510–519. https://doi.org/10.1109/tcbb.2013.2296317.

[3] Anfinsen, C. B. Principles That Govern the Folding of Protein Chains. Science 1973, 181 (4096), 223–230. https://doi.org/10.1126/science.181.4096.223.

[4] Dill, K. A. Theory for the Folding and Stability of Globular Proteins. Biochemistry 1985, 24 (6), 1501–1509. https://doi.org/10.1021/bi00327a032.

[5] Chen, M., Huang, W.-Q. A Branch and Bound Algorithm for the Protein Folding Problem in the HP Lattice Model. Genomics Proteomics Bioinformatics 2005, 3 (4), 225–230. https://doi.org/10.1016/s1672-0229(05)03031-7.

[6] Istrail, S., Lam, F. Combinatorial Algorithms for Protein Folding in Lattice Models: A Survey of Mathematical Results. Commun. Inf. Syst. 2009, 9 (4), 303–346. https://doi.org/10.4310/cis.2009.v9.n4.a2.

[7] Hart, W. E., Newman, A. Protein Structure Prediction with Lattice Models. Handb. Mol. Biol. 2006, 1–24. https://doi.org/10.1201/9781420036275-45.

[8] Stillinger, F. H., Head-Gordon, T., Hirshfeld, C. L. Toy Model for Protein Folding. Phys. Rev. E 1993, 48 (2), 1469. https://doi.org/10.1103/physreve.48.1469.

[9] Bryant, S. H., Lawrence, C. E. An Empirical Energy Function for Threading Protein Sequence through the Folding Motif. Proteins Struct. Funct. Bioinforma. 1993, 16 (1), 92–112. https://doi.org/10.1002/prot.340160110.

[10] Levitt, M., Warshel, A. Computer Simulation of Protein Folding. Nature 1975, 253 (5494), 694–698. https://doi.org/10.1038/253694a0.

[11] Li, X., Zhang, J., Yin, M. Animal Migration Optimization: An Optimization Algorithm Inspired by Animal Migration Behavior. Neural Comput. Appl. 2014, 24, 1867–1877. https://doi.org/10.1007/s00521-013-1433-8.

[12] Ülker, E. An Elitist Approach for Solving the Traveling Salesman Problem Using an Animal migration Optimization Algorithm. Turk. J. Electr. Eng. Comput. Sci. 2018, 26 (1), 605–617. https://doi.org/10.3906/elk-1705-61.

[13] Li, B., Lin, M., Liu, Q., Li, Y., Zhou, C. Protein Folding Optimization Based on 3D Off-Lattice Model via an Improved Artificial Bee Colony Algorithm. J. Mol. Model. 2015, 21, 1–15. https://doi.org/10.1007/s00894-015-2806-y.

[14] Kalegari, D. H., Lopes, H. S. An Improved Parallel Differential Evolution Approach for Protein Structure Prediction Using Both 2D and 3D Off-Lattice Models. In 2013 IEEE Symposium on Differential Evolution (SDE), IEEE, 2013, 143–150. https://doi.org/10.1109/sde.2013.6601454.

[15] Rakhshani, H., Idoumghar, L., Lepagnot, J., Brévilliers, M. Speed up Differential Evolution for Computationally Expensive Protein Structure Prediction Problems. Swarm Evol. Comput. 2019, 50, 100493. https://doi.org/10.1016/j.swevo.2019.01.009.

[16] Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A. Improved Protein Structure Prediction Using Potentials from Deep Learning. Nature 2020, 577 (7792), 706–710. https://doi.org/10.1038/s41586-019-1923-7.

[17] Schauperl, M., Denny, R.A. AI-Based protein structure prediction in drug discovery: Impacts and challenges. J. Chem. Inf. Modell, 2022, 62(13), 3142-3156. https://doi.org/10.1021/acs.jcim.2c00026.

[18] Chowdhury, R., et al. Single-sequence protein structure prediction using a language model and deep learning. Nature Biotech. 2022, 40(11), 1617-1623. https://doi.org/10.1038/s41587-022-01556-z.

[19] Weißenow, K., Heinzinger, M., Rost, B. Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction. Structure, 2022, 30(8), 1169-1177. https://doi.org/10.1016/j.str.2022.05.001.

[20] Pelta, D. A., Krasnogor, N. Multimeme Algorithms Using Fuzzy Logic Based Memes for Protein Structure Prediction. Recent Adv. Memetic Algorithms 2005, 49–64. https://doi.org/10.1007/3-540-32363-5_3.

[21] Lin, X., Zhang, X. Protein Folding Structure Optimization Based on GAPSO Algorithm in the Off-Lattice Model. In 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2014, 43–49. https://doi.org/10.1109/bibm.2014.6999246.

[22] Boiani, M., Parpinelli, R. S. A GPU-Based Hybrid JDE Algorithm Applied to the 3D-AB Protein Structure Prediction. Swarm Evol. Comput. 2020, 58, 100711. https://doi.org/10.1016/j.swevo.2020.100711.

[23] Irbäck, A., Peterson, C., Potthast, F., Sommelius, O. Local Interactions and Protein Folding: A Three-Dimensional off-Lattice Approach. J. Chem. Phys. 1997, 107 (1), 273–282. https://doi.org/10.1063/1.474357.

[24] Kyte, J., Doolittle, R. F. A Simple Method for Displaying the Hydropathic Character of a Protein. J. Mol. Biol. 1982, 157 (1), 105–132. https://doi.org/10.1016/0022-2836(82)90515-0.

[25] Wolpert, D. H., Macready, W. G. No Free Lunch Theorems for Optimization. IEEE Trans. Evol. Comput. 1997, 1 (1), 67–82. https://doi.org/10.1109/4235.585893.

[26] Hsu, H.P., Mehra, V., Grassberger, P. Structure Optimization in an Off-Lattice Protein Model. Phys. Rev. E 2003, 68 (3), 037703. https://doi.org/10.1103/physreve.68.037703.

[27] Zhu, H., Pu, C., Lin, X., Gu, J., Zhang, S., Su, M. Protein Structure Prediction with EPSO in Toy Model. In 2009 Second International Conference on Intelligent Networks and Intelligent Systems, IEEE, 2009, pp 673–676. https://doi.org/10.1109/icinis.2009.172.

[28] Li, B., Li, Y., Gong, L. Protein Secondary Structure Optimization Using an Improved Artificial Bee Colony Algorithm Based on AB Off-Lattice Model. Eng. Appl. Artif. Intell. 2014, 27, 70–79. https://doi.org/10.1016/j.engappai.2013.06.010.

[29] Bošković, B., Brest, J. Differential Evolution for Protein Folding Optimization Based on a Three-Dimensional AB off-Lattice Model. J. Mol. Model. 2016, 22, 1–15. https://doi.org/10.1007/s00894-016-3104-z.

[30] Lin, J., Zhong, Y., Li, E., Lin, X., Zhang, H. Multi-Agent Simulated Annealing Algorithm with Parallel Adaptive Multiple Sampling for Protein Structure Prediction in AB off-Lattice Model. Appl. Soft Comput. 2018, 62, 491–503. https://doi.org/10.1016/j.asoc.2017.09.037.

[31] Jana, N. D., Sil, J., Das, S. Selection of Appropriate Metaheuristic Algorithms for Protein Structure Prediction in AB Off-Lattice Model: A Perspective from Fitness Landscape Analysis. Inf. Sci. 2017, 391, 28–64. https://doi.org/10.1016/j.ins.2017.01.020.

Improved Multiclass Brain Tumor Detection using Convolutional Neural Networks and Magnetic Resonance Imaging

Mohamed Amine Mahjoubi[1], Soufiane Hamida[2], Oussama El Gannour[3], Bouchaib Cherradi[4], Ahmed El Abbassi[5], Abdelhadi Raihani[6]
EEIS Laboratory, ENSET of Mohammedia, Hassan II University of Casablanca, Mohammedia 28830, Morocco[1, 2, 3, 4, 6]
GENIUS Laboratory, SupMTI of Rabat, Rabat, Morocco[2]
STIE Team, CRMEF Casablanca-Settat, Provincial Section of El Jadida, El Jadida 24000, Morocco[4]
ERTTI Laboratory, FST of Errachidia, My Ismail University, Errachidia, Morocco[5]

***Abstract*—Recently, Deep learning algorithms, particularly Convolutional Neural Networks (CNNs), have been applied extensively for image recognition and classification tasks, with successful results in the field of medicine, such as in medical image analysis. Radiologists have a hard time categorizing this lethal illness since brain tumors include a variety of tumor cells. Lately, methods based on computer-aided diagnostics claimed to employ magnetic resonance imaging to help with the diagnosis of brain cancers (MRI). Convolutional Neural Networks (CNNs) are often used in medical image analysis, including the detection of brain cancers. This effort was motivated by the difficulty that physicians have in appropriately detecting brain tumors, particularly when they are in the early stages of brain bleeding. This proposed model categorized the brain image into four distinct classes: (Normal, Glioma, Meningioma, and Pituitary). The proposed CNN networks reach 95% of recall, 95.44% accuracy and 95.36% of F1-score.**

Keywords—Deep learning; convolutional neural networks; brain tumor; classification; magnetic resonance imaging

I. INTRODUCTION

A brain tumor develops as a result of the overproduction and proliferation of cells in the skull. The body's command center, the brain, can be burdened by tumors, which can also be harmful to a person's health [1]. It has been reported in the study that brain tumors account for between 85% and 90% of all major Central Nervous System (CNS) tumors [2]. Radiologists have extensively used the medical imaging technique for tumor detection [3], [4]. Among the current medicalities, MRI is the method of choice for brain tumors because of its astronomical nature. Radiologists routinely manually detect brain cancers. Depending on the radiologist's level of training and experience, the tumor grading process can take a while. The interpretation is expensive and wrong. The associated challenges are attributed to specific traits, such as the substantial variation in form, dimensions, and magnitude for the same tumor type. Additionally, various diseases have similar appearances [5], [6]. A successful Computer Aided Diagnosis (CAD) system requires the development of feature extraction [7]. This is a challenging process that necessitates prior knowledge of the domain problem because the accuracy of the classification depends on the correctly extracted features. Since DL is a subset of machine learning, it does not use any manually created features [8]. In several disciplines, the use of DL and ML for segmentation, detection, prediction, classification and early diagnosis using medical data has been promoted [9]–[25]. ML and DL as fields of Artificial Intelligence (AI) find their applications in many others field such as handwritten recognition and natural language processing [26]–[29].

Deep Learning (DL) is a subfield of Machine Learning (ML) that uses artificial neural networks with multiple layers to model complex patterns and relationships in data [30], [31]. Image classification is a task in computer vision where a model is trained to identify and categorize objects in images [32], [33]. In the context of brain tumors, convolutional neural networks (CNNs) are often used for image classification [34]. A CNN is a type of neural network specifically designed for image processing and analysis. It uses convolutional layers to detect local patterns and features in images, allowing it to learn and classify objects within an image [35]. In the case of brain tumor classification, a CNN is trained on a large dataset of medical images of brains, where each image is labeled as containing a tumor or not. The CNN then uses the features it has learned to classify new, unseen images as containing a tumor or not. This can aid for the early diagnosis and treatment of brain tumors.

CNN was initially utilized in 1980 [36], [37]. In essence, it is a disguised Multilayer Perceptron (MLP) network. CNN's computational capacity is based on a model of the human brain. Humans use an object's visual appearance to detect and identify it. Tens of thousands of photos of the same item are used to educate our kids how to distinguish objects. This aids a youngster in recognizing or foreseeing things that they have never encountered before. Similar in operation, CNN is well recognized for processing images.

In this paper, we proposed CNN-based model for medical image analysis, which is a sophisticated algorithm that utilizes DL techniques to categorize brain images into four distinct classes: Normal, Glioma, Meningioma, and Pituitary. These four classes encompass the most common types of brain tumors and are critical in providing accurate and precise diagnosis and treatment. The model's impressive performance metrics, including a recall rate of 95%, an accuracy of 95.44%, and an F1-score of 95.36%, demonstrate the model's ability to detect brain tumors with high precision and accuracy. The recall rate, which is a measure of the model's ability to correctly identify all positive cases, is at an impressive 95%, indicating that the model has a low false-negative rate. Additionally, the accuracy score, which measures the model's ability to classify the images correctly, is also at a remarkable 95.44%. Finally, the F1-score, which is a combined measure of the model's precision and recall, is at a remarkable 95.36%, indicating the model's overall performance in categorizing brain images. These high-performance metrics are critical in medical image analysis as they help physicians diagnose and treat brain tumors accurately and efficiently. The proposed model is an essential tool for radiologists and physicians, as it reduces the subjectivity involved in manually interpreting medical images and provides a reliable and objective method for diagnosis.

The remainder of this article is structured as follows: Section II explores the related works to the search for brain tumor detection. In Section III, the proposed CNN-based model is presented, including its architecture, training methodology, and evaluation metrics. Section IV provides an overview of the materials and methods utilized in the study. Section V presents the experimental results, and in Section VI, the study's findings and conclusions are discussed.

II. Related Works

DL and AI play a crucial role in MRI image processing through techniques such as segmentation, recognition, and categorization. They are also employed in the classification and detection of brain cancer. Numerous studies have been conducted on the identification and segmentation of brain MRI images. A review of international literature was conducted to assess the use of DL in identifying and categorizing brain tumors.

In [38], the authors used newly designed CapsNets to allow CapsNet to access neighboring tissues while remaining focused on the core target. As a result, a modified CapsNet architecture for brain tumor classification is presented, with coarse tumor boundaries incorporated as extra inputs into its pipeline to increase CapsNet's focus. The proposed method outperforms its competitors significantly.

The authors in [39] used a convolutional neural network to perform multimodal brain tumor categorization for early diagnosis (CNN). With an accuracy of 92.66 %, the CNN model can categorize brain cancers into 5 types (Normal, Glioma, Meningioma, Pituitary, and Metastatic). The grid search optimization approach is used to automatically define critical hyperparameters in CNN models. The suggested CNN model is compared to popular cutting-edge CNN models such as AlexNet, Inceptionv3, ResNet-50, VGG-16, and GoogleNet. Using huge publicly available clinical datasets, satisfactory classification results are produced. This methodology has a number of drawbacks that may be listed as follows, despite the fact that the recommended methods for classifying brain tumors differ. The accuracy provided by current techniques is unsatisfactory due to the significance of MRI classification in the medical field. Some classification algorithms could not be fully automated since they required a human to manually identify tumor locations.

The authors in [40] described a technique for improving classification performance. To put the suggested method to the test on a big dataset, the authors employ three feature extraction methods: intensity histogram, gray level co-occurrence matrix (GLCM), and bag-of-words model (BoW). Using the enlarged tumor region as the ROI enhances the intensity histogram, GLCM, and BoW model accuracies by 82.31% vs. 71.39%, 84.75% vs. 78.18%, and 88.19% vs. 83.54%, respectively. Ring partitioning can improve accuracy by up to 87.54%, 89.72%, and 91.28% in addition to increasing region. These experimental findings demonstrate that the proposed strategy for classifying brain cancers in T1-weighted CE-MRI is both possible and effective.

In [41], the authors suggested a method for classifying brain tumors utilizing a set of deep characteristics and ML classifiers. In the proposed framework, they get deep characteristics by brain magnetic resonance (MR) pictures using the notion of transfer learning and numerous pre-trained deep convolutional neural networks. The top three deep features for multiple ML classifiers are chosen and concatenated into a deep feature set, which is then fed through multiple ML classifiers to predict the final output.

The authors in [42] proposed a method to improve the segmentation of brain tumors in magnetic resonance imaging (MRI) by incorporating an additional classification network. The segmentation is performed using a convolutional neural network (CNN) which is trained on a dataset of MRI images with corresponding tumor masks. The CNN is then combined with a classification network that is trained to distinguish between different types of tumors. The output of the classification network is used to refine the segmentation results by identifying and removing false positives. The authors evaluate their method on two publicly available datasets and compare it to other state-of-the-art methods. They report that their method achieves higher accuracy and Dice similarity coefficient (DSC) scores than the other methods. They also perform ablation studies to analyze the contribution of the classification network and find that it significantly improves the segmentation results.

In [43], the authors provide an overview of the challenges in brain tumor segmentation and discuss how deep learning techniques have been applied to address these challenges. The survey covers various types of deep learning models, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs). The authors also discuss the advantages and limitations of these methods and provide insights into future directions for research.

III. The Proposed CNN-Based Model

The structure of a CNN-based model can be customized and optimized by selecting specific hyperparameters. These hyperparameters play a crucial role in determining the overall architecture of the model, including the number of convolutional layers, the activation functions used in each layer, and the number of hidden units per layer. The number of convolutional layers determines the depth of the neural network and its ability to extract high-level features from the input image. The more convolutional layers the model has, the more complex and sophisticated features it can extract, leading to better performance. However, adding too many layers can also result in overfitting, where the model becomes too specialized to the training data and performs poorly on new data.

The choice of activation functions also affects the model's ability to extract features accurately. Activation functions introduce non-linearities to the model, allowing it to model complex relationships between the input and output. Common activation functions used in CNNs include ReLU, sigmoid, and tanh, among others. The choice of activation function can significantly impact the model's performance, and selecting the right activation function is critical in optimizing the model's accuracy. After extensive experimentation, we arrived at the optimal selection of hyperparameters for the multi-layered model depicted in Fig. 1. This is chosen in: Five layers of convolution2D that differ in the number of filters, so that it doubles as you go deeper, starting with the first layer, which contains 32 filters, and ending with the last layer, which contains 512 filters, all to extract many features; five layers of max pooling in order to extract important information from the previous convolution2D layer; flattening layer, to render the information in one dimension; then a Dense layer with 128 units; finally, the Dance layer with 4 units, due to the number of final classes was used in which the SoftMax function was used because it is the most used function in multi-class models (in our case 4 classes). In all these layers, the ReLU activation function was used because it is the most advanced compared to the other functions. Fig. 2 displays the layout of our proposed CNN architecture.

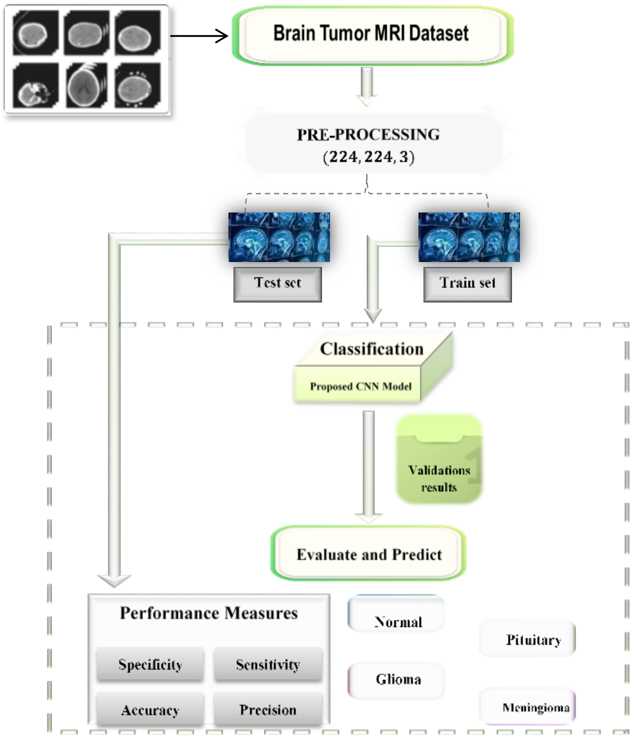


Fig. 1. Flowchart of a System for Detecting Brain Tumors.

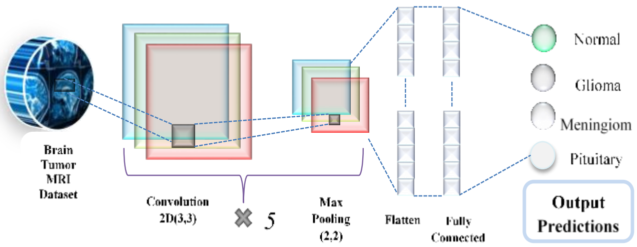


Fig. 2. The proposed CNN model architecture.

Brain tumor detection is a critical problem in medical imaging, and a Convolutional Neural Network (CNN) can be used as a detection system. A CNN is a type of DL algorithm that is designed to analyze image data and can be trained to recognize specific features in the images. In the case of brain tumor detection, a CNN is trained using a large dataset of medical images that includes both normal and abnormal scans. The CNN is then used to identify and locate areas of the brain that may contain a tumor. This is done by processing the images and identifying certain patterns and features that are characteristic of brain tumors. The output of the CNN can be used to assist medical professionals in the diagnosis of brain tumors and to guide further testing and treatment. The use of a CNN as a brain tumor detection system has the potential to improve the accuracy and speed of diagnosis, which can be crucial in treating brain tumors effectively.

IV. Materials and Methods

A. Dataset Collection

The "Brain Tumor MRI Dataset" on Kaggle is a collection of magnetic resonance imaging (MRI) scans of the brain. The dataset includes MRI scans of both healthy individuals and individuals with brain tumors, which have been annotated by medical experts. The annotations indicate the presence, location, and type of brain tumor in the MRI scans. In Figure 3, sample images from the dataset are depicted.

In fact, a publicly available Kaggle database is utilized, as is described in this section, which also provides some information on the database. The following three datasets were combined to generate this one: Figshare1, SARTAJ dataset2, Br35H3.

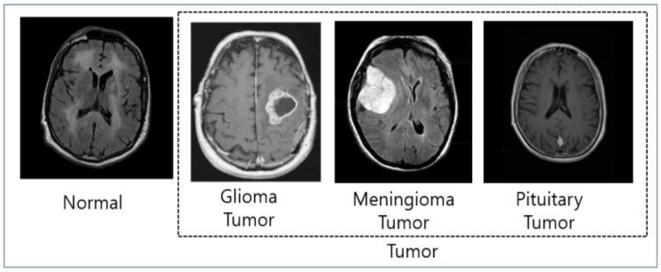


Fig. 3. Some sample images of the dataset.

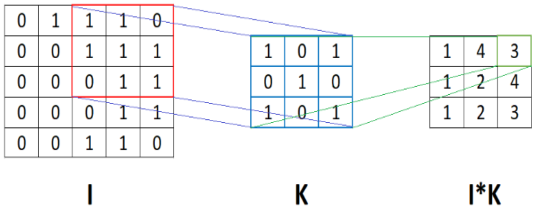
[1] https://figshare.com/articles/dataset/brain_tumor_dataset/1512427
[2] https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification
[3] https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection

The 7022 MRI scans of the human brain in this dataset are divided into four classes. The dataset can be used for a variety of research purposes, including developing and evaluating computer vision algorithms for brain tumor detection and classification, as well as for training ML models for medical diagnosis and treatment planning. However, it is important to note that the generalizability of the models developed using this dataset should be carefully evaluated, as it is based on a limited sample of MRI scans.

B. Background on CNN

DL models use a hierarchical framework to learn high-level abstractions from input images [44]. Due to the availability of large-scale annotated datasets and the fact that CNN has demonstrated to be the most effective DL method for assessing medical pictures. The well-known CNN models [45] ImageNet, AlexNet, VGG16, GoogLeNet, Inception-V3 and ResNet101 have made significant strides in image recognition. However, the area of medical imaging lacks a comparable annotated dataset. Medical image classification using CNN is frequently done using one of two methods [46]. The first is called "learning from the ground up," while the second is called "Transfer Learning." The network layers that make up the CNN include a convolution layer, an activation layer, a Maxpooling layer, and a classification layer. Each of these levels is explained as follows [47], [48].

1) Convolution layer: The convolutional layer comes first [7]. This layer is in charge of identifying an input word's attributes. This step merely combines the entry neuron by a filter depending on the input and requirement to produce the feature map [49]. A neural activation function is used to introduce nonlinearity. The animal visual cortex served as a model for CNN computing. It decodes visual data and has a fine sensitivity to the input's smallest subregions [50]. The convolutional layer's main elements are its receptive field, stride, dilation, and padding [51]. Fig. 4 displays the application of the convolutional layer.



$$I = \begin{array}{ccccc} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{array} \qquad K = \begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{array} \qquad I*K = \begin{array}{ccc} 1 & 4 & 3 \\ 1 & 2 & 4 \\ 1 & 2 & 3 \end{array}$$

Fig. 4. Convolution layer application.

2) Activation layer: The activation function is a crucial component of a Convolutional Neural Network (CNN) model as it defines the output of a neuron in response to a given input. The activation function determines the range of values that the neuron can output and allows the model to introduce non-linearity into the decision boundary, making it possible to learn complex relationships between the input and output variables. Common activation functions used in CNNs include the Rectified Linear Unit (ReLU), the Sigmoid function, and the Hyperbolic Tangent (Tanh) function. The choice of activation function depends on the specific problem being solved and can have a significant impact on the performance of the CNN. It's important to carefully choose the activation function in a CNN, as the wrong choice can lead to slow training and convergence, or even prevent the network from learning the desired patterns in the data [5].

3) Pooling layer: It comes after the convolution layer. This layer's task is to shrink the feature map, which implies fewer computations and parameters are required to operate the network. Therefore, it can be claimed that a summary of the features is the output of this layer. There are several techniques to pool data; in this instance, max pooling was utilized. The feature map's most objects that are covered by the filters are selected using max pooling [5].

4) Classification layer: In our CNN architecture, the classification layer comes last. This extensively used classifier which is a fully - connected feed-forward network. The neurons in the layers with complete connections are all connected to the neurons in the layer below. By merging the traits of the preceding layers, this layer recognizes the input image and predicts classes based on it. A total number of output classes is based on the number of classes in the source dataset. In this paper, the classification layer uses the "SoftMax" activation function to classify the features created from the input images in the previous layer into separate groups depending on the training data [5].

C. Confusion Matrix and Evaluation Metrics

A confusion matrix is a table used to evaluate the performance of a classification model by summarizing the predicted and actual class labels of a dataset. It displays the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for each class. In a binary classification problem, the confusion matrix is a 2x2 table with two rows and two columns representing the actual and predicted classes. The first row represents the actual negative and the second row represents the actual positive. The first column represents the predicted negative and the second column represents the predicted positive. For a multi-class classification problem, the confusion matrix is an n x n table, where n is the number of classes. The rows represent the actual classes and the columns represent the predicted classes. Each cell in the matrix represents the number of instances that belong to a particular actual class and a particular predicted class [52]. Based on the entries in the confusion matrix, several evaluation metrics can be computed to assess the performance of the classifier. Some commonly used evaluation metrics include:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

$$\text{F1-score} = 2\times\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{4}$$

$$\text{Precision} = \frac{\text{TN}}{\text{TN+FP}} \tag{5}$$

The results of a confusion matrix can be used to calculate various performance metrics, such as accuracy, precision, recall, and F1 score. A high number of true positives and true negatives indicate a high accuracy of the model, while a high number of false positives and false negatives indicate low accuracy. Precision indicates the proportion of positive predictions that are actually correct, while recall indicates the proportion of actual positive instances that were correctly predicted. The F1 score is the harmonic mean of precision and recall, and is a good overall indicator of a model's performance. In summary, a confusion matrix provides detailed information about the performance of a model, while evaluation metrics provide a concise, single value representation of the model's performance [53].

V. Experimental Results

A. Algorithms Best Parameters

The main objective of this project is to build a system capable of classifying medical images and giving correct decisions with a large percentage. All this depends on the quality of the existing data, the convolutional neural network, the number of its layers and coefficients, and the layers used to process the data needed to train the network.

Our study proposed a CNN architecture model; we retrieved the input 224×224 MRI image data with RGB color channels having a batch size of 32 by our CNN model. Initially, we added five convolutional layers; in addition, there are five max-pooling layers, one flattening layer, and activation (ReLU). The model develops the capacity to produce hierarchical qualities automatically by use of a succession of hidden layers. A final class label is determined by using a Softmax function on the outputs of this layer. In this proposed model, we have an output layer that generates a quadridimensional vector representing four different classifications of cerebral tumors. Table I displays the summary of our model CNN proposed architecture.

TABLE I. The Summary of Model Description

Layer (type)	Output (shape)	Parameters
Input layer	(224, 224, 3)	0
Conv2D	(224, 224, 32)	896
MaxPooling2D	(112,112,32)	0
Conv2D	(224,224,32)	18496
MaxPooling2D	(112,112,64)	0
Conv2D	(56,56,128)	73856
MaxPooling2D	(28,28,128)	0
Conv2D	(28,56,256)	295168
MaxPooling2D	(14,14,256)	0
Conv2D	(14,14,512)	1180160
MaxPooling2D	(7,7,512)	0
Flatten	(None, 25088)	0
Dense	(None, 128)	3211392
Dense	(None, 4)	516

This table describes the architecture of a CNN for image classification. The table lists each layer in the network, its type (e.g., Conv2D, MaxPooling2D, Dense, Flatten), the output shape of the layer, and the number of parameters in that layer. The network starts with an input layer that takes in an image of shape (224,224,3), which means the image has a height and width of 224 pixels and 3 color channels (representing RGB values). The next layer is a Conv2D layer with 32 filters, which means it will learn 32 different feature maps from the input image. The MaxPooling2D layer performs down-sampling by taking the maximum value in a region of the feature map and reduces its size. This process is repeated several times with increasing number of filters in the Conv2D layer and reducing the size of the feature map in the MaxPooling2D layer. Finally, the feature maps are flattened into a 1D vector and passed through two dense (fully connected) layers to output the final prediction. The numbers of parameters in each layer, along with the total number of trainable and non-trainable parameters are also listed. The total number of parameters in the network is 7,789,484, and all of them are trainable. To train the model, we need training data and for validation also we need test data. Once we are satisfied with the test result of the model, we can use it to make predictions on new data. In addition, for the properties and parameters of the model starting with the number of epochs =50 (epochs are the number of typical repetitions of the training).

To configure our training model, we need to call the compile method with the loss function we want to use. The type of optimization and the metrics our model should evaluate during training and testing we will use: The Adam optimizer is due to the fact that, as mentioned earlier, it is one of the most widely used optimizers because it is the fastest method and also converges quickly to correct for learning rate latency and high contrast. The categorical cross entropy loss function is used when there are multiple label classes. Metrics is a function used to evaluate the performance of your model.

The best parameters for algorithms in CNNS involve a variety of elements, such as learning size, batch size, optimiser, loss function, activation functions, and epochs. Different parameters have to be tuned based on the network architecture, dataset, and task. Table II presented the best parameters of our proposed model.

TABLE II. The Best Hyper-Parameters used for the TL Models in Training Phase

Network	Learning rate	Batch Size	Optimizer	Loss Function	Epochs
Proposed model	1.00e10-4	32	Adam	Categorical cross entropy	50

B. Training Results

Our experiment was based on a dataset of brain MRI images. We trained a CNN model to detect whether the MRI image contains a tumor (glioma, meningioma and pituitary tumor) or not. Finally, we compared the diagnostic and computational results of our model with the related works. On our training data set, our proposed model has a 95.44% accuracy rate. The graph depicts the accuracy and loss during the construction and validation phases of our proposed CNN model are presented in Fig. 5 and Fig. 6.

The graphs show the performance of the model in terms of accuracy and loss as it is being developed and tested. Accuracy refers to the measure of how well the model is able to correctly predict the outcomes or classes of the input data. It is usually expressed as a percentage, where higher values indicate better performance. Loss, on the other hand, is a measure of how much error there is between the predicted outputs of the model and the actual outputs. Lower values of loss indicate that the model is more accurate in its predictions.

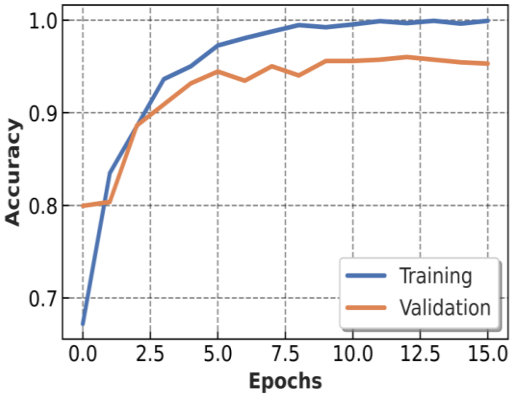


Fig. 5. Accuracy of the proposed model.

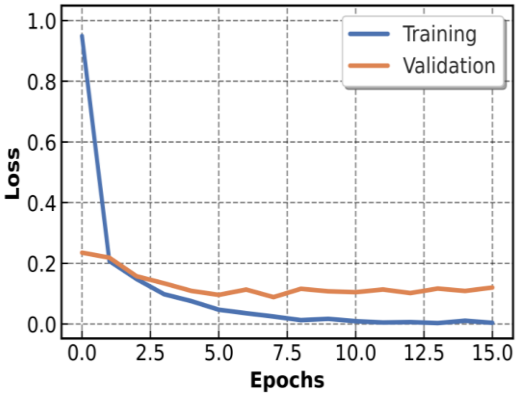


Fig. 6. Loss of the proposed model.

These graphs show the accuracy and loss of the CNN model at different stages of its development. The construction phase likely refers to the phase where the model is being trained on a dataset to learn the patterns and relationships in the data. The validation phase refers to the phase where the model is being tested on a separate set of data to evaluate its performance and generalizability.

C. Testing Results

The proposed model achieved an impressive accuracy of 95.44% in the detection of brain cancers. Table 4 presents the detection performance results of the model with an 80% train and 20% test split, showing the confusion matrix of the classifier model, which displays the number of correctly and incorrectly classified samples in each class. The confusion matrix, depicted in Figure 7, reveals the model's strong performance across all four classes: glioma, meningioma, no tumor, and pituitary. The ROC curve, as shown in Fig. 8, displays the true positive rate, also known as recall, against the false positive rate. The proposed model demonstrated excellent performance on the brain MRI dataset, accurately classifying a high number of samples in each class. For instance, 148 out of 156 samples in the glioma class and 193 out of 195 samples in the no tumor class were correctly classified. Overall, the results demonstrate the model's effectiveness in accurately detecting brain cancers, particularly in the early stages of brain bleeding. However, there are also some misclassified samples, as shown in the entries in the off-diagonal cells of the matrix. For example, five samples from the glioma class were misclassified as meningioma, and 3 samples from the pituitary class were misclassified as no tumor.

Based on this confusion matrix, it is possible to compute various evaluation metrics such as accuracy, precision, recall, and F1-score presented in Table III to quantify the performance of the proposed model. These metrics can provide a more comprehensive understanding of the model's performance and help identify areas for improvement.

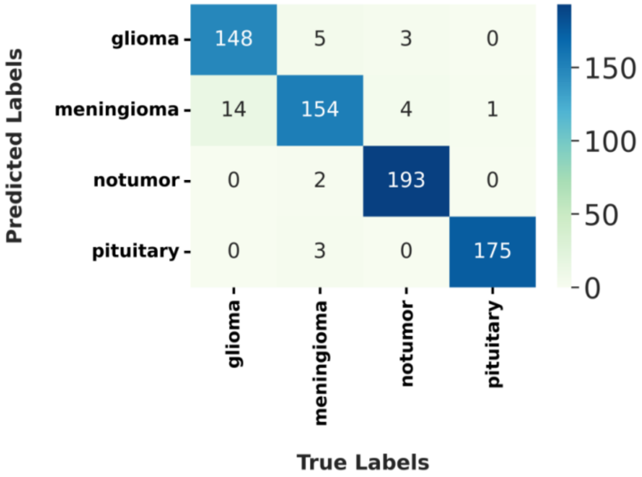


Fig. 7. The confusion matrix of the proposed model.

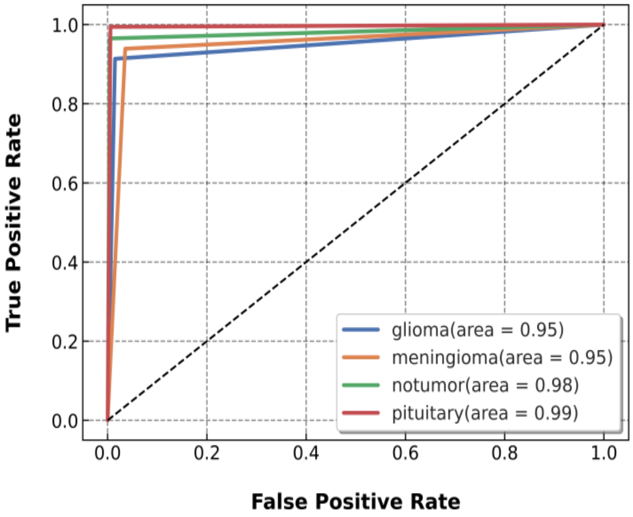


Fig. 8. ROC curves findings for the proposed model.

TABLE IV. CLASSIFICATION REPORT OF THE PROPOSED MODEL

Classes	Precision	Recall	F1-score	Support
Glioma	0.91	0.95	0.93	156
Meningioma	0.94	0.89	0.95	173
Notumor	0.96	0.99	0.98	195
Pituitary	0.99	0.98	0.99	178

This table represents the evaluation metrics of a classifier model that has been trained on a medical imaging dataset with four different classes: Glioma, Meningioma, No tumor, and Pituitary. The metrics shown are precision, recall, F1-score, and support. Precision is the fraction of true positive predictions among all positive predictions, recall is the fraction of true positive predictions among all actual positive cases, F1-score is the harmonic mean of precision and recall, and support is the number of samples in each class. The results show that the model performs well in all classes with F1-scores ranging from 0.93 to 0.99, indicating that the model has good precision and recall values.

D. Discussion

The proposed CNN model achieved the highest accuracy rate among other deep neural models for classifying cerebral tumors into four classes on MRI images. This model has the potential to aid medical professionals in diagnosing brain tumors accurately and efficiently. The architecture of the model is designed for optimal performance on medical image datasets, making it a promising tool in the field of medical image analysis. Comparison with other studies that used various techniques, such as capsule networks, data augmentation and partitioning, and feature sets with ML classifiers, showed that DCNNs are the most effective technique for classifying brain tumors based on MRI scans. The accuracy of the models ranged from 90.89% to 95.44%, indicating the effectiveness of deep learning techniques in this domain. However, it is essential to note that the generalizability of these models to other datasets and clinical scenarios should be evaluated with care. As with any ML model, its effectiveness is dependent on the quality and size of the dataset used to train and validate it. Thus, the performance of the proposed model should be assessed on other datasets, and it should be tested under different clinical scenarios before being deployed in real-world settings. The proposed method is also applicable for different MRI classifications, and it can be combined with transfer learning techniques to achieve more accurate classifications in the future. Transfer learning can help reduce the amount of data needed to train a model, and it can improve the accuracy of the model by leveraging the knowledge learned from other datasets.

While the proposed CNN model for classifying cerebral tumors based on MRI images appears to be highly effective, there are several limitations to consider. First and foremost, the model's accuracy and effectiveness may be limited to the specific dataset used to train and validate it. To ensure its generalizability, it is crucial to evaluate the model on other datasets and clinical scenarios. Another limitation of the proposed model is that it may not be suitable for classifying other types of tumors or medical conditions. It is important to recognize that the model was specifically designed and optimized for classifying cerebral tumors based on MRI images, and it may not be effective in other medical image analysis tasks. Moreover, like any deep learning model, the proposed CNN model requires a significant amount of data to be trained effectively. While the model achieved high accuracy rates on the specific dataset used, it may not perform as well on smaller or lower-quality datasets. Additionally, the model may be computationally expensive, which may limit its practical application in certain settings. Table IV shows the comparison of the proposed model's performance with other methods based on classification precision. Our proposed model achieved the highest accuracy of 95.44% compared to previous work in this field, which shows the superior quality of the model.

TABLE V. COMPARISON WITH PREVIOUS WORK

Authors	Method/ Brain tumor data	Accuracy (%)
[42]	Segmentation using a CNN	89.99
[38]	Capsule networks/MRI images	90.89
[54]	CNN + Augmenting/ MRI images	91.28
[39]	CNN Multi-classifying / MRI images	92.66
[41]	Deep feature + ML classifiers/ MRI images	93.72
[43]	CNN, RNN and GANs	95
This work	**CNN Multi-classifying/MRI images**	**95.44**

VI. CONCLUSIONS AND PERSPECTIVES

To summarise, the proposed CNN model for classifying cerebral tumors based on MRI images is a promising tool for aiding medical professionals in diagnosing brain tumors accurately and efficiently. The high accuracy rate achieved by the proposed model compared to other deep neural models indicates the effectiveness of DCNNs in classifying brain tumors based on MRI scans. The proposed model's architecture is designed for optimal performance on medical image datasets, making it a promising tool in the field of medical image analysis. However, it is essential to recognize the limitations of the proposed model, such as the need to carefully evaluate its generalizability to other datasets and clinical scenarios, as well as the potential computational cost and data requirements. Future research should focus on developing more robust and accurate models by exploring alternative approaches and using larger and more diverse datasets.

Additionally, the proposed model's applicability for different MRI classifications and potential incorporation of other types of medical imaging data, such as CT scans, PET scans, or MRS data, offer exciting possibilities for improving the accuracy and effectiveness of medical image analysis. The development of more sophisticated models and techniques will lead to more accurate and efficient diagnosis and treatment of cerebral tumors, ultimately benefiting patients and medical professionals alike.

Our future research will focus on developing more robust and accurate models by exploring alternative approaches and using larger and more diverse datasets. It may also be worth investigating the potential for incorporating other types of medical imaging data.

REFERENCES

[1] F. J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, and D. González-Ortega, “A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network,” Healthcare, vol. 9, no. 2, p. 153, Feb. 2021, doi: 10.3390/healthcare9020153.

[2] M. L. Bondy et al., “Brain tumor epidemiology: Consensus from the Brain Tumor Epidemiology Consortium,” Cancer, vol. 113, no. S7, pp. 1953–1968, Oct. 2008, doi: 10.1002/cncr.23741.

[3] Y. Zhang, A. Li, C. Peng, and M. Wang, “Improve Glioblastoma Multiforme Prognosis Prediction by Using Feature Selection and Multiple Kernel Learning,” IEEE/ACM Trans. Comput. Biol. and Bioinf., vol. 13, no. 5, pp. 825–835, Sep. 2016, doi: 10.1109/TCBB.2016.2551745.

[4] R. L. Siegel, K. D. Miller, and A. Jemal, “Cancer statistics, 2015: Cancer Statistics, 2015,” CA: A Cancer Journal for Clinicians, vol. 65, no. 1, pp. 5–29, Jan. 2015, doi: 10.3322/caac.21254.

[5] P. Tiwari et al., “CNN Based Multiclass Brain Tumor Detection Using Medical Imaging,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–8, Jun. 2022, doi: 10.1155/2022/1830010.

[6] Department of Computer Applications, National Institute of Technology, Tiruchirapalli, TamilNadu, India, Kavitha. S, and P. J. A.Alphonse, “A Hybrid Cryptosystem to Enhance Security in IoT Health Care System,” IJWMT, vol. 9, no. 1, pp. 1–10, Jan. 2019, doi: 10.5815/ijwmt.2019.01.01.

[7] E.-S. A. El-Dahshan, H. M. Mohsen, K. Revett, and A.-B. M. Salem, “Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm,” Expert Systems with Applications, vol. 41, no. 11, pp. 5526–5545, Sep. 2014, doi: 10.1016/j.eswa.2014.01.021.

[8] P. Tiwari et al., “CNN Based Multiclass Brain Tumor Detection Using Medical Imaging,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–8, Jun. 2022, doi: 10.1155/2022/1830010.

[9] B. Cherradi, O. Terrada, A. Ouhmida, S. Hamida, A. Raihani, and O. Bouattane, “Computer-Aided Diagnosis System for Early Prediction of Atherosclerosis using Machine Learning and K-fold cross-validation,” in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, Jul. 2021, pp. 1–9. doi: 10.1109/ICOTEN52080.2021.9493524.

[10] K. Usman and K. Rajpoot, “Brain tumor classification from multi-modality MRI using wavelets and machine learning,” Pattern Anal Applic, vol. 20, no. 3, pp. 871–881, Aug. 2017, doi: 10.1007/s10044-017-0597-8.

[11] S. Hamida, O. El Gannour, B. Cherradi, A. Raihani, H. Moujahid, and H. Ouajji, “A Novel COVID-19 Diagnosis Support System Using the Stacking Approach and Transfer Learning Technique on Chest X-Ray Images,” Journal of Healthcare Engineering, vol. 2021, pp. 1–17, Nov. 2021, doi: 10.1155/2021/9437538.

[12] O. Terrada, B. Cherradi, A. Raihani, and O. Bouattane, “Atherosclerosis disease prediction using Supervised Machine Learning Techniques,” in 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, Apr. 2020, pp. 1–5. doi: 10.1109/IRASET48871.2020.9092082.

[13] D. Lamrani, B. Cherradi, O. E. Gannour, M. A. Bouqentar, and L. Bahatti, “Brain Tumor Detection using MRI Images and Convolutional Neural Network,” IJACSA, vol. 13, no. 7, 2022, doi: 10.14569/IJACSA.2022.0130755.

[14] S. Laghmati, B. Cherradi, A. Tmiri, O. Daanouni, and S. Hamida, “Classification of Patients with Breast Cancer using Neighbourhood Component Analysis and Supervised Machine Learning Techniques,” in 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, Sep. 2020, pp. 1–6. doi: 10.1109/CommNet49926.2020.9199633.

[15] O. El Gannour et al., “Concatenation of Pre-Trained Convolutional Neural Networks for Enhanced COVID-19 Screening Using Transfer Learning Technique,” Electronics, vol. 11, no. 1, p. 103, Dec. 2021, doi: 10.3390/electronics11010103.

[16] H. Moujahid, B. Cherradi, O. E. Gannour, L. Bahatti, O. Terrada, and S. Hamida, “Convolutional Neural Network Based Classification of Patients with Pneumonia using X-ray Lung Images,” Adv. sci. technol. eng. syst. j., vol. 5, no. 5, pp. 167–175, 2020, doi: 10.25046/aj050522.

[17] H. Moujahid, B. Cherradi, and L. Bahatti, “Convolutional Neural Networks for Multimodal Brain MRI Images Segmentation: A Comparative Study,” in Smart Applications and Data Analysis, vol. 1207, M. Hamlich, L. Bellatreche, A. Mondal, and C. Ordonez, Eds. Cham: Springer International Publishing, 2020, pp. 329–338. doi: 10.1007/978-3-030-45183-7_25.

[18] H. Moujahid, B. Cherradi, M. Al-Sarem, and L. Bahatti, “Diagnosis of COVID-19 Disease Using Convolutional Neural Network Models Based Transfer Learning,” in Innovative Systems for Intelligent Health Informatics, vol. 72, F. Saeed, F. Mohammed, and A. Al-Nahari, Eds. Cham: Springer International Publishing, 2021, pp. 148–159. doi: 10.1007/978-3-030-70713-2_16.

[19] O. Terrada, A. Raihani, O. Bouattane, and B. Cherradi, “Fuzzy cardiovascular diagnosis system using clinical data,” in 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, Apr. 2018, pp. 1–4. doi: 10.1109/ICOA.2018.8370549.

[20] S. Hamida, O. E. Gannour, B. Cherradi, H. Ouajji, and A. Raihani, “Optimization of Machine Learning Algorithms Hyper-Parameters for Improving the Prediction of Patients Infected with COVID-19,” in 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, Dec. 2020, pp. 1–6. doi: 10.1109/ICECOCS50124.2020.9314373.

[21] O. Asmae, R. Abdelhadi, C. Bouchaib, S. Sara, and K. Tajeddine, “Parkinson’s Disease Identification using KNN and ANN Algorithms based on Voice Disorder,” in 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, Apr. 2020, pp. 1–6. doi: 10.1109/IRASET48871.2020.9092228.

[22] O. El Gannour, S. Hamida, B. Cherradi, A. Raihani, and H. Moujahid, “Performance Evaluation of Transfer Learning Technique for Automatic Detection of Patients with COVID-19 on X-Ray Images,” in 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, Dec. 2020, pp. 1–6. doi: 10.1109/ICECOCS50124.2020.9314458.

[23] O. Daanouni, B. Cherradi, and A. Tmiri, “Predicting diabetes diseases using mixed data and supervised machine learning algorithms,” in Proceedings of the 4th International Conference on Smart City Applications, Casablanca Morocco, Oct. 2019, pp. 1–6. doi: 10.1145/3368756.3369072.

[24] O. Terrada, B. Cherradi, S. Hamida, A. Raihani, H. Moujahid, and O. Bouattane, “Prediction of Patients with Heart Disease using Artificial Neural Network and Adaptive Boosting techniques,” in 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, Sep. 2020, pp. 1–6. doi: 10.1109/CommNet49926.2020.9199620.

[25] L. Hua, Y. Gu, X. Gu, J. Xue, and T. Ni, “A Novel Brain MRI Image Segmentation Method Using an Improved Multi-View Fuzzy c-Means Clustering Algorithm,” Front. Neurosci., vol. 15, p. 662674, Mar. 2021, doi: 10.3389/fnins.2021.662674.

[26] S. Hamida, B. Cherradi, O. Terrada, A. Raihani, H. Ouajji, and S. Laghmati, “A Novel Feature Extraction System for Cursive Word Vocabulary Recognition using Local Features Descriptors and Gabor Filter,” in 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, Sep. 2020, pp. 1–7. doi: 10.1109/CommNet49926.2020.9199642.

[27] S. Hamida, B. Cherradi, and H. Ouajji, “Handwritten Arabic Words Recognition System Based on HOG and Gabor Filter Descriptors,” in 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, Apr. 2020, pp. 1–4. doi: 10.1109/IRASET48871.2020.9092067.

[28] S. Hamida, B. Cherradi, O. El Gannour, O. Terrada, A. Raihani, and H. Ouajji, “New Database of French Computer Science Words Handwritten Vocabulary,” in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, Jul. 2021, pp. 1–5. doi: 10.1109/ICOTEN52080.2021.9493438.

[29] S. Hamida, O. El Gannour, B. Cherradi, H. Ouajji, and A. Raihani, “Handwritten computer science words vocabulary recognition using

concatenated convolutional neural networks," Multimed Tools Appl, Nov. 2022, doi: 10.1007/s11042-022-14105-2.

[30] S. Hamida, B. Cherradi, A. Raihani, and H. Ouajji, "Performance Evaluation of Machine Learning Algorithms in Handwritten Digits Recognition," in 2019 1st International Conference on Smart Systems and Data Science (ICSSD), Rabat, Morocco, Oct. 2019, pp. 1–6. doi: 10.1109/ICSSD47982.2019.9003052.

[31] O. El Gannour, B. Cherradi, S. Hamida, M. Jebbari, and A. Raihani, "Screening Medical Face Mask for Coronavirus Prevention using Deep Learning and AutoML," in 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, Mar. 2022, pp. 1–7. doi: 10.1109/IRASET52964.2022.9737903.

[32] K. A. Mohamed, E. Elsamahy, and A. Salem, "COVID-19 Disease Detection based on X-Ray Image Classification using CNN with GEV Activation Function," IJACSA, vol. 13, no. 9, 2022, doi: 10.14569/IJACSA.2022.01309103.

[33] K. Ejaz et al., "Segmentation Method for Pathological Brain Tumor and Accurate Detection using MRI," ijacsa, vol. 9, no. 8, 2018, doi: 10.14569/IJACSA.2018.090851.

[34] D. K. Sahoo, S. Mishra, and M. N. Mohanty, "Brain Tumor Segmentation and Classification from MRI Images using Improved FLICM Segmentation and SCA Weight Optimized Wavelet-ELM Model," IJACSA, vol. 13, no. 7, 2022, doi: 10.14569/IJACSA.2022.0130753.

[35] H. Surrisyad and W. -, "A Fast Military Object Recognition using Extreme Learning Approach on CNN," IJACSA, vol. 11, no. 12, 2020, doi: 10.14569/IJACSA.2020.0111227.

[36] P. M. Mah, I. Skalna, M. Eric, and A. John, "Influence of Internet of things on human psychology (internet of thoughts) for education, healthcare, and businesses," EAI Endorsed Trans Mob Com Appl, vol. 7, no. 2, p. e1, Aug. 2022, doi: 10.4108/eetmca.v7i2.2627.

[37] A. Dhankhar, S. Juneja, A. Juneja, and V. Bali, "Kernel Parameter Tuning to Tweak the Performance of Classifiers for Identification of Heart Diseases:," International Journal of E-Health and Medical Communications, vol. 12, no. 4, pp. 1–16, Jul. 2021, doi: 10.4018/IJEHMC.20210701.oa1.

[38] P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Capsule Networks for Brain Tumor Classification Based on MRI Images and Coarse Tumor Boundaries," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, May 2019, pp. 1368–1372. doi: 10.1109/ICASSP.2019.8683759.

[39] E. Irmak, "Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework," Iran J Sci Technol Trans Electr Eng, vol. 45, no. 3, pp. 1015–1036, Sep. 2021, doi: 10.1007/s40998-021-00426-9.

[40] Y. Yang et al., "Glioma Grading on Conventional MR Images: A Deep Learning Study With Transfer Learning," Front Neurosci, vol. 12, p. 804, 2018, doi: 10.3389/fnins.2018.00804.

[41] J. Kang, Z. Ullah, and J. Gwak, "MRI-Based Brain Tumor Classification Using Ensemble of Deep Features and Machine Learning Classifiers," Sensors, vol. 21, no. 6, p. 2222, Mar. 2021, doi: 10.3390/s21062222.

[42] H. T. Nguyen, T. T. Le, T. V. Nguyen, and N. T. Nguyen, "Enhancing MRI Brain Tumor Segmentation with an Additional Classification Network," 2020, doi: 10.48550/ARXIV.2009.12111.

[43] Z. Liu et al., "Deep learning based brain tumor segmentation: a survey," Complex Intell. Syst., vol. 9, no. 1, pp. 1001–1026, Feb. 2023, doi: 10.1007/s40747-022-00815-5.

[44] S. Sharma et al., "Deep Learning Model for Automatic Classification and Prediction of Brain Tumor," Journal of Sensors, vol. 2022, pp. 1–11, Apr. 2022, doi: 10.1155/2022/3065656.

[45] S. Juneja, A. Juneja, G. Dhiman, S. Behl, and S. Kautish, "An Approach for Thoracic Syndrome Classification with Convolutional Neural Networks," Comput Math Methods Med, vol. 2021, p. 3900254, Sep. 2021, doi: 10.1155/2021/3900254.

[46] S. Juneja, A. Juneja, G. Dhiman, S. Behl, and S. Kautish, "An Approach for Thoracic Syndrome Classification with Convolutional Neural Networks," Computational and Mathematical Methods in Medicine, vol. 2021, pp. 1–10, Sep. 2021, doi: 10.1155/2021/3900254.

[47] C. Monga, D. Gupta, D. Prasad, S. Juneja, G. Muhammad, and Z. Ali, "Sustainable Network by Enhancing Attribute-Based Selection Mechanism Using Lagrange Interpolation," Sustainability, vol. 14, no. 10, Art. no. 10, Jan. 2022, doi: 10.3390/su14106082.

[48] M. Jacobson and C. Hedgcoth, "Levels of 5,6-dihydrouridine in relaxed and chloramphenicol transfer ribonucleic acid," Biochemistry, vol. 9, no. 12, pp. 2513–2519, Jun. 1970, doi: 10.1021/bi00814a018.

[49] S. Kanwal, J. Rashid, J. Kim, S. Juneja, G. Dhiman, and A. Hussain, "Mitigating the Coexistence Technique in Wireless Body Area Networks By Using Superframe Interleaving," IETE Journal of Research, pp. 1–15, Mar. 2022, doi: 10.1080/03772063.2022.2043788.

[50] K. Kour et al., "Controlling Agronomic Variables of Saffron Crop Using IoT for Sustainable Agriculture," Sustainability, vol. 14, no. 9, Art. no. 9, Jan. 2022, doi: 10.3390/su14095607.

[51] R. Qian, S. Sengan, and S. Juneja, "English language teaching based on big data analytics in augmentative and alternative communication system," Int J Speech Technol, vol. 25, no. 2, pp. 409–420, Jun. 2022, doi: 10.1007/s10772-022-09960-1.

[52] O. El Gannour, S. Hamida, S. Saleh, Y. Lamalem, B. Cherradi, and A. Raihani, "COVID-19 Detection on X-Ray Images using a Combining Mechanism of Pre-trained CNNs," International Journal of Advanced Computer Science and Applications, pp. 564–570, 2022.

[53] H. Dalianis, "Evaluation Metrics and Evaluation," in Clinical Text Mining, Cham: Springer International Publishing, 2018, pp. 45–53. doi: 10.1007/978-3-319-78503-5_6.

[54] J. Cheng et al., "Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition," PLoS ONE, vol. 10, no. 10, p. e0140381, Oct. 2015, doi: 10.1371/journal.pone.0140381.

Sentiment Analysis on Moroccan Dialect based on ML and Social Media Content Detection

Mouaad Errami[1], Mohamed Amine Ouassil[2], Rabia Rachidi[3],
Bouchaib Cherradi[4], Soufiane Hamida[5], Abdelhadi Raihani[6]
EEIS Laboratory, ENSET of Mohammedia, Hassan II University of Casablanca, Mohammedia 28830, Morocco[1, 2, 4, 5, 6]
LaROSERI Laboratory, Faculty of Science, Chouaib Doukali University, El Jadida, Morocco[3, 4]
STIE Team, CRMEF Casablanca-Settat, Provincial Section of El Jadida, El Jadida 24000, Morocco[4]
GENIUS Laboratory, SupMTI of Rabat, Rabat, Morocco[5]

***Abstract*—As technology continues to evolve, humans tend to follow suit, and currently social media has taken place as the defacto method of communication. As it tends to happen with verbal communication, people express their opinions in written form and through an analysis of their words, one can extract what an individual wants from a product, a topic, or an event. By looking at the emotions expressed in such content, governments, businesses, and people can learn a lot that can help them improve their strategies. Therefore, in this study, we will use different algorithms to improve the Moroccan sentiment classification. The first step is to gather and prepare Moroccan Dialectal Arabic Twitter comments. Then, a lot of different combinations of extraction (n-grams) and weighting schemes (BOW/ TF-IDF) and word embedding for feature construction are applied to get the best classification models. We used Naive Bayes, Random Forests, Support Vector Machines, and Logistic regression and LSTM to classify the data we prepared. Our machine learning approach, which incorporates sentiment analysis, was designed to analyze Twitter comments written in Modern Standard Arabic or Moroccan Dialectal Arabic. As a final benchmark of our paper, we were simply a sliver shy away from the 70% mark in our accuracy by relying on the SVM algorithm. Although not a game-changing result, this was enough to encourage us to continue developing our model further.**

Keywords—Sentiment analysis; Arabic Moroccan dialect; tweets; machine learning

I. Introduction

Natural Language Processing (NLP) attempts to make the machine capable of understanding and generating human language, whether it be written or audible [1] NLP is also one of the most important and challenging areas of artificial intelligence. It has many obstacles [2], especially in Arabic, which will be a topic we discuss later in this article. Sentiment Analysis (SA), or opinion mining, is the mathematical study of people's opinions, emotions, sentiments, ratings, and attitudes about products, services, organizations, individuals, issues, events, topics, and attributes [3]. SA has become one of the most prominent applications of NLP.

In recent years, Machine Learning (ML) and Deep Learning (DL) techniques has been widely used in various classification and prediction problems such as handwritten recognition [4]–[7], medical applications [8]–[13], social media analysis [14]–[16], etc. In particular, applying Arabic sentiment analysis makes it possible to extract the public's opinion on one or many topics through tweets. The use of social media has become a major contributing factor in the performance of Arabic SA tools [17]. With the rise of social media platforms such as Twitter, Facebook, and Instagram, individuals can now express their opinions and feelings on various topics in real-time. This provides a rich source of data for researchers and businesses to understand the sentiment of the public.

Many researchers have opted to use different approaches like sentiment analysis and opinion mining to classify people's views and comments, ranging from negative, positive, and neutral perspectives. These techniques involve using Machine Learning (ML) algorithms to analyze large amounts of textual data to identify patterns and classify the sentiment expressed in the text. By using these techniques, researchers can understand the public's opinion on a variety of topics, such as politics, sports, entertainment, and social issues. One of the challenges of performing Arabic sentiment analysis is the complexity of the Arabic language. Arabic is a highly inflected language with many variations in grammar and vocabulary across different regions. This makes it difficult to develop accurate sentiment analysis tools that can accurately classify the sentiment of the text. However, recent advances in natural language processing and machine learning techniques have made it possible to overcome these challenges.

The following research is an attempt to apply sentiment analysis on Moroccan people's tweets in order to extract reactions from them by using local idioms and terms in their native dialect as a basis. With the increasing use of social media platforms such as Twitter in Morocco, it has become an important source of information for individuals, businesses, and governments. Therefore, there is a growing need for sentiment analysis to help these groups understand the opinions and emotions expressed by the public. Our goal is to develop a sentiment analysis (SA) model that can accurately classify Arabic Moroccan dialect tweets as negative, positive, or neutral. To achieve this, we gathered a large dataset of Moroccan Dialect Arabic (MDA) tweets from Twitter. However, collecting this dataset was not a straightforward task. Moroccan Arabic is a complex dialect that varies greatly from region to region, and there is no standard written form of the dialect. Therefore, we had to collect tweets that were written in the dialect and filter them to collect various sorts of terminology to identify MDA efficiently.

To develop our SA model, we employed various techniques such as word recognition, named entity recognition, stop word removal, and stemming to preprocess the collected tweets. We also experimented with different feature extraction techniques such as n-grams and weighting schemes like Bag of Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF) to construct effective features for our model. Additionally, we used word embedding to capture the semantic relationships between words and phrases. We employed several machine learning algorithms such as Naive Bayes, Random Forests, Support Vector Machines (SVM), Logistic Regression, and Long Short-Term Memory (LSTM) to classify the tweets based on their sentiment. These algorithms were trained on a portion of the dataset and tested on a separate set of tweets to evaluate their accuracy and effectiveness. Our SA model is unique in that it specifically focuses on Moroccan Dialect Arabic tweets and uses local idioms and terms to improve its accuracy. The results of our experiments are promising, as our model achieved high accuracy in classifying tweets as positive, negative, or neutral.

The structure of this article can be outlined as follows: In Section II, the related work on the topic is discussed to provide context and background for the reader. Section III is dedicated to explaining the methodology used in this research, including the steps taken to collect and prepare the data, the feature selection and extraction techniques used, and the machine learning algorithms applied for classification. The results of the experiment are presented in Section IV, where we discuss the performance of the different models. Finally, in Section V, we conclude the paper by summarizing the findings, discussing their implications, and suggesting directions for future research.

II. Related Works

We have noticed that the majority of the SA's research is focused on English. As a result, many high-quality frameworks and tools for English text are now available. For other languages, such as Arabic, however, research efforts are still needed to propose tools that are more refined. However, in recent years, many research topics continue to improve upon Arabic sentiment analysis and other dialectical Arabic sentiment analysis. Duwairi and Qarqaz build a Sentiment Analysis (SA) model for Arabic comments on social media platforms [18]. They used word bi-grams as features for representing the text in their model. They also evaluated the performance of different classifiers, including Support Vector Machine (SVM), Naive Bayes (NB), and K-Nearest Neighbour (KNN), with the use of term frequency (TF) and term frequency inverse document frequency (TF-IDF) weighting methods. Furthermore, Ayah Soufan [19] used text data from Twitter and GoodReads to conduct sentiment analysis for the Arabic language. He also worked on two different tasks: binary and multi-class categorization. He noted that the outcomes vary depending on the dataset and model employed. Modern Standard Arabic (MSA) and Arabic dialects differ on all "linguistic representation levels such as morphology, lexicon, phonology, syntax, semantics, and pragmatics," which is why the results vary depending on the dataset.

In [20], the authors compare different types of ensemble methods. By simply using individual classifiers, on Arabic sentiment analysis, it became apparent that SVM was the better performer, by a decent margin. However, when compared to the performance of a combination of different classifiers, no individual classifier performed nearly as well.

Another work [21] by using NB classifier, described a method for automatically generating a corpus that can be used to train a multilingual sentiment classifier to classify tweets into positive and negative categories.

In [22], the authors collected and labeled a dataset of 17000 Tunisian dialect comments from Facebook, which they used to build a Tunisian dialect sentiment analysis (SA) system using three classification algorithms: support vector machine (SVM), Naive Bayes (NB), and Multi-Layer Perceptron (MLP). They found that their method outperformed models trained on other dialects or on a MSA) dataset.

In the study [23], the authors proposed a framework that employs a variety of approaches and efficient models for both Arabic text pre-processing and ASA. In the case of ASA, their study revealed that Deep learning (DL) models are more efficient and accurate than ML (SVM, NB, and ME). In all of the following scenarios, DL models outperformed traditional models: when using unigrams, when using stop words, without stop words, when using stemming, a without stemming. This study demonstrated the significant accuracy and performance potential of DL for ASA.

In another work [24], the authors analyze the sentiment of 4625 comments written in MSA and multidialectal Arabic from Yahoo!-Maktoob using support vector machines (SVM) and naive Bayes (NB) classifiers, which are commonly used for sentiment classification. They tested the classifiers using both balanced and imbalanced datasets, but the results were unsatisfying as the highest accuracy achieved was 70%.

III. Materials and Methods

A. Global Overview of the Proposed System

Fig. 1 is a plain summary of the approach we decided to adopt throughout our study.

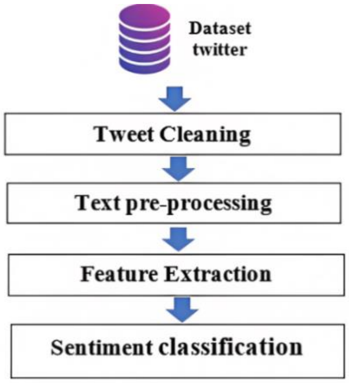


Fig. 1. Simplified Process for SA.

B. Moroccan Dialect Recognition Challenges

Darija is a Moroccan dialect (MD) of Arabic that is widely utilized by Moroccans, brand sites on social media, television programs, and Ads that tend to reach out to the public. It is influenced by various other languages, including Berber, French, Spanish, and Andalusian Arabic, and has its own distinct grammar, vocabulary, and pronunciation, However, Darija maintains its distinctive characteristics in terms of spelling, syntax, vocabulary, and pronunciation [25]. In recent times, the use of written forms of Darija has increased significantly across various platforms. Some difficulties still stand in the face processing Darija, some of which are:

- Domain Dependency [26]: Factors such as culture, context, and basic usage of the language heavily affect performance.
- Code Switching: The French-Spanish colonization of Morocco left a lasting effect on Darija's history, providing her both French and Spanish words/sentences. On top of that, Berber language, which around 40% of Moroccans speak fluently, managed to find a spot in the basic structure of the language. As a result, MSA, Berber, French, Spanish, and English can all be found in Darija writing [27].
- Romanized Arabic: There is no orthographic standard in Darija. It can be written in Arabic [28], Latin, or both (Arabizi). A combination of the two is often used on social media, with numbers being used as substitutes to inexistent letters in the Roman alphabet, such as welcome=Mer7ba = مرحبا, happy= fer7an=سعيد, sleep=n3ass=النوم.

However, the biggest challenge of them all is the limited resources, and the difficulty of extracting datasets.

C. Process of Moroccan Arabic Dialect Tweet for Sentiment Analysis

In this section, we present a Machine Learning (ML) process for SA conducted on twitter comments written in MDA. This process starts by cleaning tweets, pre-processing them and before classification, where we performed a features selection process aiming to reduce the dimensionality and improve the quality of our classification models. Finally, comes the evaluation step where the performance of our model is measured. Fig. 2 describes the proposed ML system of Arabic Tweet for sentiment analysis.

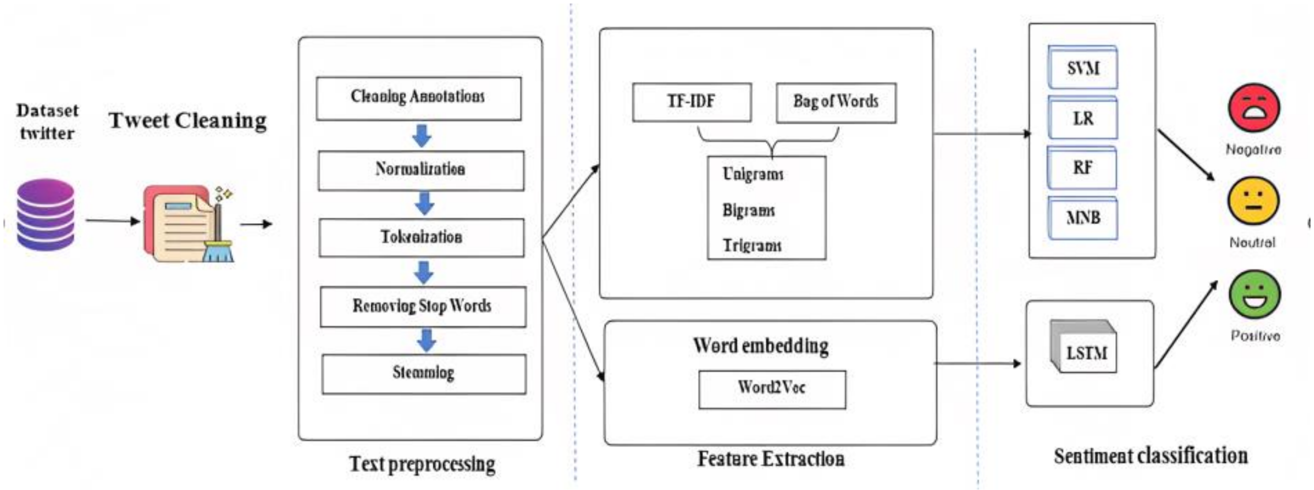


Fig. 2. Sentiment analysis: A ML approach.

D. Dataset Description

In our study, we used a dataset of SA for social media posts in Arabic dialect, publicly available from the Modeling, Simulation and Data Analysis (MSDA)[1]. The data (tweets), scrapped from active users located in a predefined set of Arab countries consisting mainly of Lebanon, Algeria, Egypt, Tunisia and Morocco, got narrowed down to tweets relevant to Morocco only. The selected dataset contains 1605 positive tweet, 1620 negative tweet and 1630 neutral tweet. In Table I, we present some illustrative samples.

The word cloud schemes in Fig. 3, Fig. 4, and Fig. 5 present the density of words in relevance to the class from where it originates.

E. Data Pre-processing Steps

1) Tweet cleaning: In order to clean our dataset, we had to remove the tweets:

- Written in Roman alphabet.
- Those are spam or insults.
- Composed of a single word.

2) Cleaning annotations: In order to rend our tweets more functional, all “http/https“ are removed, as well as special symbols (*, &, $, %,-,_,:,!,><). Whatever we remove, we replace with an empty space. We apply the same for the emoji’s.

[1] https://msda.um6p.ma/msda_datasets

TABLE I. ANNOTATED SAMPLE.

Comment	English translation	Sentiment
لن نضيع اوقات اطفالنا بتعليم لهجة ما عندها فايدة و كافي ان كل طفولتهم تروح في التعليم نزيدو لهم لهجة تتعبهم و تمحي هويتهم	We will not waste our children's time by teaching them a dialect that's useless. It's enough that their time is consumed by education, and to top it all we add a dialect that would tire them and erase their identity.	Negative
هاد الولد ممتل المغرب في أكبر مسابقة ديال صناع المحتوى في العالم العربي.. وليوم هو في النهائيات	This boy is the Moroccan representative in the biggest content creation contest in the Arabic world, and today he's a finalist.	Neutral
ترامب عين البروفيسور منصف السلاوي للإشراف على تطوير اللقاح , لمغاربة حصلو له مغربي ولا .فواش نقولو أمريكي ولا بلجيكي	Trump appointed Prof. Monsif Essalaoui as a supervisor for vaccine development. Moroccans didn't know what to call him 'Moroccan', 'American, or 'Belgium'.	Positive

Fig. 3. Positive class word occurrence.

Fig. 4. Negative class word occurrence.

Fig. 5. Neutral class word occurrence.

3) Tweets pre-processing: In order to boost the performance of the SA process, we must perform several pre-processing steps on the collected tweets. These steps are as follows:

- Normalization: Normalization is necessary, as there is no universally accepted rule for the spelling of certain Arabic letters. We perform our normalization by deleting all unnecessary spaces, and then replacing every un-normalized letter with its normalized version. As indicated in Table II, based on the Pyarabic library, normalization is as such:

- Tokenization: as an integral step in SA, reduces typographical variation of words. It is also necessary since it is required to use techniques such as Feature Extraction [29]. As the Arabic language is known to be hard to deal with, a dictionary of features that can transform words into feature vectors, or feature indexes, is a must. This way, the index of the word is linked to its frequency in the complete Dataset.

TABLE II. NORMALIZATION OF SOME ARABIC LETTERS

UnNormalized Arabic Letters	Normalized Letters
ؤ، ىء **Hamza**	ء
أ، ا **Alef**	ا
ال، أل **Lamalef**	ال
ى **Yah**	ى
ة **Hah**	ه
ؤ **Wow_hamzah**	و

- Removing stop words: Not all stop words are actually helpful in understanding the full meaning of the tweet, which is why deleting some of them is an essential step. Deleting words like articles, prepositions, conjunctions, and pronouns that are overused and do not provide any information about the subject of the documents must be carefully done.

- Data Stemming and Lemmatization: Stemming is a process where, by removing prefixes and suffixes, a word is returned to its root form [30] . Similarly, lemmatization finds the basis of the word while taking into consideration its morphological nature; a meaning conserving lemma is then extracted. We opted to use the following stemmers in our study:

- ISRI Stemmer [31] is a stemmer that uses the same sequence as other stemmers to derive the root-base of Arabic words[2]. On the other hand, ISRI does not need to confirm the extracted root with a stored root dictionary, as the other stemmer does, because locating the correct root is not critical. ISRI eliminates two- and three-letter prefixes, normalizes Hamza, and eliminates connector letters such ("ثم ").

- Assem's Arabic Light stemmer is frequently used for a variety of text processing tasks, such as text categorization, information extraction, text summarization, and search engine indexing. It is important for researchers to carefully evaluate and compare different stemmers in order to select the one that is most suitable for their specific needs and context[3]. Table III present examples of executing tasks.

[2] https://github.com/baraayouzbashi/ISRI-Java
[3] https://github.com/assem-ch/arabicstemmer_pythonlibrary

TABLE III. EXAMPLE OF EXECUTING THE PROCESSING TASKS ON A COMMENT

Task	Result	Simulated English result
Original text	تبارك الله مستوى وصل الطالب المغربي ☹□☹□☹□☹□ ما فهمتش واش خارجببين إصلاحية, تحية 👇👇ثانوية لجبببيل الذهبي	How disappointiiiiing is the current level of Moroccan students ☹□☹□☹□☹□. I do not know if they graduated high school or juvieeeeee👇👇. Salute to the golden geneeeeration.
Cleaning	تبارك الله مستوى وصل الطالب المغربي ما فهمتش واش خارجببين ثانوية إصلاحية تحية لجبببيل الذهبي	How disappointiiiiing is the current level of Moroccan students. I do not know if they graduated high school or juvieeeeee. Salute to the golden geneeeeration.
Normalizing	تبارك الله مستوى وصل الطالب المغربي ما فهمتش واش خارجين ثانوية إصلاحية تحيةً لجيل الذهبي	How disappointiiiiing is the current level of Moroccan students. I do not know if they graduated high school or juvieeeeee. Salute to the golden geneeeeration.
Tokenization	['تبارك ', 'الله', 'مستوى', 'وصل', ' ما ', 'الطالب', 'المغربي', 'فهم ', 'واش', 'ثانوية', 'إصلاحية', 'تحية', 'لجيل', 'الذهب'],	[’How’,’disappointing’,’is’,’the’,’current’,’level’,’of’, ’Moroccan’,’students’,’I’,’do’,’not’,’know’,’if’,’they’,’graduated’,’high’,’school’ ’or’,’juvie.’,’Salute’,’to’,’the’,’golden’,’generation’]
Stop words Removal	[' بركة ', 'الله', 'مستوى', 'وصل', 'الطالب', 'المغربي', 'فهم ', 'واش', 'ثانوية', 'إصلاحية', 'تحية', 'لجيل', 'الذهب'],	[‘disappointing’,’current’,’level’,’ Moroccan’,’students’, ’graduated’,’high’,’school’,’juvie’, ’Salute’,’golden’,’generation’]
Stemming	['مستوى','صل','طلب','مغرب', بركة,'إصلاح','ثان','خارج', 'جيل','ذهب']	[‘ disappoint ’,’ current ’,’ level ’,’ Moroccan ’,’ student ’,’ graduat ’,’ high ’,’ school ’,’ juvi ’,’ Salut ’,’ golden ’,’ gener ’]

- Tashaphyne is a stemmer tool specifically designed for analyzing the sentiment of Arabic roots [32]. It has been shown to be effective at mapping Arabic words to their basic roots for sentiment analysis tasks, resulting in significant improvements compared to baseline performance [33].

F. Features Extraction

N-gram is a traditional method for identifying formal terms in tweets that takes into account the occurrences of N-words in the tweet. The value of n is used to refer to N-grams of larger sizes [34]. During our research, we used unigrams, bigrams, and trigrams to get the best results. Table IV is the perfect example on the results obtained by applying this method on our tweets.

We employed two alternative feature extraction algorithms: Bag of Words [35] generates a vector that represents the frequency of occurrences of words. The Bow model, frequently used in text processing to classify and recognize documents, is known for simplicity and effectiveness. The TF-IDF was created to address the limitations of the more basic strategy of counting each term's occurrences, in which highly common terms end up with very big values and uncommon words (which are normally a good discriminant between texts) can be buried in the noise. The TF-IDF approach is used to weight terms based on their relevance within the document and corpus [36]. Words with a high TF-IDF score are those that appear frequently in a document but not across the corpus.

TABLE IV. FEATURES EXTRACTION

Task	Result	Simulated English result
Cleaned text	['مستوى','صل','طلب','مغرب', بركة,'إصلاح','ثان','خارج', 'جيل','ذهب']	[‘ disappoint ’,’ current ’,’ level ’,’ Moroccan ’,’ student ’,’ graduat ’,’ high ’,’ school ’,’ juvi ’,’ Salut ’,’ golden ’,’ gener ’]
Uni-gram n=1	['مستوى', 'صل', 'طلب', 'مغرب', 'خارج', 'إصلاح', 'بركة', 'جيل']	[‘ disappoint ’,’ current ’,’ level ’,’ Moroccan ’,’ student ’,’ graduat ’,’ high ’,’ school ’,’ juvi ’,’ Salut ’,’ golden ’,’ gener ’]
Bi-gram n=2	['مستوى صل', 'صل طلب', 'طلب مغرب', 'مغرب خارج', 'خارج إصلاح', 'إصلاح بركة', 'بركة جيل']	['disappoint current', 'current level', 'level moroccan', 'moroccan student', 'student graduat', 'graduat high', 'high school', 'school juvi', 'juvi salut', 'salut golden', 'golden gener']
Tri-gram n=3	['مستوى صل طلب', 'صل طلب مغرب', 'طلب مغرب خارج', 'مغرب خارج إصلاح', 'خارج إصلاح بركة', 'إصلاح بركة جيل']	['disappoint current level', 'current level moroccan', 'level moroccan student', 'moroccan student graduat', 'student graduat high', 'graduat high school', 'high school juvi', 'school juvi salut', 'juvi salut golden', 'salut golden gener']

Term Frequency is calculated by dividing the total number of words in a document by the number of times each term appears in the document.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \tag{1}$$

Inverse Document Frequency is the log of the number of documents divided by the number of documents that contain the word.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{2}$$

t_{ij} = Occurrences of i in j.

df_i = Documents containing i.

N = Total number of documents.

This increases the weight of rare words across all documents in the corpus. Note that when we compute the TF-IDF for every word in every document of a corpus, it will form a matrix of shape (documents * vocabulary).

Word embedding methods turn words into digital vectors. These vectors include more or less information about the semantics and syntax of the word depending on the model employed and the context in which it was used [37]. There are numerous word embedding strategies available.

Word2vec is a renowned word-embedding model developed at Google in 2013. It incorporates a neural network layer to either predict adjacent words to the target word (context) in the case of the skip gram architecture, or the word from its neighbors in the case of the CBOW (Continuous bag of words). Word2vec's input and output are a one-time encoding of the dataset's vocabulary words. During training, a window moves through the corpus [38], training the neural network to predict surrounding words or a target word for each word by assigning a probability to the words. After training, the network layer is the vector representation of a word. In our case, we implemented a Wikipedia-made word2vec model for Arabic language.

G. *The Used Classifiers*

The data was classified using four supervised machine-learning algorithms: Naive Support Vector Machine classifier (SVM), Multinomial Naive Bayes (MNB), Logistic regression (LR), Random forest (RF), and one DL algorithm: Long Short-Term Memory (LSTM). We aim to answer the following question: Can ensemble learning (combining different classification algorithms) improve Arabic sentiment classification?

In what follows, we explain those algorithms:

1) Support vector machine: Support vector machine is a supervised learning technique commonly used to classify jobs. In a high-dimensional space, it is extremely efficient. It also works well when the number of dimensions is more than the number of data. SVM is a discriminative classifier whose basic principle is to construct decision boundaries that distinguish between a set of objects belonging to various classes [39].

2) Multinomial Naive Bayes (MNB): Multinomial Naive Bayes is a supervised machine-learning algorithm that relies on annotated data for training. It is based on the Bayes theorem and is well suited for high-dimensional input data like text. It is a generative model that relies on conditional probabilities and assumes that the features are conditionally independent [40]. Despite its simplicity, it is a very effective algorithm and can outperform more complex classifiers on short datasets. It is also relatively robust, easy to implement, and fast to run.

3) Logistic regression: The statistical model of logistic regression is used to investigate the associations between a set of qualitative variables X and a qualitative variable Y. A logistic function is used as a link function in the generalized linear model. The optimization of the regression coefficients in a logistic regression model can also be used to forecast whether an event will occur (value of 1) or not (value of 0). This outcome is always between 0 and 1 [41]. The event is more likely to happen if the anticipated value is above a certain threshold, but not if it is below the same threshold. The hypothesis representation of logistic regression defined as follow:

$$h\theta(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} \qquad (3)$$

The cost function of LR is defined as follow:

$$\mathrm{J}(\theta)=\frac{1}{\mathrm{M}}\sum_{\mathrm{i}=1}^{\mathrm{M}} \mathrm{cost}\left(\mathrm{h}\theta\left(\mathrm{x}^{(\mathrm{i})}\right),\mathrm{y}^{(\mathrm{i})}\right) \qquad (4)$$

With M is the size of training set.

4) Random forest: Random forest is a consensus approach for solving regression and classification problems in supervised machine learning (ML). Each random forest is made up of several decision trees that work together to provide a single prediction [42].

5) Long short-term memory: Long short-term memory (LSTM) networks are a type of recurrent neural networks (RNN) that have a longer memory than traditional RNNs. They are well suited for learning from large, complex data sets that have long delays between important events. LSTM units are used to build the layers of an RNN, which is then called an LSTM network. LSTMs allow RNNs to retain information over a long period of time [43] by using a memory cell that is similar to computer memory, where it can read, write, and delete information as needed. This memory cell is gated, meaning that it decides whether to store or delete information based on its perceived importance, as determined by learned weights. There are three gates in an LSTM: an input gate, a forget gate, and an output gate. Fig. 6 illustrates a LSTM memory cell structure. These gates control whether new input is accepted (input gate), whether information is suppressed because it is not important (forget gate), or whether it is allowed to influence the output at the current time step (output gate) [44]. The importance of each piece of information is determined by the learned weights of the LSTM, which are updated as the algorithm learns over time.

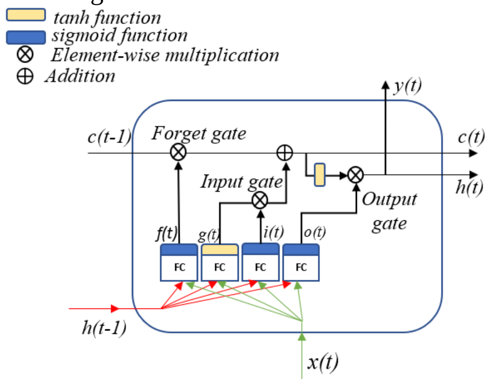


Fig. 6. Single LSTM memory cell.

IV. Results and Discussion

A. *Confusion Matrix and Performance Evaluation Measures*

A confusion matrix, also known as a contingency table, is a tool used to evaluate the accuracy of a machine learning model's predictions in classification problems. It compares the model's predictions to reality and shows how often the predictions are correct. To understand how a confusion matrix works, it is important to familiarize oneself with the four key terms: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These terms represent the different outcomes of a prediction and are defined as follows:

- TP (True Positives): the cases where the prediction is positive, and where the actual value is actually positive. Example: the doctor tells you that you are pregnant, and you are indeed pregnant.
- TN (True Negatives): the cases where the prediction is negative, and where the actual value is actually

negative. Example: the doctor tells you that you are not pregnant, and you are indeed not pregnant.

- FP (False Positive): Cases where the prediction is positive, but the actual value is negative. Example: the doctor tells you that you are pregnant, but you are not pregnant.
- FN (False Negative): Cases where the prediction is negative, but the actual value is positive. Example: the doctor tells you that you are not pregnant, but you are pregnant. Various measures can be derived from a confusion matrix. In this work, we utilized four scoring performance metrics: accuracy, precision, recall, and F1 score. The following equations provide these metrics:

$$F_1 \text{ score} = \frac{2*(\text{ precision} \times \text{recall})}{\text{precision} + \text{recall}} \quad (5)$$

$$\text{Recall}= \frac{\text{TP}}{\text{FN+TP}} \quad (6)$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (7)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

B. Training of ML Models Results

Our implementation and training of the model heavily relied on the Scikit-Learn library. Vectorization, pipeline and training were imported from its sub libraries. Similarly, our different scripts (based on SVM, Random Forest, Naïve Bayes, and Logistic regression) were applied and tuned using this library. GridSearchCV adjusted the parameters, and followed up with training and comparison; for LSTM, it was the TenserFlow library. We also used Google Colab platform to train our models.

C. Testing of ML Models Results

From Table V we are able to gauge the average accuracy obtained by using different algorithms on top of two separate vectorization methods (BOW, TF-IDF), which is in and of itself topped off with stemming algorithms, including the ISRI Stemmer, Assem's Arabic Light Stemmer and Tashafyn light Stemmer. Several tests were carried out on the dataset in order to assess the influence of stemming after it had been pre-processed (without undergoing stemming). Based on the outcomes of these tests, we can safely deduce that the SVM algorithm is the highest performer, boasting an accuracy of 0.68. Combining ISRI Stemmer, TF-IDF, Uni-gam + Bi-gam, and the SVM classifier resulted in this accuracy. However, when looking at the Tashafyn light stemmer system, we obtained the highest accuracy after using a mix of Bag of Words, Uni-gam + Bi-gam + Tri-Gam, and the SVM algorithm once again. This accuracy clocked at 0.678. As we compare these results with those obtained when using no Stemming, we notice that stemming proved itself an indispensable method that reliably improves performance and accuracy.

TABLE V. CLASSIFICATION OF EXPERIMENTATION RESULTS

Stemming Mechanism	**Classifier**	**TF-IDF**			**Bag of Words**			**Word Embedding**
		Uni-gram	**1g+2g**	**1g+2g+3g**	**Uni-gram**	**1g+2g**	**1g+2g+3g**	
No-stemming	LR	63.00%	63.00%	65.00%	64.00%	64.00%	65.00%	-
	RF	59.00%	58.00%	57.00%	59.00%	57.00%	57.00%	-
	NB	60.00%	59.00%	58.00%	60.00%	59.00%	58.00%	-
	SVM	64.00%	65.00%	64.00%	63.00%	64.00%	64.00%	-
	LSTM	-	-	-	-	-	-	61.50%
Tashafyn Light Stemmer	LR	66.39%	66.59%	66.80%	66.39%	66.25%	66.00%	-
	RF	61.51%	61.30%	60.14%	63.16%	61.30%	61.70%	-
	NB	63.78%	63.51%	63.23%	63.78%	63.30%	64.00%	-
	SVM	66.39%	68.52%	67.42%	65.49%	67.14%	**67.80%**	-
	LSTM	-	-	-	-	-	-	63.00%
ISRI Stemmer	LR	66.87%	67.36%	67.14%	66.87%	66.59%	66.94%	-
	RF	63.23%	62.34%	62.40%	63.57%	63.09%	61.85%	-
	NB	63.37%	62.82%	62.34%	63.37%	62.82%	62.34%	-
	SVM	**67.62%**	**68.59%**	**68.27%**	**67.88%**	**67.56%**	67.13%	-
	LSTM	-	-	-	-	-	-	**63.90%**
Assem's Arabic Light Stemmer	LR	64.05%	65.01%	65.15%	64.06%	64.95%	65.22%	-
	RF	63.24%	61.37%	60.82%	63.30%	61.51%	60.93%	-
	NB	62.54%	61.85%	61.37%	62.54%	61.58%	61.37%	-
	SVM	65.43%	65.15%	65.08%	63.98%	63.36%	64.88%	-
	LSTM	-	-	-	-	-	-	62.00%

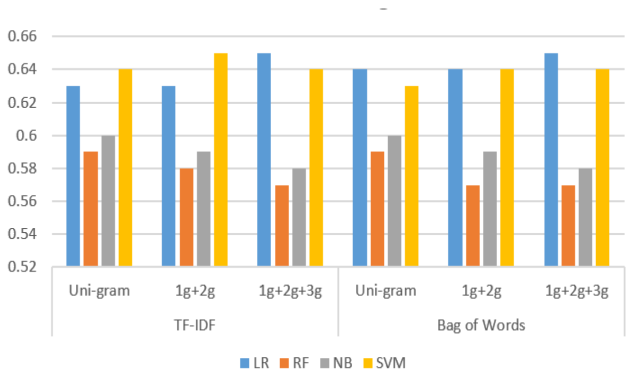


Fig. 7. Classifiers' accuracy with no stemmers.

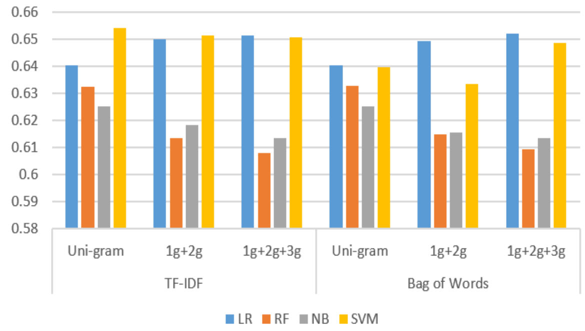


Fig. 8. Classifiers' accuracy with assem's arabic light stemmer.

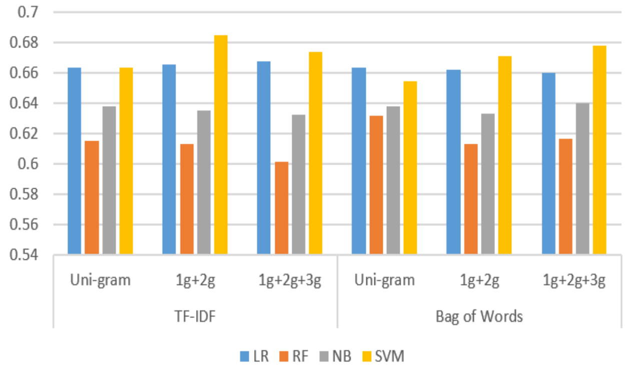


Fig. 9. Classifiers' accuracy with tashafyn light stemmers.

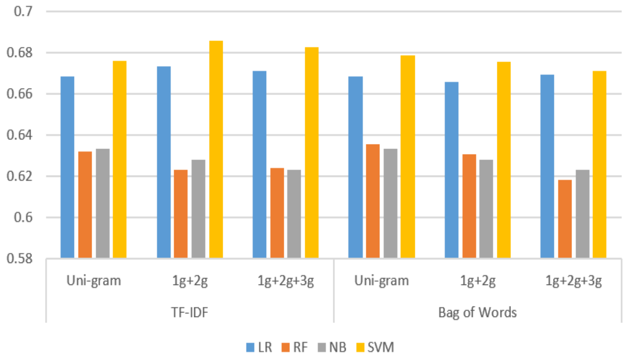


Fig. 10. Classifiers' accuracy with ISRI stemmers.

Fig. 7, Fig. 8, Fig. 9 and Fig. 10, which managed to better visualize the performance spread using the different stemmers, show a clear pattern where SVM and LR are the top performers even when taking into account the different feature extraction methods implemented (TF-IDF and Bag of Words). We managed to push the SVM accuracy all the way to 68.59% using the ISRI Stemmer as well as the combination of TF-IDF and 1g-2g feature extraction methods (these same methods also resulted in the highest value for LR: 67.36%). It should also be noted that despite implementing three different types of stemmers, we do not see very strong variations in the accuracy score across the board. This is largely because these custom-made stemmers for the Arabic language still leave a lot to be desired in terms of efficiency and performance.

Table VI summarizes our top scorer's (ISRI Stemmer+ (Uni-gram+Bigram) + SVM+TD-IDF (feature extraction)) Performance evaluation measures- the Precision, Recall, and F1-score for each one of our selected sentiments (positive, negative, or neutral). We notice that our model is extremely accurate when classifying neutral sentiments, as it has the highest precision score of 0.74.

TABLE VI. PERFORMANCE EVALUATION METRICS

Model		Precision	Recall	F1-score
ISRI Stemmer +(Uni-gram+Bigram)+SVM+TD-IDF (feature extraction)	neg	0.64	0.64	0.64
	neu	0.74	0.74	0.74
	pos	0.66	0.66	0.66

The confusion matrix of the model that had better performance in the detection sentiment analysis is shown in Fig. 11. On the diagonal, we can see 260 neutral sentiments were correctly predicted as neutral, and 198 positive sentiments were also correctly predicted as positive, and the same can be said for the 168 negative sentiment. On the other hand, it is obvious that the outliers of the matrix are heavy on numbers, as the maximum that a model should have wrongly predicted should not exceed 5-10.

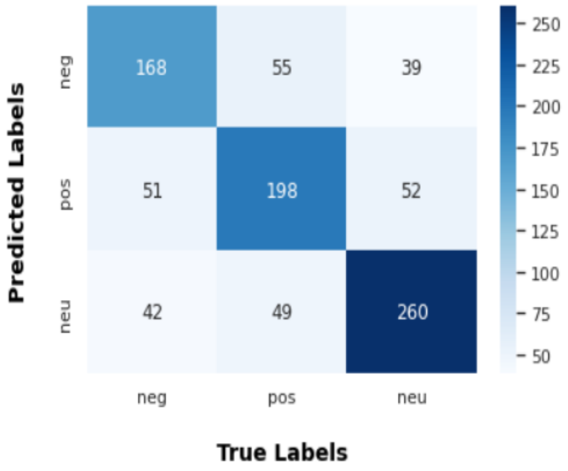


Fig. 11. Confusion matrix of the model (ISRI Stemmer+ (Uni-gram+Bigram) + SVM+TD-IDF (feature extraction).

For LSTM model that we adapted for our research, we managed to achieve the results described in Table VII. We immediately notice that the Accuracy, Recall, and F1-score all receive a slight bump in their values as we apply the different Stemmers. However, for our Precision score, our max value remains the one where no Stemmer is applied.

TABLE VII. LSTM'S MODELS PERFORMANCE

Stemmers	Accuracy	Recall	F1-score	Precision
No-stemming	61.50%	68.57%	67.29%	66.06%
Tashafyn Light Stemmer	63.00%	72.83%	68.97%	65.49%
ISRI Stemmer	63.90%	73.10%	69.25%	65.79%
Assem's Arabic Light Stemmer	62.00%	71.70%	68.30%	65.22%

D. *Discussion*

The ROC curve is an evaluation metric commonly used to plot the True Positive Rate vs. the False Positive Rate. While this metric is mainly used for classification problems that are binary, we managed to push it in order to encompass out multiclass classification problem, all by using the one vs. Rest technique (We calculate the AUC-ROC curve by considering each label independently). In Fig. 12, based on the results from the SVM model, we see how our Neutral label is our best performer, while the Negative and Positive labels are lagging slightly behind. While the Negative and Positive labels intertwine at different thresholds, the Neutral label maintains an advantage throughout. Our results are far from perfect, but they are not disappointing as there is still much to improve upon our study.

The support vector machine (SVM) algorithm outperformed all other classifiers on all datasets, showing a significant difference in performance. SVM is a popular choice for sentiment analysis studies due to its various advantages, such as its ability to handle high-dimensional spaces efficiently and its robustness when working with a sparse set of samples. It also considers all features relevant.

As one can tell from Table VIII, LSTM was nowhere near as powerfully performant as the SVM algorithm. Though Word Embedding is said to be more advanced compared to the bag of words method, in our case where we are processing the Darija Language, it is actually held back by the fact that the Dictionary used in the Word2Vec method is based on the Arabic language and not Darija. This means that when classifying Darija words, many are found to be unclassifiable.

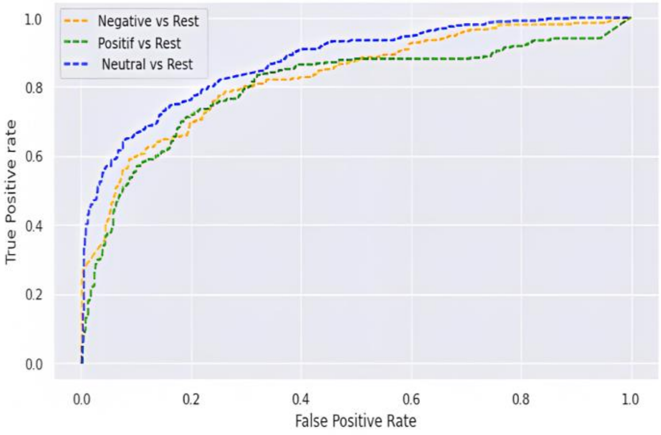


Fig. 12. ROC curves one vs rest.

TABLE VIII. SUMMARY OF ALL THE EXPERIMENTAL FINDINGS PROVIDED IN THIS RESEARCH

Reference	Dataset	Feature Representation	Classification Algorithm	Best Accuracy
[18]	Scrapping dataset	(TF) and term frequency inverse document frequency (TF-IDF)	Supervised Classification(SVM, NB, and K-nearest neighbour (KNN)	67,19%
[19]	QRCI , ArTwitter, ASTD Comb,LABR	Word embedding	SVM,NMB,LSTM,CNN	92,4 %
[20]	MSAC (Moroccan Sentiment Analysis Corpus)	Bag of words	Naive Bayes classifier (NB), Support Vector Machine classifier (SVM), Maximum Entropy (ME)	82.5 %
[22]	TSAC (Tunisian Sentiment Analysis Corpus)	None	(SVM) and Naive Bayes (NB)	76,41%
[24]	Yahoo!-Maktoob	None	SVM, NB	68.2%
This work	**dataset of Sentiment Analysis for social media posts in Arabic dialect from MSDA**	**(TF) and term frequency inverse document frequency (TF-IDF),** **Word embedding** **Bag of words, N-gram**	**LR, RF, MNB, SVM, LSTM**	**68,59%**

V. CONCLUSION AND PERSPECTIVES

In this study, we went through the steps and methods that allowed us to test the different approaches of sentiment analysis' performance and efficiency. Our Dataset, consisting of 4855 tweets/comments split into three balanced groups of negative, positive and neutral sentiments, underwent multiple thorough processes in order to take advantage of it. We began by incorporating a variety of pre-processing techniques (stemming, normalization, tokenization, stop words, etc.) to improve Sentiment Analysis of Moroccan Tweets. Then, in the hopes of drawing out the maximum possible accuracy, five classification algorithms: NB, LR, RF, SVM, and LSTM, were combined as an application of the Ensemble Method. Finally, we delved into a comparison of three types of FE method: BOW, TF-IDF and WE. The results for individual classifiers made it clear that SVM performed on a higher level compared to the other algorithms. There was also a noticeable performance boost when it came to using Stemmers vs. using no stemmers; the only drawback being an increase in the computational time. Our next step is to use other DL models such as BERT

We plan to further improve our SA model by incorporating more advanced deep learning techniques, such as BERT, and exploring other FE methods that may provide better accuracy. We also aim to expand our dataset to include more diverse sources of Moroccan Arabic language and include more topics to enhance the model's performance on a wider range of subjects. Additionally, we plan to extend our research to cover other Arabic dialects to provide more comprehensive sentiment analysis for the Arabic language. Finally, we hope to collaborate with other researchers in the field to develop a standardized evaluation framework for Arabic sentiment analysis to facilitate comparison and benchmarking between different approaches and models.

REFERENCES

[1] R. Mihalcea, H. Liu, and H. Lieberman, "NLP (Natural Language Processing) for NLP (Natural Language Programming)," in Computational Linguistics and Intelligent Text Processing, vol. 3878, A. Gelbukh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 319–330. doi: 10.1007/11671299_34.

[2] S. Elkateb, W. Black, and D. Farwell, "Arabic WordNet and the challenges of Arabic," p. 10.

[3] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," Knowl Inf Syst, vol. 60, no. 2, pp. 617–663, Aug. 2019, doi: 10.1007/s10115-018-1236-4.

[4] S. Hamida, B. Cherradi, O. El Gannour, O. Terrada, A. Raihani, and H. Ouajji, "New Database of French Computer Science Words Handwritten Vocabulary," in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, Jul. 2021, pp. 1–5. doi: 10.1109/ICOTEN52080.2021.9493438.

[5] S. Hamida, B. Cherradi, and H. Ouajji, "Handwritten Arabic Words Recognition System Based on HOG and Gabor Filter Descriptors," in 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, Apr. 2020, pp. 1–4. doi: 10.1109/IRASET48871.2020.9092067.

[6] S. Hamida, B. Cherradi, O. Terrada, A. Raihani, H. Ouajji, and S. Laghmati, "A Novel Feature Extraction System for Cursive Word Vocabulary Recognition using Local Features Descriptors and Gabor Filter," in 2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, Sep. 2020, pp. 1–7. doi: 10.1109/CommNet49926.2020.9199642.

[7] S. Aqab and M. Usman, "Handwriting Recognition using Artificial Intelligence Neural Network and Image Processing," IJACSA, vol. 11, no. 7, 2020, doi: 10.14569/IJACSA.2020.0110719.

[8] B. Cherradi, O. Terrada, A. Ouhmida, S. Hamida, A. Raihani, and O. Bouattane, "Computer-Aided Diagnosis System for Early Prediction of Atherosclerosis using Machine Learning and K-fold cross-validation," in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, Jul. 2021, pp. 1–9. doi: 10.1109/ICOTEN52080.2021.9493524.

[9] O. El Gannour et al., "Concatenation of Pre-Trained Convolutional Neural Networks for Enhanced COVID-19 Screening Using Transfer Learning Technique," Electronics, vol. 11, no. 1, p. 103, Dec. 2021, doi: 10.3390/electronics11010103.

[10] N. A. Ali, A. E. abbassi, and B. Cherradi, "The performances of iterative type-2 fuzzy C-mean on GPU for image segmentation," J Supercomput, vol. 78, no. 2, pp. 1583–1601, Feb. 2022, doi: 10.1007/s11227-021-03928-9.

[11] H. Moujahid et al., "Combining CNN and Grad-Cam for COVID-19 Disease Prediction and Visual Explanation," Intelligent Automation & Soft Computing, vol. 32, no. 2, pp. 723–745, 2022, doi: 10.32604/iasc.2022.022179.

[12] O. Daanouni, B. Cherradi, and A. Tmiri, "Predicting diabetes diseases using mixed data and supervised machine learning algorithms," in Proceedings of the 4th International Conference on Smart City Applications, Casablanca Morocco, Oct. 2019, pp. 1–6. doi: 10.1145/3368756.3369072.

[13] S. Hamida, O. El Gannour, B. Cherradi, A. Raihani, H. Moujahid, and H. Ouajji, "A Novel COVID-19 Diagnosis Support System Using the Stacking Approach and Transfer Learning Technique on Chest X-Ray Images," Journal of Healthcare Engineering, vol. 2021, pp. 1–17, Nov. 2021, doi: 10.1155/2021/9437538.

[14] D. Irawan, D. I. Sensuse, P. A. W. Putro, and A. Prasetyo, "Public Response to the Legalization of The Criminal Code Bill with Twitter Data Sentiment Analysis," IJACSA, vol. 14, no. 2, 2023, doi: 10.14569/IJACSA.2023.0140236.

[15] M. Amine, H. Ait, R. Moulouki, S. Nkiri, and M. Azouazi, "An Improved Social Media Analysis on 3 Layers: A Real Time Enhanced Recommendation System," ijacsa, vol. 9, no. 2, 2018, doi: 10.14569/IJACSA.2018.090234.

[16] M.-A. Ouassil, B. Cherradi, S. Hamida, M. Errami, O. E. Gannour, and A. Raihani, "A Fake News Detection System based on Combination of Word Embedded Techniques and Hybrid Deep Learning Model," IJACSA, vol. 13, no. 10, 2022, doi: 10.14569/IJACSA.2022.0131061.

[17] M. Maghfour and A. Elouardighi, "Standard and Dialectal Arabic Text Classification for Sentiment Analysis," in Model and Data Engineering, vol. 11163, E. H. Abdelwahed, L. Bellatreche, M. Golfarelli, D. Méry, and C. Ordonez, Eds. Cham: Springer International Publishing, 2018, pp. 282–291. doi: 10.1007/978-3-030-00856-7_18.

[18] R. M. Duwairi and I. Qarqaz, "Arabic Sentiment Analysis Using Supervised Classification," in 2014 International Conference on Future Internet of Things and Cloud, Barcelona, Spain, Aug. 2014, pp. 579–583. doi: 10.1109/FiCloud.2014.100.

[19] A. Soufan, "Deep Learning for Sentiment Analysis of Arabic Text," in Proceedings of the ArabWIC 6th Annual International Conference Research Track on - ArabWIC 2019, Rabat, Morocco, 2019, pp. 1–8. doi: 10.1145/3333165.3333185.

[20] A. Oussous, A. A. Lahcen, and S. Belfkih, "Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning," in Big Data, Cloud and Applications, vol. 872, Y. Tabii, M. Lazaar, M. Al Achhab, and N. Enneya, Eds. Cham: Springer International Publishing, 2018, pp. 91–104. doi: 10.1007/978-3-319-96292-4_8.

[21] A. E. Abdouli, L. Hassouni, and H. Anoun, "Sentiment Analysis of Moroccan Tweets using Naive Bayes Algorithm," vol. 15, no. 12, p. 11, 2017.

[22] S. Medhaffar, F. Bougares, Y. Estève, and L. Hadrich-Belguith, "Sentiment Analysis of Tunisian Dialects: Linguistic Ressources and Experiments," in Proceedings of the Third Arabic Natural Language Processing Workshop, Valencia, Spain, 2017, pp. 55–61. doi: 10.18653/v1/W17-1307.

[23] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "ASA: A framework for Arabic sentiment analysis," Journal of Information Science, vol. 46, no. 4, pp. 544–559, Aug. 2020, doi: 10.1177/0165551519849516.

[24] M. N. Al-Kabi, N. A. Abdulla, and M. Al-Ayyoub, "An analytical study of Arabic sentiments: Maktoob case study," in 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), London, United Kingdom, Dec. 2013, pp. 89–94. doi: 10.1109/ICITST.2013.6750168.

[25] S. Harrat, K. Meftouh, and K. Smaïli, "Maghrebi Arabic dialect processing: an overview," vol. 1, no. 1, p. 8, 2018.

[26] A. Ligthart, C. Catal, and B. Tekinerdogan, "Systematic reviews in sentiment analysis: a tertiary study," Artif Intell Rev, vol. 54, no. 7, pp. 4997–5053, Oct. 2021, doi: 10.1007/s10462-021-09973-3.

[27] Y. Samih and W. Maier, "An Arabic-Moroccan Darija Code-Switched Corpus," p. 6.

[28] R. Tachicart, K. Bouzoubaa, and H. Jaafar, "Lexical differences and similarities between Moroccan dialect and Arabic," in 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, Oct. 2016, pp. 331–337. doi: 10.1109/CIST.2016.7805066.

[29] J. J. Webster and C. Kit, "Tokenization as the initial phase in NLP," in Proceedings of the 14th conference on Computational linguistics -, Nantes, France, 1992, vol. 4, p. 1106. doi: 10.3115/992424.992434.

[30] H. A. Almuzaini and A. M. Azmi, “Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization,” IEEE Access, vol. 8, pp. 127913–127928, 2020, doi: 10.1109/ACCESS.2020.3009217.

[31] K. Taghva, R. Elkhoury, and J. Coombs, “Arabic stemming without a root dictionary,” in International Conference on Information Technology: Coding and Computing (ITCC’05) - Volume II, Las Vegas, NV, USA, 2005, pp. 152-157 Vol. 1. doi: 10.1109/ITCC.2005.90.

[32] “T. Zerrouki, (2010). ‘Tashaphyne, Arabic Light Stemmer/Segment’. http://tashaphyne.sourceforge.net.”

[33] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, “A Study of the Effects of Stemming Strategies on Arabic Document Classification,” IEEE Access, vol. 7, pp. 32664–32671, 2019, doi: 10.1109/ACCESS.2019.2903331.

[34] A. Dey, M. Jenamani, and J. J. Thakkar, “Senti-N-Gram : An n -gram lexicon for sentiment analysis,” Expert Systems with Applications, vol. 103, pp. 92–105, Aug. 2018, doi: 10.1016/j.eswa.2018.03.004.

[35] D. Mohey, “Enhancement Bag-of-Words Model for Solving the Challenges of Sentiment Analysis,” ijacsa, vol. 7, no. 1, 2016, doi: 10.14569/IJACSA.2016.070134.

[36] B. Das and S. Chakraborty, “An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation,” 2018, doi: 10.48550/ARXIV.1806.06407.

[37] B. Oscar Deho, A. William Agangiba, L. Felix Aryeh, and A. Jeffery Ansah, “Sentiment Analysis with Word Embedding,” in 2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST), Accra, Aug. 2018, pp. 1–4. doi: 10.1109/ICASTECH.2018.8506717.

[38] Y. Wang, Z. Li, J. Liu, Z. He, Y. Huang, and D. Li, “Word Vector Modeling for Sentiment Analysis of Product Reviews,” in Natural Language Processing and Chinese Computing, vol. 496, C. Zong, J.-Y. Nie, D. Zhao, and Y. Feng, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 168–180. doi: 10.1007/978-3-662-45924-9_16.

[39] D. Dqg, “Sentiment Analysis Using Support Vector Machine,” p. 5, 2019.

[40] M. Abbas, A. Kamran, Memon, A. A. Jamali, Saleemullah Memon, and Anees Ahmed, “Multinomial Naive Bayes Classification Model for Sentiment Analysis,” 2019, doi: 10.13140/RG.2.2.30021.40169.

[41] W. P. Ramadhan, S. T. M. T. Astri Novianty, and S. T. M. T. Casi Setianingsih, “Sentiment analysis using multinomial logistic regression,” in 2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC), Yogyakarta, Sep. 2017, pp. 46–49. doi: 10.1109/ICCEREC.2017.8226700.

[42] P. Karthika, R. Murugeswari, and R. Manoranjithem, “Sentiment Analysis of Social Media Network Using Random Forest Algorithm,” in 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, Apr. 2019, pp. 1–5. doi: 10.1109/INCOS45849.2019.8951367.

[43] X. Bai, “Text classification based on LSTM and attention,” in 2018 Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, Sep. 2018, pp. 29–32. doi: 10.1109/ICDIM.2018.8847061.

[44] Venkatesh, S. U. Hegde, Z. A. S, and N. Y, “Hybrid CNN-LSTM Model with GloVe Word Vector for Sentiment Analysis on Football Specific Tweets,” in 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, Feb. 2021, pp. 1–8. doi: 10.1109/ICAECT49130.2021.9392516.

1D Convolutional Neural Network for Detecting Heart Diseases using Phonocardiograms

Meirzhan Baikuvekov[1], Abdimukhan Tolep[2], Daniyar Sultan[3],
Dinara Kassymova[4], Leilya Kuntunova[5], Kanat Aidarov[6]
Al-Farabi Kazakh National Unviersity, Almaty, Kazakhstan[1, 3, 6]
Khoja Akhmet Yassawi International Kazakh-Turkish University, Turkistan, Kazakhstan[2]
Academy of Logistics and Transport, Almaty, Kazakhstan[4, 5]

***Abstract*—According to estimations made by World Health Organization, heart disease is the largest cause of mortality throughout the globe, and it is safe to assume that diagnosing heart diseases in their earliest stages is very essential. Diagnosis of cardiovascular disease may be carried out by detection of interference in cardiac signals, one of which is called phonocardiography, and it can be accomplished in a number of various ways. Using phonocardiogram (PCG) inputs and deep learning, the researchers aim to develop a classification system for different types of heart illness. The slicing and normalization of the signal served as the first step in the study's signal preprocessing, which was subsequently followed by a wavelet based transformation method that employs mother wavelet analytic morlet. The results of the decomposition are first shown with the use of a scalogram, afterwards, they are utilized as input for the deep CNN. In this investigation, the analyzed PCG signals were separated into categories, denoting normal and pathological heart sounds. The entire utilized data was divided into two categories as training and test data as 80% to 20%. The developed model demonstrates the degree of clinical diagnosis, sensitivity, specificity and AUC-ROC value. As a result, it has been determined that the proposed method was superior to the mother wavelet as well as other classifier approaches. Consequently, we were able to acquire an electronic stethoscope that has a diagnostic accuracy of more than 90% when it comes to identifying cardiac problems. To be more specific, the proposed deep CNN model has an accuracy of 93.25% in identifying aberrant heart sounds and 93.50% in identifying regular heartbeats. In addition, given the fact that an examination may be completed in only 15 seconds, speed is the primary advantage offered by the suggested stethoscope.**

***Keywords*—Deep learning; CNN; heart disease; phonocardiogram; classification; detection; PCG**

I. Introduction

It is common knowledge that cardiovascular diseases are now among the most serious and widespread. [1] Disorders of the cardiovascular system are the primary cause of mortality on a global scale [2], thus determining the significance of clinical and scientific substantiation and the importance of ensuring early diagnosis of heart diseases [3]. The ease of implementation, functional value, and dependability of these approaches are the most vital qualities to look for in a solution.

There is a variety of approaches to make a heart disease diagnosis [4]. One of the most employed approaches is electrocardiography (ECG) [5]. On the other hand, the electrocardiogram provides a direct description of the status of the heart at the moment of registration. In certain instances, the electrocardiogram does not accurately represent all of the current problems (such as the existence of cardiac disturbances), which necessitates the fulfillment of additional requirements for registration [6].

Phonocardiography, often known as PCG, is a useful adjunct to electrocardiography since it enables investigation and detection of the existence of abnormalities in the cardiac cycle and its valve system [7]. Heart sound measurement that is produced by the diastolic and systolic phase are known as phonocardiograms. The technique involves capturing and analyzing heart sounds, made at various stages, including during its contraction and relaxation. Moreover, this approach can detect the functional condition of cardiovascular illnesses in a manner that is both reasonably priced and not too complicated. In this case, the phonocardiosignal serves as the diagnostic information source, and phonocardiography is the term for the applied recording technique.

In recent times, the field of cardiology has seen a surge in the number of research projects that make use of data analysis. For the purpose of making an accurate diagnosis, the data obtained from PCG and ECG are examined [8-11]. The auscultation of the heart remains an essential diagnostic tool for determining the well-being of the cardiovascular system [12]. As a result, diagnostic techniques have been developed that reasonably minimize the need for non-invasive detection of cardiac disease [13]. The development of prediction models that can determine whether or not a patient has a disease is one of these methods. These models are used to describe the presence of pathology in a patient. The approach of artificial intelligence is the one that works best for these kinds of jobs.

In the field of clinical cardiology, diagnostic and prognostic studies, particularly those involving patients who have cardiac illnesses, the use of artificial intelligence is advancing at an increasingly rapid pace. At the same time, the vast majority of the research that has been conducted on this topic stresses the need of multidisciplinary scientific collaboration as the only means by which improvements in machine learning methods may be implemented.

The paper is structured as follows: In the Section II, a review of the most recent research in this field is presented. Section III contains the discussion of the properties of heart sounds. The Section IV of this paper presents the proposed architecture. In the Section V, we look at the possibility of using machine learning techniques to solve the heart sound categorization issue. The outcomes of the experiment, as well as the future opportunities for the proposed model, are introduced in Section VI. Section VII is the discussion. In the Section VIII, conclusion, we will summarize our study by pointing out some potential future lines of inquiry.

II. Literature Review

The ECG signals has been extensively used in research on the diagnosis of cardiac illness [7–11]. PCG signals are made up of two primary tones, which are denoted by the notations "first sound" (S1) and "second sound" (S2) [12]. When it comes to the aberrant cardiac PCG signals, it includes more than two different sounds and disturbances [13]. A physiological anomaly may cause the blood to circulate through the heart with an irregularity, which can be heard as a murmur. Having a dysfunctional heart valve, a septal defect, or a coarctation of the aorta may all lead to the development of cardiac rhythm disturbances [14]. Applying digital signal decomposing, it is possible to conduct an investigation of the properties of the PCG data [15]. Decomposing digital signals may be accomplished by the use of a wide variety of techniques, such as the Fourier transformation or the wavelet transformation. [16].

A research has been conducted on analyzing cardiac disease based on the PCG signal [17]. In addition, the study that employs wavelet transforms in conjunction with the ML approach for classification has seen considerable appliance [18]. It is now commonly regarded that the continuous wavelet transform (CWT) approach is the best suited for evaluating non-stationary PCG signals (having diverse frequencies and in time) [19-21]. The main advantage of this method is that it uses a continuous wavelet transform rather than discrete. There has also been a study conducted by a number of researchers [22] on the categorization of PCG signals using CWT.

In the previous research, the majority of machine learning approaches were used to the categorization of heart sounds as well as the detection of CVD. Next study describes an innovative approach for identifying the various heart noises [23]. A discrete wavelet transform was applied to one cycle of a heart sound in order to clarify it. In the past, a number of different machine learning techniques were used for the purpose of heart sound feature extraction and classification. These techniques include the empirical wavelet transform [24], combined spectral amplitude and wavelet entropy [25], hidden Markov models [26], support vector machines (SVM) [27], k-nearest neighbours and deep learning models like convolutional neural networks or long-short term memory [28]. There also have been previous applications of spectrograms using wavelets [29] and frequency cepstrum coefficients [30]. The amount of time needed for the pre-processing of signals during early-stage CVD detection is the primary challenge. Additionally, the necessity of feature engineering extends the amount of time needed for signal processing and adds to the complexity of the system, which makes it troublesome to implement in real-time. This problem was handled by the authors of this study utilizing power spectrograms. They eliminated the need for preprocessing and feature engineering of signals, decreased the amount of time required for processing, and created a generalized model by augmenting PCG.

In recent times, the CNNs have achieved a great deal of success in the field of machine learning, and image analysis [31]. Additionally, it has begun to arouse the interests of academics in its usage to ECG and PCG classification [32]. Basically, the CNN is designed to learn high-level representations on its own via the construction of numerous hidden layers and convolutional processes. As a result, the process of extracting features is accomplished in a more effortless way since it eliminates the need to design acceptable features based on feature engineering that is informed by expert knowledge. Researchers such as Acharya et al. [33] and Tan et al. [34] looked into the possibility of identifying CAD by analyzing ECG signals. Both of them, on the other hand, relied on a limited open-source database that included information from 47 participants. The one-dimensional convolutional neural network has been used extensively for the classification tasks in the research on the identification of different heart disorders, with the electrocardiogram signal serving as the input [35].

III. Characteristics of Heart Sounds

A. Heart Sounds

While both S1 and S2 sound like high-frequency noises, listening to them via the stethoscope's diaphragm, one may hear them quite clearly. The usual range for S1 in the heart is between 50 and 60 Hz, whereas the normal range for S2 is between 80 and 90 Hz [36]. The pre-diastolic low signal known as S3 has a bandwidth limit of around 20 to 30 Hz. S4 is also a low signal that occurs towards the conclusion of diastole and may be easily identified with a stethoscope. It takes place at the end of diastole. A frequency of less than 20 Hz [37] characterizes the aberrant S4 waveform.

A number of anomalies cause S1 and S2 to have variable intensities, and as a result, they may sound so quiet that they seem inaudible, despite the fact that they may be heard. Both S1 and S2 do not have consistent frequencies; rather, they move across a variety of frequency ranges depending on the phase of the heartbeat. In order to deal with these constraints on heart sound segmentation, researchers have developed a highly particular technique [38].

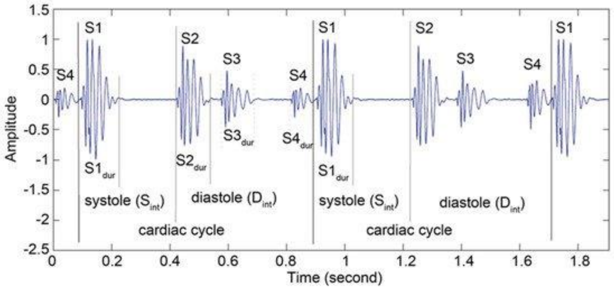


Fig. 1. Heart sounds

The complete view on categories and functions of the HSs are shown in Fig. 1. There is a correlation between one or two HSs and each of the cardiac conditions. After an initial high-pitched sound caused by tricuspid stenosis, all further noises coming from the heart will produce a shrill, higher-pitched noise (TS). The ejection sound is the most well-known example of an early systolic sound, and it comes from an irregular and rapid halting of the semilunar cusps as they open during early systole (ES) [39].

The physicians, who investigate the sounds of an abnormal heartbeat, may provide information that could be helpful for diagnosis.

B. Phonocardiography

Phonocardiography is a non-invasive diagnostic tool used to assess the sounds made by the heart during the cardiac cycle. It involves the use of a specialized microphone called a phonocardiogram or PCG to capture the heart sounds, which are then amplified and recorded for analysis.

The sounds produced by the heart during the cardiac cycle are created by the opening and closing of the heart valves, the turbulence of blood flow, and the movement of the heart muscle. These sounds can be divided into four distinct components known as the first heart sound (S1), the second heart sound (S2), the third heart sound (S3), and the fourth heart sound (S4).

Phonocardiography is useful in diagnosing various cardiac conditions, such as heart valve disease, myocardial infarction (heart attack), and heart failure. It can also be used to monitor the progression of these conditions and evaluate the effectiveness of treatment.

Phonocardiography is typically performed in a quiet room, and the PCG is placed on various locations on the chest to capture the heart sounds. The resulting PCG waveform is then analyzed by a physician or trained technician to identify any abnormalities.

Overall, phonocardiography is a safe and effective diagnostic tool for evaluating heart sounds and detecting abnormalities that may indicate cardiac disease. In Fig. 2, there is an example of the PCG tracing of murmurs that are linked with aortic valvular disease. These murmurs may be heard during an echocardiogram.

At the present time, PCG is not a technique that is typically included in the toolkit of cardiologists. Despite the fact that the technique has been around for many decades, it has been essentially replaced by echocardiography in its diagnostic utility.

On the other hand, there are a number of studies which are conducted to investigate its practicality in the identification of structural heart disease, in particular, congenital heart disease in children. When dealing with patients from this category, one of the most prevalent diagnostic challenges is making a distinction between normal and pathologic murmurs. In this particular setting, PCG has the potential to be an invaluable tool for both pediatric cardiologists and physicians alike. Additionally, the advancement of machine learning and artificial intelligence has resulted in a significant increase in the effectiveness of PCG.

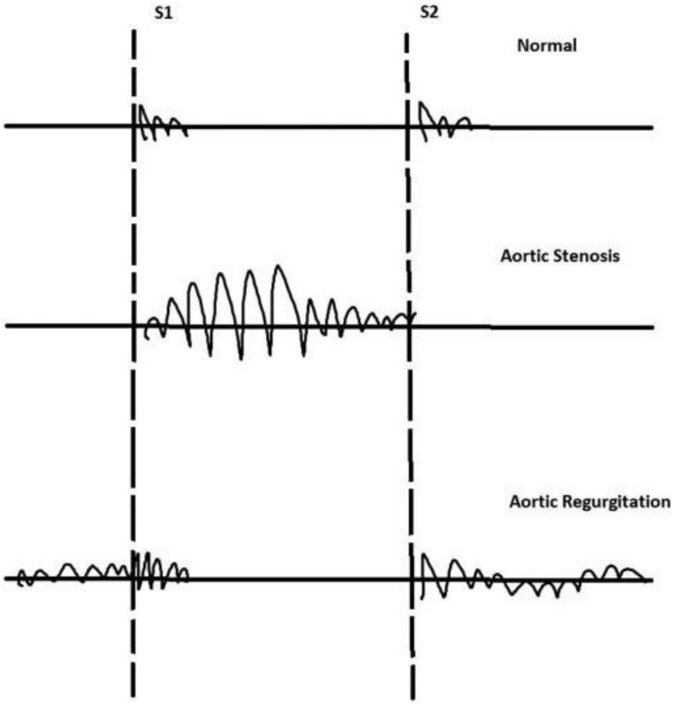


Fig. 2. Model PCG tracings of characteristic murmurs of heart disease.

IV. Materials and Methods

The construction of a deep CNN classification model, along with the assistance of PCG-based heartbeats, which is capable of enabling automated diagnosis of serious cardiovascular disorders is the primary objective of the study. This goal may be accomplished by the combination of a power spectrogram and a CNN. A block diagram is used to illustrate the suggested model in Fig. 3, which provides a graphical depiction of the model.

The suggested approach may be broken down into these three key sections. The first block provides information on the capture of data and the conversion of spectrograms. Following the acquisition of audio cardiac samples of PCG impulses from patients or datasets, the next step is the creation of the data corpus. Additionally, data augmentation and the transformation of audio signals into power spectrograms are included into these blocks. Meanwhile, two distinct spectrogram datasets are currently being created, one of which does not include augmentation, and the other of which includes both augmented and unaugmented spectrogram data. In the second section, the emphasis is mostly placed on the training plan. The spectrogram datasets that were acquired during the first block were split into two groups with a 9:1 split between them. The first step is to train the model, and the second step aims to test the proposed model using 10-fold cross validation. The last block is for the proposed model that contains the proposed CNN model for multi-classification of cardiac abnormalities. This architecture will be helpful for the early diagnosis of four primary types of cardio diseases with the assistance of ground truth established by health personnel.

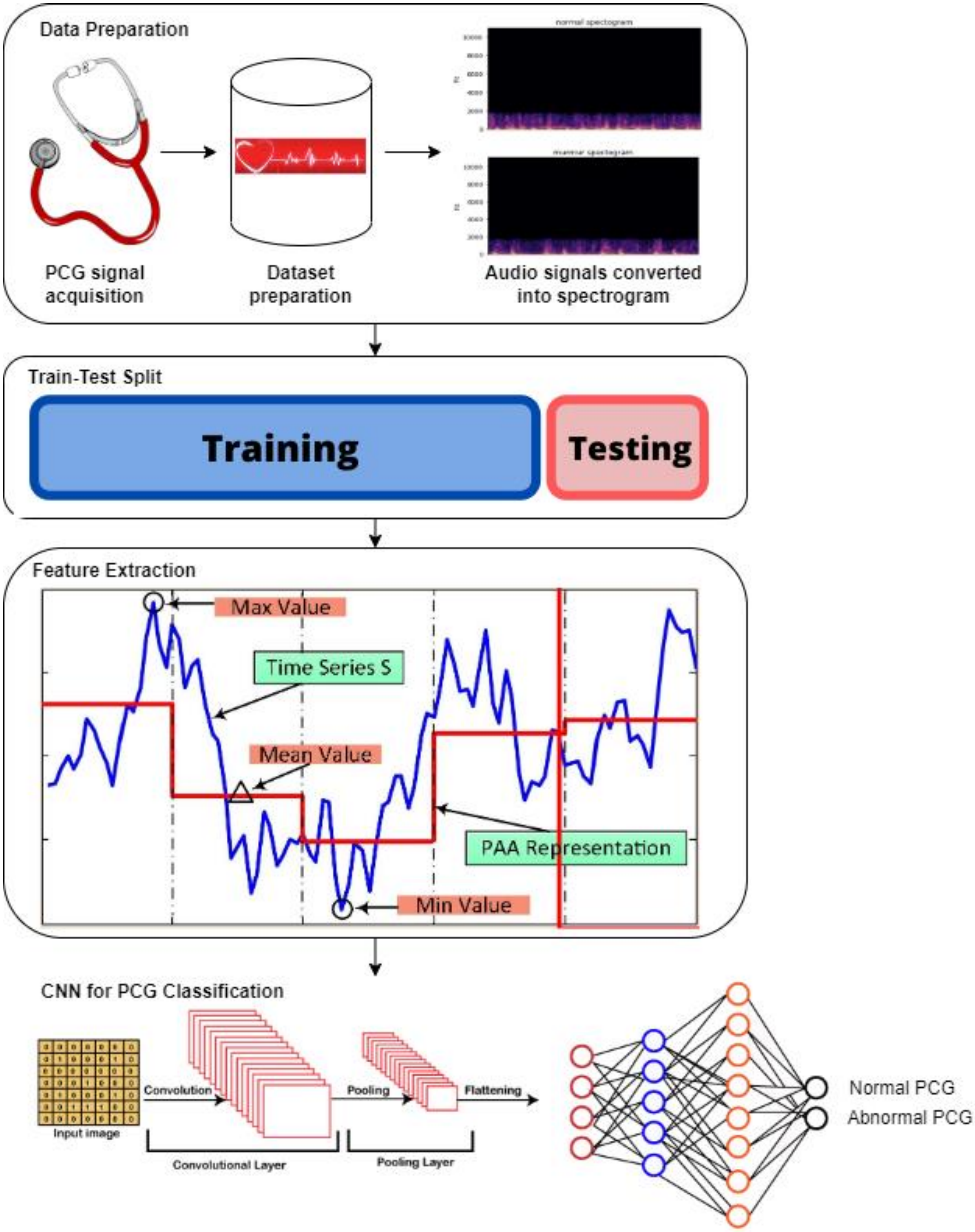


Fig. 3. Flowchart of the proposed framework for heart disease detection.

V. Proposed Model

The proposed model was created to function in tandem of CNN and phonocardiograms as an input data. The expertise gained from previous research led to the conclusion that a combined model typically results in improved operational efficiency. The capacity of such a model to recognize spatial information as well as temporal characteristics polluted within signals has been the primary reason to make such an assumption. The next paragraph will provide a quick description of the architecture that is used by each network. In addition, the comprehensive topology of the network that was used in this investigation is shown in Fig. 4.

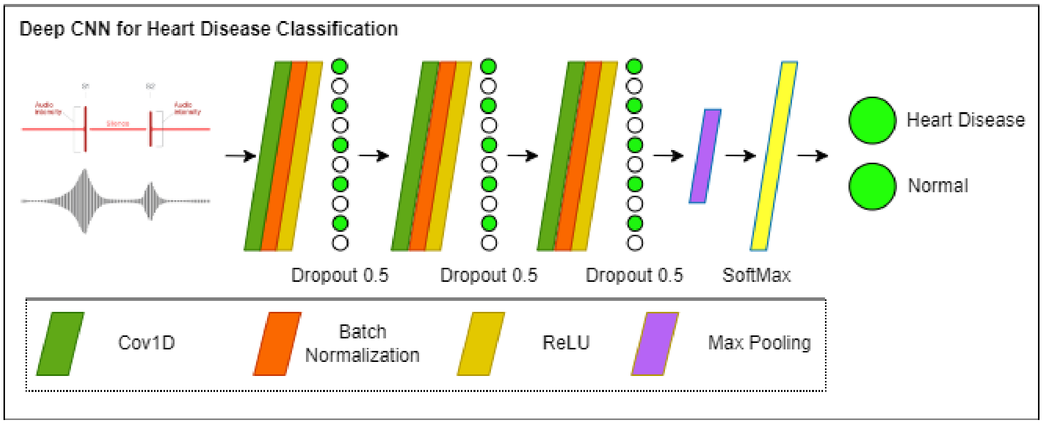


Fig. 4. The proposed deep CNN for heart disease classification.

In the context of deep learning, the term "1D convolutions" refers to the application of a large number of dot products on a window that is comprised of some of the signal. CNNs quickly sprang to prominence as one of the most widely used machine learning methods due to its impressive capacity for automatically identification of necessary characteristics that have been tampered with inside objects [40]. A straightforward CNN architecture is made up of a number of layers. Within the network, each layer is accountable for a certain feature that it provides. When doing convolutions, a large number of filters work together in parallel to extract outputs and then express those extracted outputs as activations. When several convolutions are used, the activations become even more expansive, which ultimately results in the formation of a feature map or vector for the associated input [41]. The equation that is applied in the case of a single convolution of a signal $x_i^0 = \left[x_1, x_2, \ldots, x_n\right]$, where n is the total number of points, is as follows:

$$c_i^{lj} = h\left(b_j + \sum_{m=1}^{M} w_{m=1}^{M} w_m^j x_{i+m-1}^j\right) \tag{1}$$

where l represents the layer index, h represents the activation function, b represents the bias of the j-th feature map, M represents the kernel size, w_m^j represents the weight of the feature map and filter index m-th.

The network that had been constructed over the course of this research included an architecture with a total of three convolutions. These layers were interconnected by additional layers in order to boost their efficiency in extracting features. Firstly, the network was built using a 1D input layer with a dimension of [9600,1] for accepting data from the outside world. The first convolution known as Conv1D, was intended to contain a kernel size of [64,1] and a total of 16 filters in its configuration. In order to simplify the process and cut down on the number of repeats, a stride of [30,1] was used for the first convolutional window, and a stride of [2,1] was applied for the windows that followed. After the convolution step, the batch normalization (BN) and rectified linear unit (ReLU) layers were implemented. Their respective purposes were to equalize the input data across filters and to provide a threshold of zero for values that were less than zero in the produced feature map. In order to glean more in-depth characteristics from the inputs, these three layers—the Conv1D, the batch normalization, and the ReLU—were iterated a total of two more times. The next step was to implement a max-pooling layer, which consisted of a [2,1] kernel that moved with a stride of [2,1], and was done so in order to minimize the dimension of the feature space. It is worth to note that in order to prevent the trained model from becoming over fit, a dropout of fifty percent was implemented after the first two ReLU layers.

VI. Experimental Setup

A. Evaluation Parameters

In the process of assessment, the goal should be to identify as many instances as possible from a community in order to carry out a screening method; hence, the number of false negatives should be kept to a minimum, even if this may lead to an increase in the number of false positives. As a result, it is essential to establish the following three primary parameters: the true positives (TP), the false positives (FP), and the accuracy (ACC). In medical language, the first parameter is denoted by the symbol sensitivity (SEN), and it is defined as follows:

$$TPR = SEN = \frac{TP}{P} \tag{2}$$

In this scenario, count of true negatives is denoted by TP, whereas count of genuine positive cases is marked by P.

The following is an estimate of the second term, which refers to the false positive rate:

$$FPR = \frac{FP}{N} \tag{3}$$

The total number of negative occurrences in the population is denoted by the letter N, while FP stands for the proportion of false positives. This statistic, on the other hand, is best comprehended in terms of the ratio of real negatives to actual negatives. This ratio is referred to as the specificity in medical language, and it is defined as the proportion of genuine negatives to actual negatives:

$$SPEC = \frac{TN}{N} \tag{4}$$

Where the total number of instances that are true negative is denoted by N, and the number of cases that are true negative individually is marked by TN.

Finally, accuracy is what determines whether or not there is a balance between real positives and true negatives. This may be a very helpful statistic in situations in which the number of positive and negative examples is not comparable to one another. This is expressed as the following:

$$ACC = \frac{TP + TN}{P + N} \tag{5}$$

In the end, a ratio between the false positive rate (FPR) and the true positive rate (TPR) was devised so that the efficacy of the algorithm could be evaluated, as well as so that it could be utilized to simultaneously record the highest possible sensitivity and the lowest possible FPR.

$$R = \frac{FPR}{TPR} \tag{6}$$

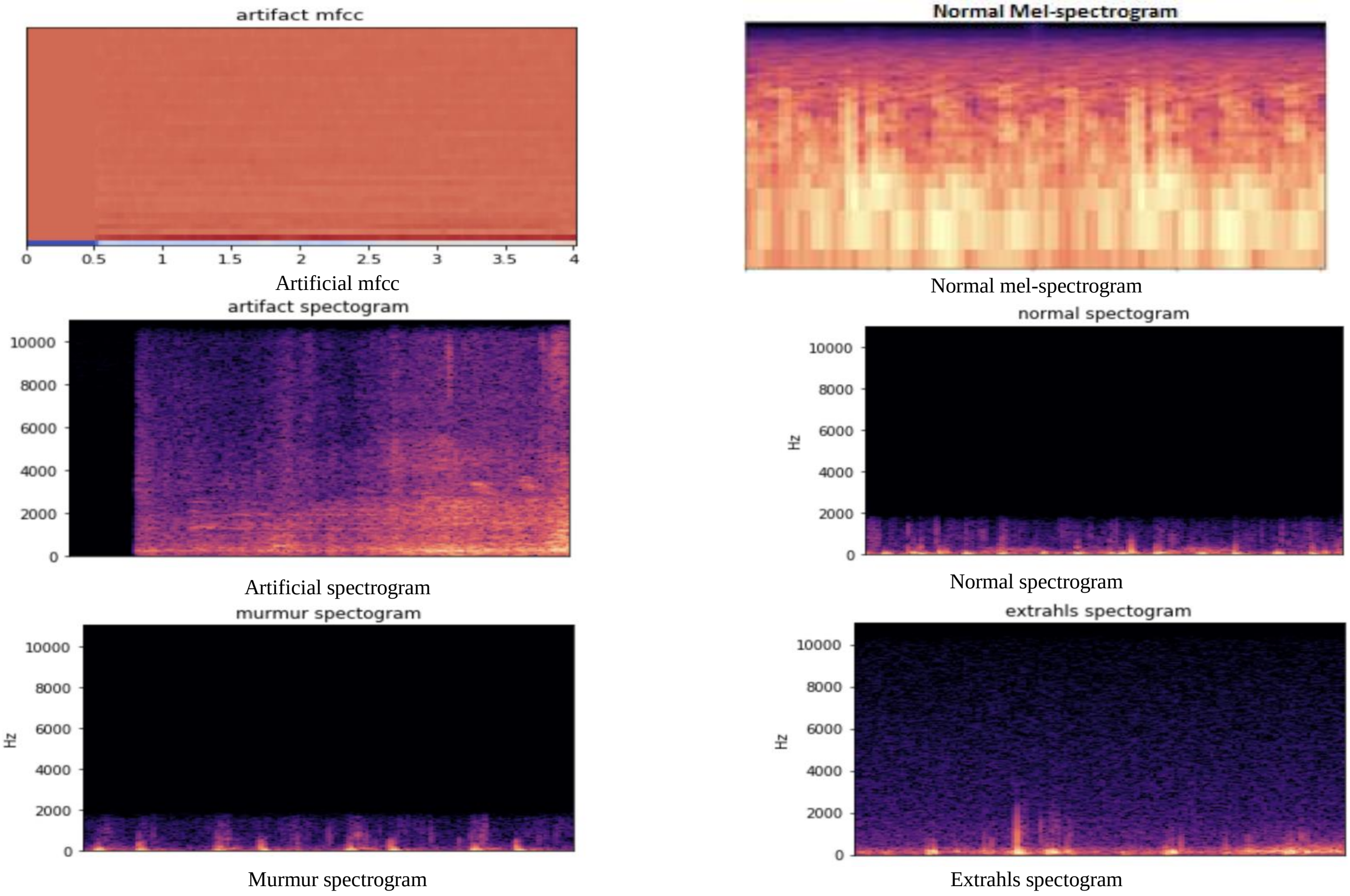
The effectiveness of the method is evaluated based on the criteria that are established, and the present state of the ANN is held steady until those criteria are satisfied. Each iteration of the training process for the algorithm will be referred to by these. The three requirements are as follows:

- When the number of false positives relative to the number of real positives reaches a minimum.
- Until the requirements for complete specificity are fulfilled.
- When the amount of training error has decreased to a point or less.

The first objective that has to be established is whether or not the process can accurately identify OP instances when it's being utilized as a screening approach. Next stage is to evaluate whether or not the method is effective in separating the unhealthy participants from the healthy ones and cutting down on the number of false positives and negatives. The third criteria are the one that determines when the algorithm should halt, and meeting these criteria indicates that overfitting has begun.

B. Experiment Results

Fig. 5 is an illustration that depicts several sorts of heartbeats. Sounds produced by a heart that is working normally; A murmur is an additional sound that occurs when there is a vibration in the blood flow, which in turn causes additional vibrations that are audible; supplemental sound, often known as extrahls There is a huge range of different noises coming from the artifacts.



Artificial mfcc

Normal mel-spectrogram

Artificial spectrogram

Normal spectrogram

Murmur spectrogram

Extrahls spectogram

Fig. 5. Phonocardiogram time series for sixt types of heartbeats.

Fig. 6 illustrates the process of training and validating the model to identify abnormal cardiac rates. The training and validation accuracy results are shown in the figure up to 300 epochs in length.

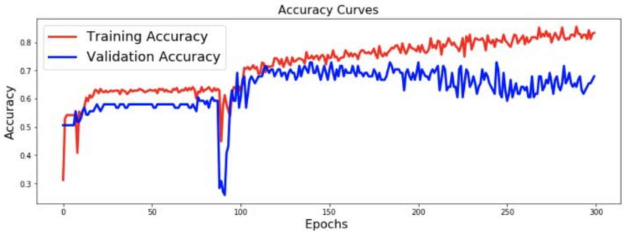


Fig. 6. Training and validation accuracy for hear disease detection.

Fig. 7 illustrates the outcomes of the training loss and validation loss that occurred throughout the training process. After around 300 epochs, the losses demonstrate that they have reached a stable state.

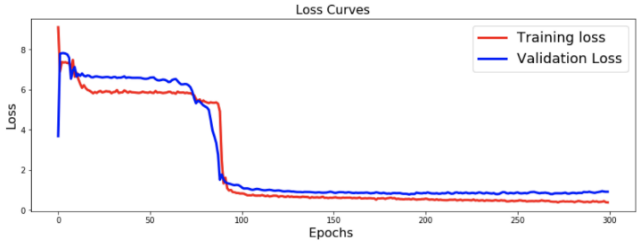


Fig. 7. Training and validation loss for heart disease detection.

The confusion matrix for identifying the five different kinds of heartbeat conditions is shown in Fig. 8. These conditions include murmur, extrahls, extrasystole, artifacts, and normal heartbeat. The findings indicate a high level of accuracy in the categorization of heartbeat sounds and the identification of irregular heartbeats.

The results of classifying normal and pathological heart sounds are shown in the Table I and Table II, respectively. According to the data, illustrated in the tables, the proposed model has a detection accuracy of 93.4% (94.34% for S1, 92.46% for S2 cases) in average, when it comes to irregular heartbeats.

Table II demonstrates results of abnormal cardiac sound detection results for 373 cardiac sounds. As a result, the proposed model has shown average 93.195% (94.18% for S1, 92.21% for S2 cases) accuracy in abnormal cardiac sound detection. The obtained results confirm that the proposed deep convolutional neural network is applicable for real case to classify normal and abnormal heart sounds using phonocardiography signals.

The area under the curve receiver operating characteristics (AUC-ROC) curves that were acquired by the proposed deep convolutional neural network on all five folds of cross-validation are shown in Fig. 9. The findings that were collected indicate that the suggested deep convolutional neural network provides high accuracy in the identification of heart disease, with an AUC-ROC values ranging from 0.979 to 0.988. Overall, the obtained results confirm that the proposed model can be applied for detection and classification of abnormal heartbeats using phonocardiograms.

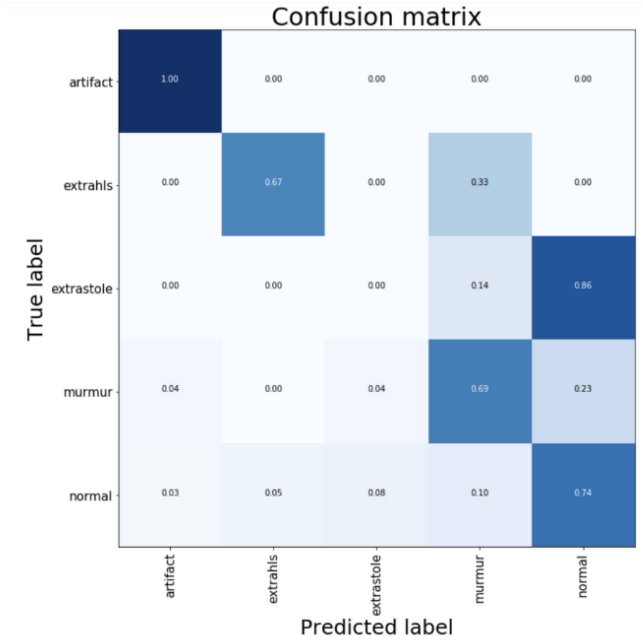


Fig. 8. Confusion matrix for heart disease classification by PCG.

TABLE I. NORMAL HEART SOUNDS DETECTION RESULTS

Normal sounds	True positive	True negative	False positive	False negative	Sensitivity1010	Train accuracy	Test accuracy
S1	374	4	12	10	96.64	97.34	94.34
S2	370	6	10	14	97.32	96.13	92.46
Total	744	10	22	24	96.98	96.735	93.4

TABLE II. ABNORMAL HEART SOUNDS DETECTION RESULTS

Abnormal sounds	True positive	True negative	False positive	False negative	Sensitivity	Train accuracy	Test accuracy
S1	376	6	8	10	97. 88	97.91	94.18
S2	370	8	10	12	97.41	97.19	92.21
Total	746	14	18	22	97.645	97.55	93.195

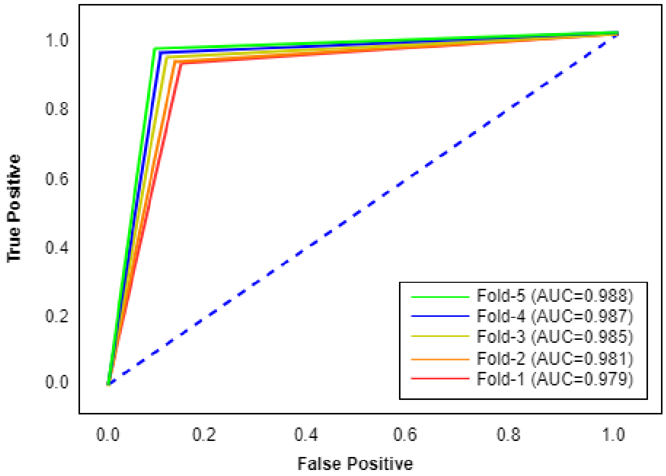


Fig. 9. AUC-ROC curve for 5 fold cross validation.

In Table III, a comparison is made between the proposed deep CNN and the state-of-the-art research that are devoted to the identification of pneumonia using deep learning. According to the findings, the deep CNN that was suggested has a high level of performance across a variety of assessment metrics.

TABLE III. COMPARISON OF APPROACHES FOR HEART DISEASE DETECTION WITH STATE-OF-THE-ART STUDIES

Reference	Applied method	Featucres	Dataset	Results
-	**Ptoposed Deep CNN**	**PCG, S1, S2**	**Own collected dataset**	**93.5% accuracy in normal heartbeats, 93.25% accuracy in abnormal heartbeats**
[33]	SVM	Magnitude	CAPCEA	69% accuracy
[34]	KMC algorithm	-	0.76	70% accuracy
[35]	HKMC, ANN	Magnitude	BMKG, USGS	Between 56% to 72% when M>=6
[36]	ACC algorithm	-	-	41.488 average distance
[37]	HKMC, ANN	-	BMKG, USGS	75% accuracy
[38]	RNN, RF, LP Boost	-	CES and USGS	79% accuracy

VII. DISCUSSION

The use of 1D convolutional neural networks (CNNs) has gained traction in various fields of research, including the healthcare industry. In this regard, the research paper titled "1D Convolutional Neural Network for Detecting Heart Diseases Using Phonocardiograms" presents an interesting approach to diagnose heart diseases using phonocardiogram signals.

The paper highlights the importance of early detection and diagnosis of heart diseases, which can help in preventing fatal consequences. Traditionally, auscultation is used to detect heart diseases, which involves listening to heart sounds through a stethoscope. However, this approach is subjective and heavily dependent on the experience and skills of the healthcare professional.

To overcome this limitation, the researchers proposed the use of 1D CNNs to automatically classify heart diseases using phonocardiogram signals. The study involved collecting phonocardiogram signals from patients with different types of heart diseases, including mitral stenosis, aortic regurgitation, and normal heart sounds. The collected data were preprocessed, and feature extraction was performed using a wavelet transform.

The extracted features were then used to train and test the 1D CNN model. The results of the study showed that the proposed approach achieved an accuracy of 98.6% in detecting heart diseases, outperforming traditional auscultation methods.

The use of 1D CNNs for diagnosing heart diseases using phonocardiograms is a significant advancement in the field of cardiology. The proposed approach has the potential to improve the accuracy and speed of heart disease diagnosis, which can lead to better patient outcomes. Additionally, the approach can be extended to other fields of medicine where sound signals are used for diagnosis, such as respiratory and gastrointestinal diseases.

However, the study had some limitations that need to be addressed in future research. For instance, the study involved a relatively small sample size, and the results need to be validated on a larger dataset. Additionally, the study focused on a limited number of heart diseases, and the approach needs to be tested on a broader range of heart conditions.

Thus, the use of 1D CNNs for detecting heart diseases using phonocardiograms is a promising approach that has the potential to revolutionize the field of cardiology. The study provides a strong foundation for future research in this area, and further studies can build upon these findings to improve heart disease diagnosis and treatment.

VIII. CONCLUSION

The use of a digital PCG signals, which functions as a non-invasive acoustic device for identifying irregularities in the heart, can be beneficial not only for medical professionals but also for ordinary people, in general. Besides, teleconsultations with patients about their cardiac conditions are another possibility that the proposed model can offer. In addition, the early detection of cardiac problems in patients might minimize the need for additional surgical operations, provided that the necessary medical treatments are used.

In this research, we propose a deep CNN model that can be applied in electronic stethoscope that is capable of receiving heart sounds from a patient, processing those sounds, classifying those sounds and as a result, diagnosing the patient in real time, indicating whether or not a patient has a pathology in the heart. The proposed method is able to recognize aberrant cardiac sounds in a short amount of time with a high degree of accuracy, which is the next significant differentiating factor between it and the previous investigations. Moreover, the proposed framework does not transfer server information since everything that is stored on your phone is kept intact. It is imperative that audio recordings should be sent to the physician in order to avoid any issues. The identification of normal heartbeats has reached 93.5% accuracy, while the detection of aberrant heartbeats has reached 93.25% accuracy. Therefore, the issue of categorization of phonocardiograms may now be solved using techniques of machine learning, which is already achievable and offers a high level of efficiency.

In conclusion, we would like to note straightforwardness and feasibleness of the proposed technology as the main benefits. In the future, we will be able to increase the quantity of heart diseases that can be detected by an intelligent stethoscope by increasing the level of accuracy of the stethoscope.

ACKNOWLEDGMENT

This work was supported by the research project “Application of machine learning methods for early diagnosis of pathologies of the cardiovascular system” funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan. Grant No. IRN AP13068289. Supervisor of the project if Batyrkhan Omarov.

REFERENCES

[1] M. E. Chowdhury, A. Khandakar, K. Alzoubi, S. Mansoor, A. Tahir et al., “Real-time smart-digital stethoscope system for heart diseases monitoring,” Sensors, vol. 19, no. 12, pp. 2781, 2019.

[2] M. Elhilali and J.E. West, “The stethoscope gets smart: Engineers from Johns Hopkins are giving the humble stethoscope an AI upgrade,” IEEE Spectrum, vol. 56, no. 2, pp. 36-41, 2019.

[3] M.N. Türker, Y.C. Çağan, B. Yildirim, M. Demirel, A. Özmen et al., “Smart Stethoscope,” In 2020 Medical Technologies Congress, Izmir, Turkey, pp. 1-4, 2020.

[4] Y.J. Lin, C.W. Chuang, C.Y. Yen, S.H. Huang, P.W. Huang et al., “An intelligent stethoscope with ECG and heart sound synchronous display” In 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, pp. 1-4, 2019.

[5] B. Omarov, A. Batyrbekov, A. Suliman, B. Omarov, Y. Sabdenbekov et al., “Electronic stethoscope for detecting heart abnormalities in athletes,” In 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, pp. 1-5, 2020.

[6] V.T. Tran and W.H. Tsai, “Stethoscope-sensed speech and breath-sounds for person identification with sparse training data,” IEEE Sensors Journal, vol. 20, no. 2, pp. 848-859, 2019.

[7] A.A. Shkel and E.S. Kim, “Continuous health monitoring with resonant-microphone-array-based wearable stethoscope,” IEEE Sensors Journal, vol. 19, no. 12, pp. 4629-4638, 2019.

[8] H. Bello, B. Zhou and P. Lukowicz, “Facial muscle activity recognition with reconfigurable differential stethoscope-microphones,” Sensors, vol. 20, no. 17, pp. 4904, 2020.

[9] Murugesan, G., Ahmed, T. I., Shabaz, M., Bhola, J., Omarov, B., Swaminathan, R., ... & Sumi, S. A. (2022). Assessment of mental workload by visual motor activity among control group and patient suffering from depressive disorder. Computational Intelligence and Neuroscience, 2022.

[10] Omarov, B., Altayeva, A., Suleimenov, Z., Im Cho, Y., & Omarov, B. (2017, April). Design of fuzzy logic based controller for energy efficient operation in smart buildings. In 2017 First IEEE International Conference on Robotic Computing (IRC) (pp. 346-351). IEEE.

[11] V. Arora, R. Leekha, R. Singh and I. Chana, “Heart sound classification using machine learning and phonocardiogram,” Modern Physics Letters B, vol. 33, no. 26, pp. 1950321, 2019.

[12] Onalbek, Z. K., Omarov, B. S., Berkimbayev, K. M., Mukhamedzhanov, B. K., Usenbek, R. R., Kendzhaeva, B. B., & Mukhamedzhanova, M. Z. (2013). Forming of professional competence of future tyeacher-trainers as a factor of increasing the quality. Middle East Journal of Scientific Research, 15(9), 1272-1276.

[13] M.N. Homsi and P. Warrick, “Ensemble methods with outliers for phonocardiogram classification,” Physiological measurement, vol. 38, no. 8, pp. 1631, 2017.

[14] G. Son and S. Kwon, “Classification of heart sound signal using multiple features,” Applied Sciences, vol. 8, no. 12, pp. 2344, 2018.

[15] M. Chowdhury, A. Khandakar, K. Alzoubi, S. Mansoor, A. Tahir et al., “Real-time smart-digital stethoscope system for heart diseases monitoring,” Sensors, vol. 19, no. 12, pp. 2781, 2019.

[16] M. Suboh, M. Yaakop, M. Ali, M. Mashor, A. Saad et al., “Portable heart valve disease screening device using electronic stethoscope,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 15, no. 1, pp. 122-132, 2019.

[17] V. Varghees and K. Ramachandran, “Effective heart sound segmentation and murmur classification using empirical wavelet transform and instantaneous phase for electronic stethoscope,” IEEE Sensors Journal, vol. 17, no. 12, pp. 3861-3872, 2017.

[18] J. Roy, T. Roy and S. Mukhopadhyay, “Heart sound: Detection and analytical approach towards diseases,” Smart Sensors, Measurement and Instrumentation, vol. 29, no. 1, pp. 103-145, 2019.

[19] K. Babu and B. Ramkumar, “Automatic Detection and Classification of Systolic and Diastolic Profiles of PCG Corrupted Due to Limitations of Electronic Stethoscope Recording,” IEEE Sensors Journal, vol. 21, no. 4, pp. 5292-5302, 2021.

[20] Altayeva, A., Omarov, B., Suleimenov, Z., & Im Cho, Y. (2017, June). Application of multi-agent control systems in energy-efficient intelligent building. In 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS) (pp. 1-5). IEEE.

[21] Issayev, A., Ortayev, B., Issayev, G., Baurzhan, D., & Gulzhaina, A. (2022). Improving the Supervisory Competence of Future Teacher Trainers with the Help of Innovative Technologies. World Journal on Educational Technology: Current Issues, 14(3), 692-703.

[22] Y. Khalifa, J. Coyle and E. Sejdić, “Non-invasive identification of swallows via deep learning in high resolution cervical auscultation recordings,” Scientific Reports, vol. 10, no. 1, pp. 1-13, 2020.

[23] H. Li, X. Wang, C. Liu, Q. Zeng, Y. Zheng et al., “A fusion framework based on multi-domain features and deep learning features of phonocardiogram for coronary artery disease detection,” Computers in biology and medicine, vol. 120, pp. 103733, 2020.

[24] Kaldarova, B., Omarov, B., Zhaidakbayeva, L., Tursynbayev, A., Beissenova, G., Kurmanbayev, B., & Anarbayev, A. Applying Game-based Learning to a Primary School Class in Computer Science Terminology Learning. In Frontiers in Education (Vol. 8, p. 26). Frontiers.

[25] Omarov, B., Orazbaev, E., Baimukhanbetov, B., Abusseitov, B., Khudiyarov, G., & Anarbayev, A. (2017). Test battery for comprehensive control in the training system of highly Skilled Wrestlers of Kazakhstan on National wrestling “Kazaksha Kuresi”. Man In India, 97(11), 453-462.

[26] Anand, M., Sahay, K. B., Ahmed, M. A., Sultan, D., Chandan, R. R., & Singh, B. (2023). Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. Theoretical Computer Science, 943, 203-218.

[27] R. Banerjee, S. Biswas, S. Banerjee, A.D. Choudhury, and T. Chattopadhyay et al., “Time-frequency analysis of phonocardiogram for classifying heart disease,” In 2016 Computing in Cardiology Conference (CinC), Vancouver, BC, Canada, pp. 573-576, 2016.

[28] Sultan, D., Toktarova, A., Zhumadillayeva, A., Aldeshov, S., Mussiraliyeva, S., Beissenova, G., ... & Imanbayeva, A. (2023). Cyberbullying-related hate speech detection using shallow-to-deep learning. CMC-COMPUTERS MATERIALS & CONTINUA, 74(1), 2115-2131.

[29] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*.

[30] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[31] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[32] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[33] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[34] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[35] Rajeshwari, B. S., Patra, M., Sinha, A., Sengupta, A., & Ghosh, N. (2023). Detection of Phonocardiogram Event Patterns in Mitral Valve Prolapse: An Automated Clinically Relevant Explainable Diagnostic Framework. IEEE Transactions on Instrumentation and Measurement.

[36] Zheng, Y., Guo, X., Yang, Y., Wang, H., Liao, K., & Qin, J. (2023). Phonocardiogram transfer learning-based CatBoost model for diastolic dysfunction identification using multiple domain-specific deep feature fusion. Computers in Biology and Medicine, 106707.

[37] Chen, Y., Su, B., Zeng, W., Yuan, C., & Ji, B. (2023). Abnormal heart sound detection from unsegmented phonocardiogram using deep features and shallow classifiers. Multimedia Tools and Applications, 1-25.

[38] Sletta, Ø. S., Molinas, M., Kumar, M., & Cheema, A. (2023). Classifying Unsegmented Phonocardiogram Signals using Cepstral, Temporal, and Wavelet Scattering Features.

[39] Ge, B., Yang, H., Ma, P., Guo, T., Pan, J., & Wang, W. (2023). Detection of pulmonary hypertension associated with congenital heart disease based on time-frequency domain and deep learning features. Biomedical Signal Processing and Control, 81, 104316.

[40] Ma, P., Ge, B., Yang, H., Guo, T., Pan, J., & Wang, W. (2023). Application of time-frequency domain and deep learning fusion feature in non-invasive diagnosis of congenital heart disease-related pulmonary arterial hypertension. MethodsX, 102032.

[41] Ismail, S., & Ismail, B. (2023). PCG signal classification using a hybrid multi round transfer learning classifier. Biocybernetics and Biomedical Engineering.

CNN-BiLSTM Hybrid Model for Network Anomaly Detection in Internet of Things

Bauyrzhan Omarov[1], Omirlan Auelbekov[2], Azizah Suliman[3], Ainur Zhaxanova[4]
Al-Farabi Kazakh National University, Almaty, Kazakhstan[1]
Institute Information and Computational Technologies CS MES RK, Almaty, Kazakhstan[2]
INTI International University, Kuala Lumpur, Malaysia[3]
M. Auezov South Kazakhstan University, Shymkent, Kazakhstan[4]

***Abstract*—Anomaly detection in internet of things network traffic is a critical aspect of intrusion and attack detection, in which a deviation from typical behavior signals the existence of malicious or inadvertent assaults, faults, flaws, and other issues. The necessity to examine a large number of security events to identify anomalous behavior of smart devices adds to the urgency of addressing the challenge of picking machine-learning and deep learning models for identifying anomalies in network traffic. For the challenge of binary data categorization, a software implementation of an intrusion detection system based on supervised-learning algorithms has been completed. The UNSW-NB15 open dataset, which contains 2,540,044 records - vectors of TCP/IP network connection signals and their associated class labels are used to train and test the system. This research compares different machine-learning models and proposes CNN-BiLSTM hybrid model for IoT network intrusion detection. The metrics for measuring the quality of classification and the running duration of algorithms for different ratios of train and test samples are the result of the built framework testing.**

Keywords—IoT; internet of things; network anomalies; network security; anomaly attack; machine learning; supervised learning; UNSW-NB15

I. Introduction

The Internet of Things (IoT) is a network of electronic devices with built-in technologies that allow them to connect with one another and with the outside world. The Internet of Things (IoT) idea has been ingrained in our daily lives, presenting consumers with new options ranging from home automation to medical equipment [1]. IoT devices can effectively gather, analyze, and send massive volumes of data thanks to ultra-high-speed wireless networks and a sophisticated electronic database. Microelectronic improvements combined with low power consumption have made it increasingly easier to operate IoT devices in remote places with minimum physical oversight and maintenance [2]. Although IoT devices appear to be innocent, they are not without security and privacy concerns, since the present IoT framework contains several risks and vulnerabilities.

According to analysts, the Internet of Things will soon become a part of everyday life. According to IDC, the worldwide market for relevant solutions was valued at $ 646 billion in 2018, and it will surpass the trillion-dollar level by 2022. All of this pushes us to learn more about the security of IoT systems [3].

Automated methods, for managing and interpreting the data are required due to the complexity and diversity of data created by heterogeneous devices. Therefore, machine learning technologies that enable the development of profiles of device behavior in the network, anomalies detection and prediction of abnormal scenarios, claim the role of technology in automatically detecting dependencies and connecting devices [1].

Peripherals, sensors, gateways based on industrial communication protocols, centralized data storage; and end devices users interact with the four major pieces of an Internet of Things system. The addition of big data tools and systems based on machine learning technologies to this setup results in the creation of a new block (Fig. 1) that is responsible for the quality of data and, as a result, the quality of the system's choices and alerts. Furthermore, centralized or cloud data storage expenses are decreased due to adaptive prioritizing and filtering of the information [2-3].

The difficulty with the advancement of attacks is that it is getting more difficult to detect and distinguish between legal and malicious network data. Intrusion detection systems (IDS) [4] do a good job of identifying malicious traffic, but they must be regularly updated with rule sets and upgrades in order to remain relevant when it comes to detecting changing threat vectors. Even if the major corporations disclose fresh sets of regulations on a regular basis, this may not be enough. As a result, the question of employing different methods for identifying irregular incursions becomes significant. The use of machine learning algorithms [5] is one of these ways. Machine learning is used because it can help automate threat processing and keep the system up to date by studying and detecting threats. That is, the software is taught to detect different types of communications in order to classify them and reject or skip them [6].

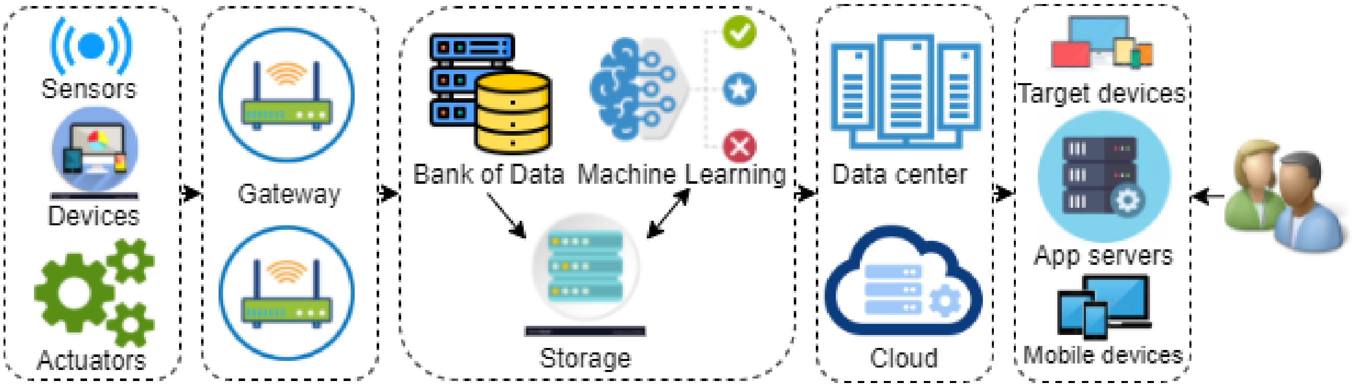


Fig. 1. Comparison of machine learning methods in network anomaly detection.

The following is a reminder of the paper. The next section discusses relevant work on detecting Internet of Things network anomalies using various machine learning algorithms. The third section discusses the problem statement. Section IV depicts the materials and procedures employed in the current study, as well as the research flowchart, dataset, and assessment criteria. In Section V, we provide the outcomes of the experiments and compare machine-learning approaches based on various factors. The results are discussed in Section VI by mentioning obstacles, open questions, and future views. The paper comes to a close in the Section VII.

II. Related Works

In this part, we look at studies that employ machine learning-based techniques to solve the challenge of detecting network abnormalities. Recent research suggests that machine learning (ML) techniques might be ideal for detecting anomalies in network data [7]. For example, Abou Daya et al. [8] used machine learning to leverage correlations between packet and flow-level data. On many anomaly detection tasks, Gaddam et al. [9] offered a solution that combined K-means clustering with an ID3 decision tree. For DDOS detection in self-defined networks, Alamri and Thayananthan [10] used XGBoost [11]. Shone et al. [12] developed a deep autoencoder (NDAE) for unsupervised feature learning and intrusion detection utilizing stacked NDAEs. To learn from anomalous traffic, Zhange et al. [13] created a semi-supervised learning system. For intrusion detection, Ullah et al. [14] developed an LSTM-based model using autoencoders. An XGBoost-DNN model was presented by Devan et al. [15] to identify cyber assaults. To solve the unbalanced class problem, Du et al. [16] integrated reinforcement learning with the SMOTE method. For the network anomaly detection problem, we now look at each machine learning approach independently.

K nearest neighbour (K-NN). The KNN technique is one of the most basic and widely used nonparametric methods. It estimates the approximate distances between the input vectors' different points, then assigns the unlabeled point to the class of its K-nearest neighbor. When building a KNN classifier, the parameter (K) is crucial, and various values (K) might have varied outcomes. If K is big, the neighbors utilized for prediction will take a long time to classify and have an impact on accuracy [17].

Zhu et al. offer a Grid-based Approximate Average Outlier Detection (GAAOD) framework to maintain KNN-based anomaly recognition in network traffic streaming data [18]. In the first stage, the proposed framework presents a grid-based coefficient to control resulting data. It can self-adaptively configure the resolution of units, and reach the target of effectively filtering items that cannot become outliers. In the second stage, GAAOD framework utilizes a min-heap-based method to calculate the upper-/lower-bound distance between items and their k-th nearest neighbors. In the third stage, the author applies a k-skyband based method to support anomaly items and possible anomaly items. Technical outcomes prove the effectiveness and high correctness of the proposed approach.

Bayesian networks. A Bayesian network (BN) is a mathematical model for encoding probabilistic correlations between variables. This strategy is typically used in conjunction with statistical schemes for intrusion detection. It has several benefits, including the capacity to encode interdependencies between variables and predict occurrences, as well as the ability to incorporate existing knowledge and data [19].

The BN system, according to Lotfollahi et al. [20], provides the necessary mathematical foundation for making an apparently complex operation simple. They expected that by comparing the measurements of each network traffic sample, BN-based IDS would be able to identify assaults from regular network activity. Mohammed et al. [21] employed a controlled Naive Bayesian classifier and 248 function streams to distinguish between several sorts of information, including packet length and delivery time, as well as a variety of TCP headers. To find strong functions, feature selection correlation was performed, and it revealed that just a small subset of fewer than 20 features is required for accurate classification.

Neural networks (NNs). The behavior of numerous users and daemons in a system is predicted by NNS. If correctly planned and executed, NNS can alleviate many of the issues that rule-based systems have. The key benefit of NNS is their tolerance for erroneous data and information, as well as their capacity to generate solutions without prior understanding of data patterns [22].

This, paired with their capacity to generalize the facts under investigation, qualified them for IDS. Data representing attacks and non-attacks must be fed into the machine learning model for automated modification of network coefficients during the training stage in order to use this technique to IDS. The most prevalent types of regulated neural networks are multilayer perceptron (MLP) and radial basis function (RBF) [23].

Only linearly separable instances of sets may be systematized using MLPs. The perceptron will be able to discover a solution if a straight line or a plane can be drawn to partition input examples into permissible categories, and the input instances are linearly separable. Learning will never reach the point where all examples are adequately systematized if the instances are not linearly separable. To address this issue, multilayer perceptrons (artificial neural networks) were developed.

There have been studies that have used multilayer perceptions to develop intrusion detection system for network traffics, which has the capacity to identify both legitimate and malicious connections, such as [24]. MLPs of three and four layers of a neural network were used to implement them.

Another prominent form of neural network is the Radial Basis Function (RBF). RBF networks are significantly quicker than back propagation because they accomplish classification by measuring the distance between inputs and RBF centers of hidden neurons. They are best suited for problems with a high sample size.

Decision tree (DT). Quinlan [25], for example, characterized decision trees as "a useful and widely used categorization and forecasting method. A decision tree is a tree made up of three primary parts: nodes, arcs, and leaves. Each node has a unique characteristic that is the most informative of the features not yet examined on the path from the root. Each sheet is allocated to a category or class, and each arc from the node identifies the values of the node attribute. Starting at the root of the tree and working down until a node leaf is reached, a decision tree may be used to categorize a data point. The data point is classified using the node sheet. Quinlan's ID3 and C4.5 are the most widely used decision tree implementation alternatives."

As an intrusion detection model, Davahli et al. [26] recommended employing decision trees (DT) and the support vector machine (SVM). They also created a hybrid DTSVM technique that employs both SAM and DT as fundamental classifiers. Decision trees were adapted by Ghanem et al. [27] for DDoS attacks, R2 as well as U2R assaults, and scanning attacks. The ID3 method is used as a learning algorithm to generate a decision tree automatically.

Support Vector Machine (SVM). Cortes and Vapnik [28] proposed the support vector machine (SVM) technique. The input vector is transformed into a multidimensional feature space by SVM, which then finds the best separating hyperplane in a high-dimensional feature space. Furthermore, because the boundary solution, i.e. the separating hyperplane, determines the reference vector rather than the whole training sample, it is impervious to significantly deviating values. SVM is especially well-suited to binary classification. That is, to distinguish between two sets of training vectors with distinct class labels. The penalty function, which is a user-defined parameter in SVM, is also available. This helps users to strike a balance between the amount of samples and the erroneous solution border width categorization.

Mukkamala et al. [29] used SVM "core classifiers and classifier design approaches to apply to the network with the task of identifying abnormalities." They looked at the impact of core type values and parameters on the Support Vector Machine's (SVM) intrusion classification accuracy. The PSA-SVM model was suggested by Gauthama Raman et al. [30], where the PSO standard is used to establish the free parameters of the support vectors and the binary PSO is utilized to produce the optimal subset function in the intrusion detection system. Eskandari et al. [31] provided a model of an intrusion detection system based on network traffic behavior and message analysis and categorization. Anomalies are detected using two artificial intelligence methods: the Kohonen neural network (KSN) and support vectors (SVM).

Deep learning. Recurrent neural networks paired with long short-term memory are investigated in this research [32] for their ability to identify Internet of Things malware. Models constructed using more traditional machine learning techniques are compared to the results of the experiment. These techniques include the Support Vector Machine, the Naive Bayes classifier, the random Forest, adaptive Boosting, and the k-nearest neighbors algorithm. According to the findings of the inquiry, the technique based on deep learning gives the greatest outcomes. Other deep learning models were not compared since there was none.

As described in the study [33], a variety of deep learning methods for recognizing DDoS attacks are being researched, including multilayer perceptron, convolutional neural network, RNN-LSTM, CNN+LST ensemble, and RNN-LSTM and CNN. Their performance is compared to that of standard machine learning algorithms such as the support vector machine, Bayesian classifier, and random forest, among others. They reach the conclusion that deep learning approaches, particularly recurrent networks, are more successful than standard methods.

It is proposed in the research [34] that an auto-encoder and a deep neural network with direct communication be utilized to develop their own anomaly detection solution for industrial Internet of Things systems that they feel will be effective. When the properties of the newly constructed model are compared to those of many previously developed anomaly detection approaches, such as the deep trust network [35], the recurrent network [36], the DNN [37], and the Ensemble-DNN [38], the results show that the newly constructed model outperforms them all. Meanwhile, these models were evaluated on multiple subsets of the source data as well as on a range of different hardware and software platforms at various points in time, according to the research.

III. Problem Statement

It is required to define the mathematical and software techniques in order to analyze abnormalities in network traffics. Anomaly detection, according to our findings, leads to a data categorization issue. We divide the traffic into two categories: regular traffic and abnormal traffic. As a result, the issue is a binary classification problem. We will utilize basic mathematical methods to identify severe fluctuations in the graph, such as:

$$S = \int_{t_1}^{t_2} |x'(t)| dt \qquad (1)$$

This is the total of all potential variations from time t1 to time t2. The formula will look like this since the function is discrete:

$$S = \sum_{t=t_1}^{t_2-1} |x(t+1) - x(t)| \qquad (2)$$

In the next part, we utilize machine learning approaches to discover IoT network abnormalities and assess them using various measurement parameters for the supplied dataset.

IV. Materials and Methods

In this part, we describe the whole outline of the Machine Learning (ML)-based system that has been recommended for fault and attack differentiation. According to the results presented in Section III, it might be difficult to differentiate between assaults that behave similarly to node issues at the receiving ends due to the fact that their impact on the communication channel is identical. If we monitor the state of the channel, there is a chance that we will be able to record the state transition activities that the attackers execute in order to produce a number of attacks. We came to the conclusion that the best way to overcome the challenge of differentiating between assaults and difficulties on the receiving end was to directly monitor the channel data. Next, in order to differentiate between the two abnormality groups based on channel qualities, we used machine learning models to fit those measurements (and hence channel state).

A. *Methodology*

As can be seen in Fig. 2, the whole process may be broken down into three distinct stages. In the initial step of development, the system is modeled for the normal, faulty, and attack classes respectively. As a consequence of this, the second step entails conducting a number of execution scenarios with the purpose of constructing datasets that define the behavior of the system under normal, faulty, and attack settings. In the third phase, the gathered datasets are put to use in order to assess a number of supervised machine learning algorithms for classification purposes in relation to the differentiation issue.

As a result, the proposed framework is flexible in that it may be used to investigate multiple classes of defects and assaults in a variety of experimental setups, as well as to evaluate the datasets generated by different supervised machine learning algorithms. Furthermore, by concentrating solely on the features of the communication channel, this framework is insensitive to the characteristics of the devices employed in any cyber-physical system of any type.

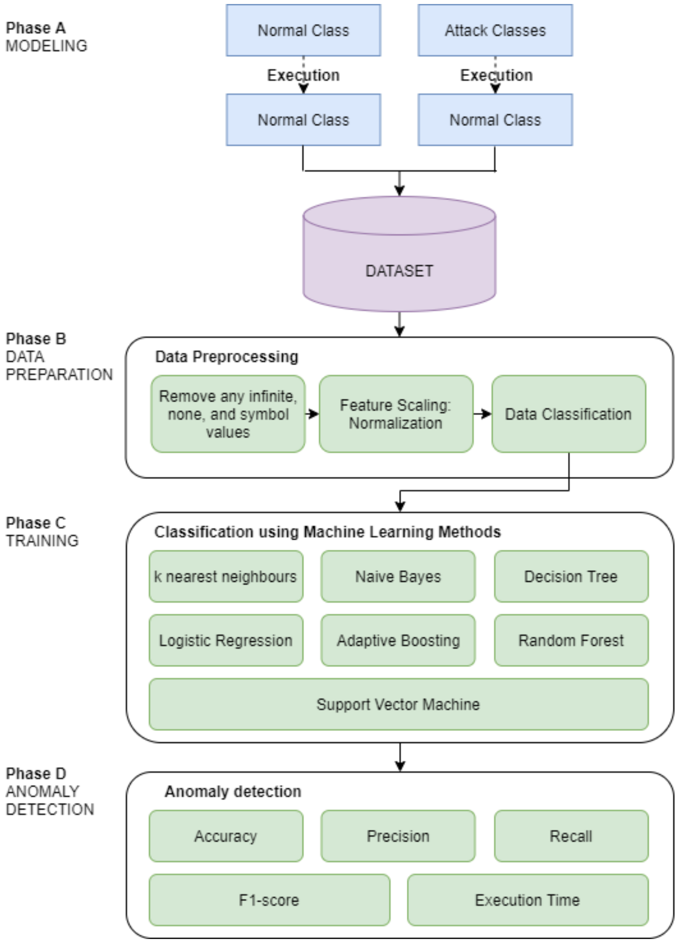


Fig. 2. Comparison of machine learning methods in network anomaly detection.

This part focuses on this general framework and goes through the anomaly classes, various ML classification techniques we are looking at, and the evaluation metrics we are using to evaluate the algorithms.

B. *Data*

An open data collection UNSW-NB15 [39, 40] was chosen as experimental data for the examination of DNN models in the tasks of detecting network abnormalities in the Internet of Things. It contains 2,540,044 records - vectors of TCP/IP network connection attributes and their related class labels. Network packets in this collection of data provide information about typical network activity as well as nine different forms of attacks: fuzzers, analyzers, backdoors, denial of service (DOS), exploits, generic, Reconnaissance, shellcode, and worms. UNSW-NB15 data contains 47 characteristics, such as IP addresses, port numbers, transaction bytes, and so on [41], as well as two class labels — the attack category and the connection abnormality label — for training and testing intrusion detection systems. The first 35 characteristics are for integrating data packet information, while the remainder is for connection circumstances.

The process of detecting deviations from the system's typical profile is known as anomaly detection. To detect anomalies in UNSW-NB15 network data, a binary classification is utilized, with the connection anomaly criteria serving as a class label, with 0 corresponding to the normal profile and 1 corresponding to anomalies.

C. The Proposed CNN-BiLSTM Hybrid Model

This study uses BiLSTM as the model's foundation since it can successfully extract data characteristics. It can perform high-level abstraction and nonlinear transformation of intrusion data, evaluate two-way data information, and give more fine-grained computation. BiLSTM is an upgraded variant of LSTM. Fig. 3 displays the CNN-BiLSTM structure that has been suggested.

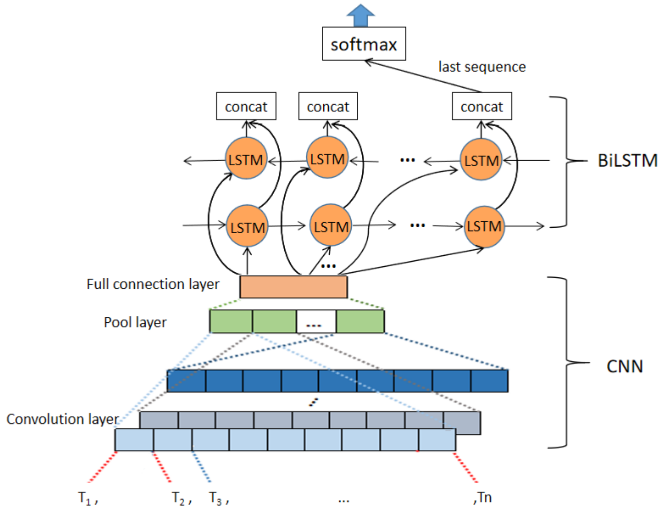


Fig. 3. Proposed CNN-BiLSTM architecture.

The distribution of the data in the neural network may alter after BiLSTM analysis of the data. Batch Normalization process is used to address the inconsistent data distribution problem while deep neural networks are being trained. Deep neural network training may be sped up by batch normalization. After the nonlinear transformation of the activation function, it normalizes the input data of the preceding layer, ensuring the network's trainability and enabling the neural network to continuously maintain the consistency of the input data distribution, thereby minimizing significant changes in the network's node distribution. The network's convergence rate may be accelerated while maintaining the neural network's capacity for representation.

In the IoT, information flow often exhibit significant local correlations, and some of this information even directly correlates with information across a long span. The Bidirectional LSTM network can handle this time-sequential data successfully by using an algorithm to filter out the important and irrelevant information from the data. Hence, in order to enhance the detection capabilities of the detection system, this study incorporates the BiLSTM network based on CNN. The suggested CNN-BiLSTM IoT intrusion detection model is shown in Fig. 4.

The first thing that has to be done in the detection model is to do some kind of preprocessing on the original data set. The process begins by converting all of the data into numerical data, which is followed by the standardization and normalization steps. The data that has been processed will now go into the record representation layer. When the data has been preprocessed, the record presentation layer will use an embedded representation for each individual item of data. The output feature is generated once the features of all the data have been twisted using the convolution check.

While obtaining the feature sequence, all of the features acquired by convolution are layered on one another. The pooling layer receives the feature map from the convolution layer after it has been processed by the convolution layer to produce the feature map. The feature sequences are then pooled together by the pooling layer. The eigenvector may be obtained by first dividing the input data into M blocks, then taking the maximum value for each block, and then splicing all of the results together. This process is known as maximum pooling.

After the pooling of the data in the layer for pooling the data, the acquired feature sequence is then fed into the layer for the BiLSTM. The long-term memory layer is made up of two LSTM modules that are facing in opposite directions, and various weights that are shared between them. The BiLSTM module will choose and then delete each individual piece of data in sequence.

Upon the completion of the data processing, the CNN-BiLSTM network acquires the data features. In order to integrate these feature sequences, a full connection layer is used, and the results that are acquired from the utilization of the full connection layer are then entered into the softmax classifier. In the last step, the results of classifying each piece of information are acquired.

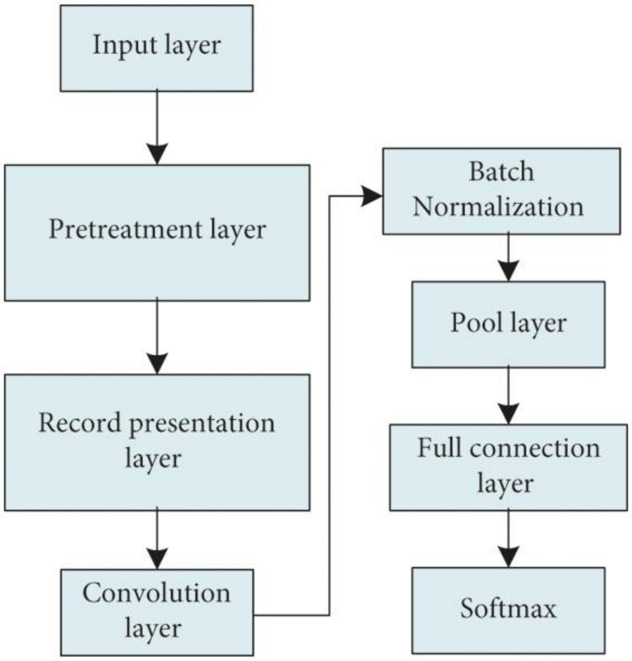


Fig. 4. Intrusion detection model of industrial Internet of Things based on CNN-BiLSTM.

D. Evaluation Metrics

In machine learning tasks, the following metrics are most often used to evaluate the effectiveness of constructed models [42]: accuracy (precision), completeness (recall), F-measure (F-score), ROC-Curve (from the English Receiver Operating Characteristic curve - error curve), AUC-ROC and AUC-PR (from the English Area Under Curve - the area under the error curve and the area under the precision-recall curve) [43].

After classification, to obtain four types of results is possible. Table I demonstrates different classification parameters, where yy' is the algorithm response on the object, and yy is the true class label on this object.

TABLE I. DATASET DESCRIPTION

Dataset	Y=1	Y=0
Y'=1	True Positive (TP)	False Positive (FP)
Y'=0	False Negative (TN)	True Negative (TN)

Overall accuracy or accuracy is an indicator that evaluates the correctness of anomaly detection. The overall accuracy determines what percentage of the data the system or algorithm can classify correctly. Calculated by the formula:

$$Accuracy = \frac{TP + TN}{Pos + Neg} \quad (3)$$

The precision of a classification system may be measured by the percentage of items that are labeled positive by the classifier and are, in fact, positive:

$$precision = \frac{TP}{TP + FP} \quad (4)$$

Completeness (recall) shows the proportion of correctly labeled positive objects among all objects of a positive class:

$$recall = \frac{TP}{TP + FN} \quad (5)$$

The completeness of the data is not affected by the distribution of the data, in contrast to accuracy. Completeness does not represent the number of things that are wrongly identified as positive, and accuracy does not provide any information about the number of positive objects that are incorrectly identified [44].

The (F-score, Fß) combines the above two metrics into one measurement parameter:

$$F_{\beta} = \left(1 + \beta^2\right) \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \quad (6)$$

Where β - takes values in the range $0 < \beta < 1$ if accuracy is given priority, and $\beta > 1$ if completeness is given priority.

The F-measure reaches a maximum with completeness and accuracy equal to one, and is close to zero if one of the arguments is close to zero.

The ROC curve, also known as the error curve, is a graph that shows the relationship between the algorithm's sensitivity (TPR, True Positive Rate) and the proportion of objects in a negative class that the algorithm predicted incorrectly (FPR, False Positive Rate) when the threshold of the decisive rule is changed [45]:

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

In addition to these evaluation parameters, we used area under the curve receiver operating characteristics (AUC-ROC) parameters.

V. EXPERIMENTAL RESULTS

Data preparation (1) entails preparing an input data set, which includes 47 indicators of network connections and class labels, in a manner that can be fed into the studied models. To nominal-type information like IP addresses, protocol names, and data transfer services, one-hot encoding, a method of describing categorical variables in the form of binary vectors, is used. After that, all sign values are normalized to the range [0...1]. Because an imbalance between the values of features can create model instability, degrade learning results, and slow down the modeling process, data normalization is done. A total of 80% of the original data set (1,547,081 records) is chosen for model training, while 20% (386,771 records) is chosen for model testing.

Fig. 5 and Fig. 6 demonstrate the model accuracy and model loss for the proposed CNN-BiLSTM. Fig. 5 demonstrates accuracy of the proposed CNN-BiLSTM model. The results show that the proposed model show high accuracy during the tested 12 epochs of training. The results show that, the proposed model is applicable for practical cases to detect IoT network intrusions or anomalies.

Fig. 6 demonstrates the model loss of the proposed deep CNN-BiLSTM model for intrusion detection problem in internet of things network. There, we show the results for 12 epochs of learning. The results show that the model loss is low from the 4th epochs of training. The result of 12th epochs demonstrates little loss and high accuracy, respectively.

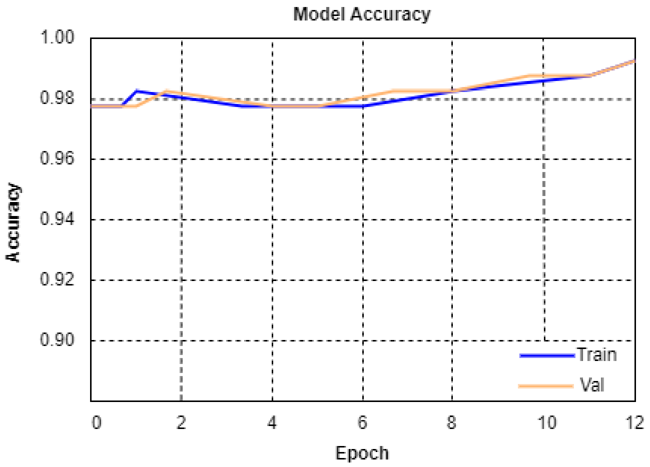


Fig. 5. CNN-BiLSTM Model accuracy.

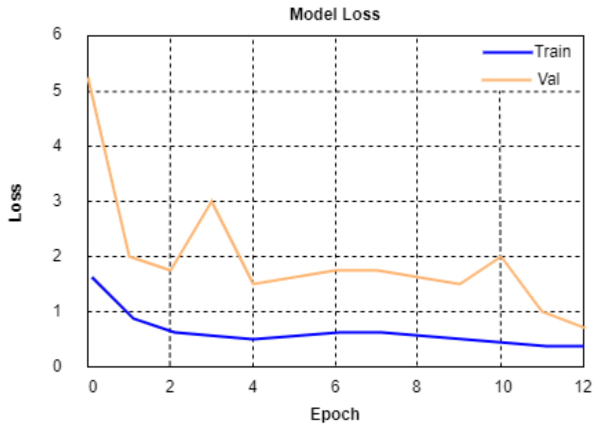


Fig. 6. CNN-BiLSTM Model loss.

In the IoT network anomalies detection challenge, Table II shows a comparison of the investigated machine learning algorithms and training time values. As shown in the table, support vector machine (SVM) has a high level of accuracy in detecting network anomalies, but it takes a long time to train. As a result, it is unfit for real-time anomaly identification. In comparison, for the provided dataset, logistic regression is the best approach for detecting network abnormalities in internet of things.

Fig. 7 and Fig. 8 demonstrate performance evaluation and training time comparison in graphical form. In Fig. 7, we compare six machine learning methods by four evaluation parameters as accuracy, precision, recall, and F1 score. As it is illustrated in the figure, random forest, Adaptive Boosting (AdaBoost), and k nearest neighbours (KNN) show higher performance in the measured evaluation parameters than the other machine learning methods. Nevertheless, we can also consider training and testing time of each algorithm to understand how fast the applied method copes with the given problem.

Fig. 8 demonstrates training times of each algorithm in network anomalies detection. For convenience, the figure is illustrated in logarithmic scale. If we compare the three methods that shown high performance, KNN has the longest training time, Adaboost and Random Forest gives shorter training time, that makes the methods suitable for practical use.

TABLE II. COMPARISON OF THE PROPOSED MODEL WITH MACHINE LEARNING METHODS

Classification method	Accuracy	Precision	Recall	F-measure	Execution Time, s
Proposed CNN-BiLSTM	**96.28**	**96.17**	**95.14**	**95.09**	**47.16**
KNN	85.02	85.12	85.02	85.02	12698.27
Naïve Bayes	82.05	83.29	82	82.56	175.48
DT	81.17	81.17	81.06	81.23	846.24
SVM	88.26	88.32	88.26	88.26	10624.85
Logistic Regression	85.83	85.89	85.83	85.85	178.56
AdaBoost	87.34	87.34	87.34	87.34	965.45
Random Forest	87.62	87.66	87.66	87.66	574.20

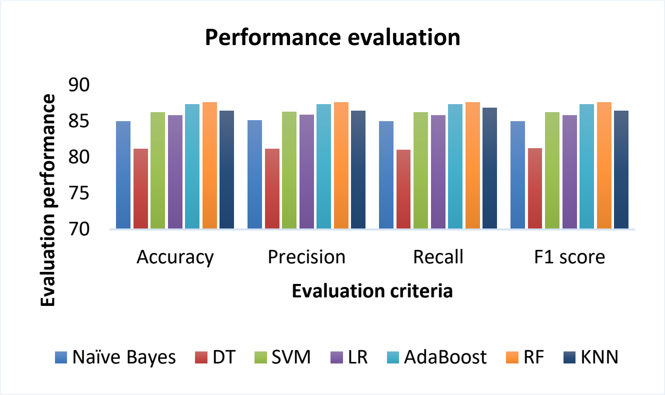


Fig. 7. Comparison of machine learning methods in network anomaly detection.

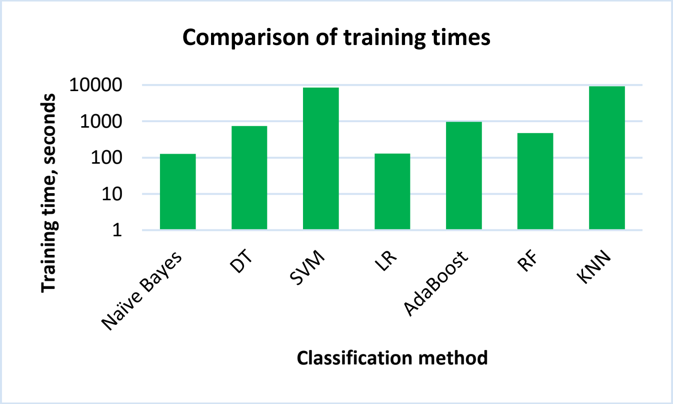


Fig. 8. Comparison of training times (in logarithmic scale).

Fig. 9 indicates the ROC curves of each method. The applied methods show high results in the given problem. The results show, that machine learning techniques can be successful in internet of things network traffic anomaly detection.

Thus, we compared different machine learning methods for network anomalies detection problem in two types of performance parameters. The results show that Logistic Regression is more suitable for practical use than the other methods in intersection of two indicators. It has comparatively short training time and high accuracy in network anomalies detection.

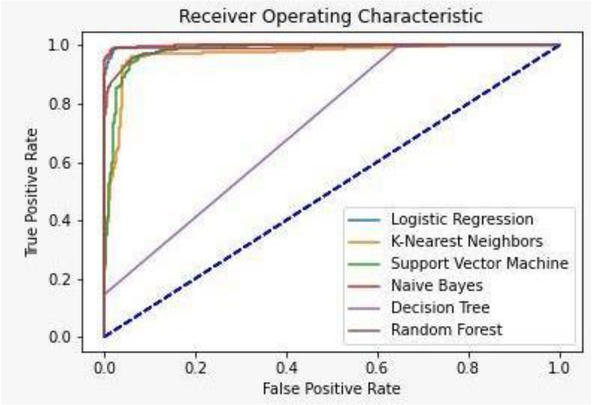


Fig. 9. The ROC curve of the applied methods for IoT anomalies detection.

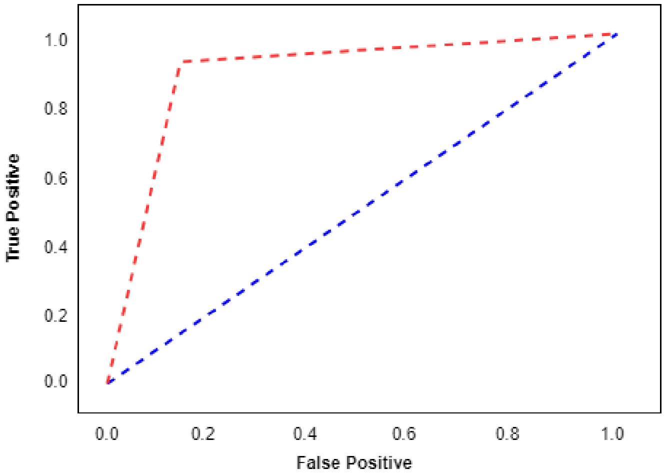


Fig. 10. The ROC curve of the applied method for IoT anomalies detection.

Fig. 10 demonstrates the ROC curve of the proposed CNN-BiLSTM for anomalies detection in internet of things. The obtained results show that, the AUC-ROC curve show high value. The obtained results of all evaluation parameters demonstrate that the proposed CNN-BiLSTM model is applicable for practical cases.

VI. Discussion

The goal of this research is to explore machine learning and deep learning models for identifying abnormalities in Internet of Things network data and develop a new deep learning model for the given problem. Deep learning models were evaluated utilizing a single set of hardware and software, as well as equal sections of the UNSW-NB15 dataset for training and testing. Test models include logistic regression, random forest, KNN, decision tree, Naive Bayes, SVM, and Adaptive Boosting. The built models have high rates of IoT network anomaly detection accuracy, ranging from 80% to 88%. The article proposes CNN-BiLSTM hybrid model for detection of anomalies in internet of things network. The proposed deep model shown about 96% accuracy. In addition, the paper chooses the best machine learning model based on the amount of time it takes to train the model and the importance of identifying abnormalities in internet of things network traffic.

It is intended to continue examining the properties of models employed in cybersecurity jobs in the future. One of the upcoming research objectives is to look at the effect of internet of things network traffic topology on the performance metrics of deep learning models [46]. Based on the findings, a deep CNN-BiLSTM strategy is proposed for recognizing and linking security incidents.

VII. Conclusion

The goal of this research is to look at machine learning models for identifying abnormalities in Internet of Things network data. Deep learning models were evaluated utilizing a single set of hardware and software, as well as equal sections of the UNSW-NB15 dataset for training and testing. Test models include logistic regression, random forest, KNN, decision tree, Naive Bayes, SVM, and Adaptive Boosting. The built models have high rates of network anomaly detection accuracy, ranging from 80% to 88%. The article offers suggestions for selecting the best deep learning model based on the amount of time it takes to train the model and the importance of identifying abnormalities in network traffic.

It is intended to continue examining the properties of models employed in cybersecurity jobs in the future. One of the upcoming research objectives is to look at the effect of network traffic topology on the performance metrics of deep learning models. Based on the findings, it is proposed to build a deep learning-based strategy to recognizing and linking security incidents.

References

[1] A. Nauman, Y. Qadri, M. Amjad, Y. Zikria, M. Afzal et. al. “Multimedia Internet of Things: A comprehensive survey,” IEEE Access, vol. 8, no. 1, pp. 8202-8250, 2020.

[2] Z. Lv, L. Qiao, J. Li, J. and H. Song. “Deep-Learning-Enabled Security Issues in the Internet of Things,” IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9531-9538, 2020.

[3] Sultan, D., Toktarova, A., Zhumadillayeva, A., Aldeshov, S., Mussiraliyeva, S., Beissenova, G., ... & Imanbayeva, A. (2023). Cyberbullying-related hate speech detection using shallow-to-deep learning. CMC-COMPUTERS MATERIALS & CONTINUA, 74(1), 2115-2131.

[4] S. Park, G. Li and J. Hong. “A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning,” Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 4, pp. 1405-1412, 2020.

[5] Murzamadieva, M., Ivashov, A., Omarov, B., Omarov, B., Kendzhayeva, B., & Abdrakhmanov, R. (2021, January). Development of a system for ensuring humidity in sport complexes. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 530-535). IEEE.

[6] Omarov, B., Altayeva, A., Turganbayeva, A., Abdulkarimova, G., Gusmanova, F., Sarbasova, A., ... & Omarov, N. (2019). Agent based modeling of smart grids in smart cities. In Electronic Governance and Open Society: Challenges in Eurasia: 5th International Conference, EGOSE 2018, St. Petersburg, Russia, November 14-16, 2018, Revised Selected Papers 5 (pp. 3-13). Springer International Publishing.

[7] F. Cauteruccio, L. Cinelli, E. Corradini, G. Terracina, D. Ursino et. al. “A framework for anomaly detection and classification in Multiple IoT scenarios,” Future Generation Computer Systems, vol. 114, pp. 322-335, 2021.

[8] A. Abou Daya, M. Salahuddin, N. Limam and R. Boutaba. “Botchase: Graph-based bot detection using machine learning,” IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 15-29, 2020.

[9] S. Gaddam, V. Phoha and K. Balagani, “K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, pp. 345-354, 2007.

[10] H. Alamri and V. Thayananthan. “Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks,” IEEE Access, vol. 8, no. 1, pp. 194269-194288, 2020.

[11] H. Jiang, Z. He, G. Ye and H. Zhang. “Network intrusion detection based on PSO-XGBoost model,” IEEE Access, vol. 8, no. 1, pp. 58392-58401, 2020.

[12] N. Shone, T. Ngoc, V. Phai and Q. Shi, “A deep learning approach to network intrusion detection”, IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, 2018.

[13] Y. Zhang, M. Li, Z. Ji, W. Fan, S. Yuan et. al. “Twin self-supervision based semi-supervised learning (TS-SSL): Retinal anomaly classification in SD-OCT images,” Neurocomputing, vol. 462, no. 1, pp. 491-505, 2021.

[14] W. Ullah, A. Ullah, I. Haq, K. Muhammad, M. Sajjad et. al. “CNN features with bi-directional LSTM for real-time anomaly detection in

surveillance networks". Multimedia Tools and Applications, vol. 80, no. 11, pp. 16979-16995, 2021.

[15] P. Devan and N. Khare, "An efficient xgboost–dnn-based classification model for network intrusion detection system," Neural Computing and Applications, vol. 32, no. 1, pp. 1-16, 2020.

[16] X. Du, W. Susilo, M. Guizani and Z. Tian, Z. "Introduction to the Special Section on Artificial Intelligence Security: Adversarial Attack and Defense," IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 905-907, 2021.

[17] Z. Geler, V. Kurbalija, M. Ivanović, and M. Radovanović, M. "Weighted kNN and constrained elastic distances for time-series classification," Expert Systems with Applications, vol. 162, no. 1, pp. 113829, 2020.

[18] R. Zhu, X. Ji, D. Yu, Z. Tan, L. Zhao et. al. "KNN-based approximate outlier detection algorithm over IoT streaming data," IEEE Access, vol. 8, pp. 42749-42759, 2020.

[19] Omarov, B., Orazbaev, E., Baimukhanbetov, B., Abusseitov, B., Khudiyarov, G., & Anarbayev, A. (2017). Test battery for comprehensive control in the training system of highly Skilled Wrestlers of Kazakhstan on National wrestling "Kazaksha Kuresi". Man In India, 97(11), 453-462.

[20] M. Lotfollahi, M. Siavoshani, R. Zade and M. Saberian, M. "Deep packet: A novel approach for encrypted traffic classification using deep learning," Soft Computing, vol. 24, no. 3, pp. 1999-2012, 2020.

[21] B. Mohammed, M. Hamdan, J. Bassi, H. Jamil, S. Khan et. al. "Edge Computing Intelligence Using Robust Feature Selection for Network Traffic Classification in Internet-of-Things," IEEE Access, vol. 8, no. 1, pp. 224059-224070, 2020.

[22] M. Xibilia, M. Latino, Z. Marinković, A. Atanasković and N. Donato. "Soft sensors based on deep neural networks for applications in security and safety," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 10, pp. 7869-7876, 2020.

[23] Omarov, B., Suliman, A., & Kushibar, K. (2016). Face recognition using artificial neural networks in parallel architecture. Journal of Theoretical and Applied Information Technology. Vol.91., No.2, pp. 238-248.

[24] Onalbek, Z. K., Omarov, B. S., Berkimbayev, K. M., Mukhamedzhanov, B. K., Usenbek, R. R., Kendzhaeva, B. B., & Mukhamedzhanova, M. Z. (2013). Forming of professional competence of future tyeacher-trainers as a factor of increasing the quality. Middle East Journal of Scientific Research, 15(9), 1272-1276.

[25] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas. "Modeling Intrusion Detection System using Hybrid Intelligent Systems," Journal of Network and Computer Applications, Vol. 30, no1, pp. 114–132, 2007.

[26] A. Davahli, M. Shamsi and Abaei. "A lightweight Anomaly detection model using SVM for WSNs in IoT through a hybrid feature selection algorithm based on GA and GWO," Journal of Computing and Security, vol. 7, no. 1, pp. 63-79, 2020.

[27] W. Ghanem and A. Jantan. "Training a neural network for cyberattack classification applications using hybridization of an artificial bee colony and monarch butterfly optimization," Neural Processing Letters, vol. 51, no. 1, pp. 905-946, 2020.

[28] C. Cortes and V. Vapnik. "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273-297, 1995.

[29] S. Dwivedi, M. Vardhan, S. Tripathi and A. Shukla. "Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection," Evolutionary Intelligence, vol. 13, no. 1, pp. 103-117, 2020.

[30] M. Gauthama Raman, N. Somu, S. Jagarapu, T. Manghnani, T. Selvam et. al. "An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm," Artificial Intelligence Review, vol. 53, no. 5, pp. 3255-3286, 2020.

[31] M. Eskandari, Z. Janjua, M. Vecchio and F. Antonelli. "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6882-6897, 2020.

[32] H. HaddadPajouh, A. Dehghantanha, R. Khayami and K. Choo. "A Deep recurrent neural network based approach for internet of things malware threat hunting," Future Generation Computer Systems, vol. 85, pp. 88–96, 2018.

[33] S. Rathore and J. Park. "A blockchain-based deep learning approach for cyber security in next generation industrial cyber-physical systems," IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5522-5532, 2020.

[34] A. Muna, N. Moustafa, E. Sitnikova. "Identification of malicious activities in industrial internet of things based on deep learning models," Journal of Information Security and Applications, vol. 41, pp. 1–11, 2018.

[35] S. Huda S. Miah, J. Yearwood, S. Alyahya, H. Al-Dossari et. al. "A malicious threat detection model for cloud assisted internet of things (CoT) based industrial control system (ICS) networks using deep belief network," Journal of Parallel and Distributed Computing, vol. 120, no. 1, pp. 23-31, 2018.

[36] O. Alkadi, N. Moustafa, B. Turnbull and K. Choo. "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9463-9472, 2020.

[37] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad. "Survey on SDN based network intrusion detection system using machine learning approaches," Peer-to-Peer Networking and Applications, vol. 12, no. 2, pp. 493-501, 2019.

[38] R. Abdulhammed, M. Faezipour, A. Abuzneid and A. AbuMallouh. "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," IEEE sensors letters, vol. 3, no. 1, pp. 1-4, 2018.

[39] UNSW-NB15 Network Dataset, 2021. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFANB15- Datasets/.

[40] N. Moustafa and J. Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," In 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6.

[41] N. Moustafa, B. Turnbull and K. Choo. "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," IEEE Internet of Things Journal, 2019, vol. 6, no. 3, pp. 4815–4830.

[42] Murugesan, G., Ahmed, T. I., Shabaz, M., Bhola, J., Omarov, B., Swaminathan, R., ... & Sumi, S. A. (2022). Assessment of mental workload by visual motor activity among control group and patient suffering from depressive disorder. Computational Intelligence and Neuroscience, 2022.

[43] Anand, M., Sahay, K. B., Ahmed, M. A., Sultan, D., Chandan, R. R., & Singh, B. (2023). Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. Theoretical Computer Science, 943, 203-218.

[44] R. Soleymani, E. Granger and G. Fumera, G. "F-measure curves: A tool to visualize classifier performance under imbalance,". Pattern Recognition, vol. 100, pp. 107146.

[45] M. Abu-Alhaija and N. Turab. "Automated Learning of ECG Streaming Data Through Machine Learning Internet of Things," Intelligent Automation & Soft Computing, vol. 32, no. 1, pp. 45–53, 2022.

[46] T. Heinis, C. Loy and Meboldt, M. "Improving Usage Metrics for Pay-per-Use Pricing with IoT Technology and Machine Learning: IoT technology and machine learning can identify and capture advanced metrics that make pay-per-use servitization models viable for a wider range of applications," Research-Technology Management, vol. 61, no. 5, pp. 32-40, 2018.

Scouting Firefly Algorithm and its Performance on Global Optimization Problems

Jolitte A. Villaruz[1], Bobby D. Gerardo[2], Ariel O. Gamao[3], Ruji P. Medina[4]
Graduate Programs, Technological Institute of the Philippines, Quezon City, Philippines[1, 2, 4]
College of Information and Computing Studies, Northern Iloilo State University, Iloilo City, Philippines[2]
Institute of Computing, Davao del Norte State College, Panabo City, Philippines[3]

***Abstract*—For effective optimization, metaheuristics should maintain the proper balance between exploration and exploitation. However, the standard firefly algorithm (FA) posted some limitations in its exploration process that can eventually lead to premature convergence, affecting its performance and adding uncertainty to the optimization results. To address these constraints, this study introduces an additional novel search mechanism for the standard FA inspired by the behavior of the scout bee in the artificial bee colony (ABC) algorithm, termed the "Scouting FA". Specifically, fireflies stuck in the local optima will take directed extra random walks to escape toward the region of the optimum solution, thus improving convergence accuracy. Empirical findings on the five standard benchmark functions have validated the effects of this modification and revealed that Scouting FA is superior to its original version.**

***Keywords*—Metaheuristics; firefly algorithm; modified firefly algorithm; global optimization; scout bee; exploitation and exploration**

I. Introduction

The firefly algorithm (FA) is a nature-inspired metaheuristics mimicking how fireflies behave. It was introduced by Yang [1] in 2009 to optimize multimodal problems. Over competing algorithms, FA proved advantageous owing to its simplicity, flexibility, ease of implementation, and few parameters to tune. As a result, FA quickly gains popularity in the scientific community. Furthermore, it has been empirically proven to handle NP-hard problems effectively [2]. Since its inception almost 15 years ago, FA and its modified variants have demonstrated significant success in various fields of application. For example, in multilevel image segmentation [3], as a way to reduce the number of dimensions [4], optimizing convolutional neural networks [5], solving course timetabling problems [6], and dealing with complex engineering tasks [7], [8], among other things. Knowing that FA has a universal application makes it a fascinating subject to pursue. In fact, this metaheuristic approach can be investigated further to provide solutions to real-world problems, such as IoT-based applications [9], time-series forecasting [10], [11], and machine vision-based tasks [12].

Metaheuristic optimization algorithms yielded approximations. Even though it does not guarantee the best solution, it gives the best result possible. The two main concepts of metaheuristics are exploration and exploitation [13]. Exploration searches space globally to locate the region with the optimal solution. On the other hand, exploitation seeks the optimal location of convergence by doing a local search within the area discovered by exploration. Because exploration and exploitation are inherently contrasting processes, it is critical to establish a robust exploration mechanism before starting the exploitation process. Likewise, it is also vital to maintain the proper balance between the two techniques during the search. That is why, if exploration is inefficient, the solution may converge too soon because it will become stuck in suboptimal domains before finding the optimal region [14], [15].

Previous studies reveal that FA is relatively robust at exploitation, although its exploratory ability can be improved [16]. Weaknesses in fireflies' exploration ability can eventually impact their convergence accuracy, adding uncertainty to the optimization results [17], [18]. Consequently, this limitation has narrowed the scope of the FA's applications. A recent approach to counter this drawback applied a damped vibration distribution factor to enhance the attractiveness and randomization formula [2]. Another notable solution utilized a quasi-reflection-based learning approach in the initiation stage of implementation to diversify the FA's population [19]. In addition, [20] enhanced the FA's exploration by adding genetic operators and using a dynamically modified step size. A hybrid method [21] used the group search approach established from the social network search (SNS) algorithm [22]. In [23], a novel method for updating the firefly's new location and the ABC [24] algorithm's scout bee search technique was also introduced to supplement the FA's search operation.

After numerous successful deployments of updated and hybridized FA variants, there are still opportunities for improvement. Initiatives to hybridize metaheuristic algorithms, such as those described in [25]–[27] have emerged. Similarly, the exploration mechanism of the FA can be combined with processes from various metaheuristics. Thus, the strengths of each algorithm could be merged to develop new algorithms that are more robust and accurate.

This work is yet another attempt to improve the FA's exploration process. An additional novel search mechanism inspired by the behavior of the scout bee in the ABC algorithm complements the original implementation. Specifically, fireflies stuck in the local optima will take directed extra random walks to escape toward the region of the optimum solution, thus improving convergence accuracy. This method is known as the scouting firefly algorithm (Scouting FA). This study's most significant contribution is providing a better version of the widely used FA that fixes the established shortcomings of the original version.

The remainder of the paper is organized as follows: Section II highlights the FA and ABC algorithms, and related works. Section III suggests a novel way to improve the FA. Section IV describes the experimental setup. The results and discussions are elucidated in Section V. Section VI gives a summary of the conclusions and future works.

II. Background and Related Works

A. Standard Firefly Algorithm (FA)

The fundamental concept of FA relates to the insects' bioluminescence, which is used to communicate with other fireflies. The following assumptions were made to represent the algorithm mathematically:

1) All fireflies are unisex and attract each other without regard to their sex;

2) Attractiveness is linked to brightness; thus, the firefly with lower intensity will approach the brighter one. Unless there is a brighter firefly, it will move randomly; and

3) The firefly's brightness corresponds to the optimization problem's objective function.

In the FA, convergence accuracy and speed largely depend on two key factors: the variation of light intensity and the formulation of attractiveness [28]. The attractiveness β is relative, depending on how a particular firefly perceives other fireflies. The relativity of β is computed relative to the distance r_{ij} between the firefly i and j. Accordingly, the farther apart the fireflies are, the less light they can see from one another, as governed by the inverse square law. Besides, the fact that light is absorbed by the atmosphere is also a significant factor to consider.

The Cartesian distance r_{ij} between any two fireflies i and j located at x_i and x_j is given by:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \quad (1)$$

where d is the dimension of the optimization problem.

Correspondingly, the light intensity $I(r)$ varies with the distance r monotonically and exponentially, as depicted in the formula:

$$I = I_0 e^{-\gamma r} \quad (2)$$

where I_0 is the initial light intensity and γ is the light absorption coefficient that controls the light intensity.

As a firefly's attractiveness is proportional to the light intensity perceived by adjacent fireflies, the attractiveness β of a firefly can be derived by:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (3)$$

where β_0 is the attractiveness at $r = 0$.

The formula to determine the movement of a less-bright firefly i that is attracted to brighter firefly j is given by:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha\varepsilon \quad (4)$$

where the $\beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i)$ calculates the attraction, while the α is randomization with the vector of random variables ε_I taken from a Gaussian distribution or uniform distribution in range [0, 1].

B. Artificial Bee Colony (ABC) Algorithm

The ABC algorithm is another well-known nature-inspired metaheuristic [24]. It was formally established by Karaboga in 2005 and modeled on how honeybee swarms search for food. Artificial bee colonies are classified as employed, onlookers, and scouts. Employed bees identify food sources within the search space and relay information about the food sources to onlookers via dance moves. The onlookers choose one of the food sources based on its attributes, while the scouts search for new food sources randomly. Initially, all the bees in the colony were scouts.

When the scout finds a new food source using Eq. 5 and starts to consume it, that scout is turned into an employed bee. The abandonment of the food sources will commence once the "limit" given by Eq. (6) is met. Conversely, the employed bee will become a scout once the "trial" exceeds the "limit." The "trial" counter will increase if the scout cannot discover a new food source; otherwise, it will be reset to zero.

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \quad (5)$$

Assume that the abandoned source is x_i and $j \in \{1, 2,...,D\}$, then the scout discovers a new food source to be replaced with x_i.

The limit parameter l is computed using:

$$l = (CS * D)/2 \quad (6)$$

where *CS* is the colony size and *D* is the dimension of the problem.

C. Related Works

According to the no-free-lunch theorem [29], no single approach can solve all optimization problems. Thus, the original implementations of metaheuristics are modified to perform better. Two outstanding FA versions incorporating the ABC algorithm are discussed below.

A hybrid of firefly and multi-strategy ABC for optimizing single-objective problems is presented in [27]. The FA performs the global search, while the novel multi-strategy ABC does the local search. Nevertheless, this approach displayed computational complexity because two independent search techniques coexisted throughout the search process. Furthermore, switching the search from FA to the multi-strategy ABC employs a diversity measure that raises computational costs.

The work of [23] is another impactful study that enhanced the exploration of FA using the scout bee search mechanism of the ABC algorithm. In their approach, non-improving fireflies will be replaced by new fireflies at random locations within the specified lower (lb_j) and upper bounds (ub_j) based on the formula:

$$x_{i,j} = lb_j + rand * (ub_j - lb_j) \tag{7}$$

where $x_{i,j}$ signifies the *i*th firefly and the corresponding *j*th element; and *rand* is a uniform random number.

The above method provides an entirely random position for fireflies that exhibits no improvements above a specific threshold limit. This can cause fireflies already nearing convergence to spread farther away, necessitating additional exploitation and exploration. This implementation method may impact both the convergence accuracy and the convergence rate.

III. Scouting Firefly Algorithm

The Scouting FA aims to improve the standard FA's exploration ability by allowing the fireflies stuck in the local optimum over a specific threshold limit ("limit") to take directed extra random walks to scout unfamiliar regions in the search space further. The formula for "limit" is identical to that stated in Eq. 6. Afterward, a greedy selection will be utilized. If the new solution has a higher fitness value than the previous one, its position will be updated by Eq. 8; otherwise, no movement will occur.

The directed extra random walk formula is defined as:

$$x_{i+1} = x_i + (\alpha_i - 0.5) * (ub - lb) \tag{8}$$

where x_{i+1} refers to the new position of the firefly after taking a random walk, x_i is the current position, α_i denotes a random number drawn from uniform distribution $U(0, 1)$, while *ub* stands for upper bound and *lb* for the lower bound. $(\alpha_i - 0.5)$ will ensure that a directed extra random walk is provided, such that, if the generated α_i is < 0.50, the firefly will move backward, and if it is > 0.50, a forward movement will be made.

The novelty of Scouting FA over [23] is that instead of providing an entirely random position for fireflies that exhibits no improvements, directed extra random walks with greedy selection is applied. This is to avoid spreading away fireflies that were already nearing convergence. Furthermore, unlike [27], as an alternative for having two independent search techniques that coexisted throughout the search process and the required diversity measure for switching between searches, this study complemented the standard FA as the extra random walks will only be executed when it improves the fitness value even further. The pseudo code of the Scouting FA is described in Algorithm 1.

Algorithm 1: Pseudocode of the Scouting Firefly Algorithm

```
Objective function $f(x)$, $x = (x_1, ..., x_d)^T$.
Generate an initial population of $n$ fireflies $x_i$ $(i = 1, 2, ..., n)$.
Light intensity $I_i$ at $x_i$ is determined by $f(x_i)$.
Define FA parameter: $\alpha$, $\beta$, $\gamma$.
Set value for the threshold limit value, limit, using Eq. 6
while ($t$ < maxGeneration),
    for $i$ = 1:$n$ (all $n$ fireflies)
        for $j$ = 1:$n$ (all $n$ fireflies) (inner loop)
            if ($I_i < I_j$)
                Move firefly $i$ towards $j$ using Eq. 4
                Reset $trial_i = 0$
            else // solution has no improvement
                $trial_i$++
            end if
            Vary attractiveness with distance $r$ via exp[$-\gamma r2$].
            Evaluate new solutions and update light intensity.
        end for $j$
    end for $i$
    for $i$ = 1:$n$ (all $n$ fireflies)
        if ($trial_i$ >= limit) // no improvement for limit times
            Scout for a new position, P, via random walk using Eq. 8
            if ($I_P > I_i$) // greedy selection
                Move firefly $i$ to position P
            end if
        end if
    end for
    Rank the fireflies and find the current global best $g*$.
end while
Postprocess results and visualization.
```

IV. Experimental Set-up

Five well-known benchmark functions were used to validate the performance of the Scouting FA compared to the original implementation. Table I lists these functions, their formula, variable limits, and global optimum.

TABLE I. Benchmark Functions Used in the Experiments

Function	Formula	Limits	Optimum
Sphere	$f_1(x) = \sum_{i=1}^{d} x_i^2$	[-5.12, 5.12]	f(x)=0 at x=(0,0)
Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	[-10, 10]	f(x)=0 at x=(1,3)
Easom	$f_3(x) = -cos(x_i)\ cos(x_i)\ exp(-\ (x_1 - \pi)^2 - (x_2 - \pi)^2)$	[-100, 100]	f(x)=-1 at x=(π,π)
Rosenbrock	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_1^2)^2 + (x_i - 1)^2]$	[-5, 10]	f(x)=0 at x=(1,1)
Ackley	$f_2(x) = -20 \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} cos(cx_i)\right) + a + \exp(1)$	[-5, 5]	f(x)=0 at x=(0,0)

The control parameter values used in the simulations are given in Table II.

TABLE II. CONTROL PARAMETERS OF STANDARD FA AND SCOUTING FA AS IMPLEMENTED IN THE STUDY

Parameter	Value
Maximum generation (*maxGeneration*)	300
Population size (*n*)	30
Randomization parameter (α)	1.0
Attractiveness (β) at *r = 0*	1.0
Light absorption coefficient (γ)	0.97

All experiments were carried out in Python 3.10.5 on an Intel(R) Core (TM) i7-11370H processor running at 3.30 GHz and with 40 GB of random-access memory.

V. RESULTS AND DISCUSSION

Due to the stochastic nature of the metaheuristic algorithms, each iteration is seeded with a random number to ensure that each solution is unique. Experimental results are shown in Table III. The best, worst, and mean optimal fitness value were noted and compared. The results of the mean optimal fitness value were generated in 100 independent runs to eliminate the effect of the stochastic simulation discrepancy [28]. If an algorithm gets the best results for the performance metric, the results are shown in bold and in a slightly bigger font size.

TABLE III. COMPARATIVE ANALYSIS WITH STANDARD FA AND SCOUTING FA IMPLEMENTATIONS FOR THE FIVE BENCHMARKS

Functions	Optimization Method	Best	Worst	Mean Optimal Fitness Value
Sphere	Standard FA	1.84E-05	1.83E+00	6.44E-02
	Scouting FA	**2.27E-06**	**1.25E-02**	**1.96E-03**
Booth	Standard FA	2.65E-04	4.80E+00	4.79E-01
	Scouting FA	**1.65E-06**	**1.58E-01**	**6.23E-03**
Easom	Standard FA	-1.00E+00	-8.02E-05	-7.11E-01
	Scouting FA	**-1.00E+00**	**-9.70E-01**	**-9.98E-01**
Rosenbrock	Standard FA	8.66E-03	5.30E+01	2.48E+00
	Scouting FA	**4.51E-04**	**1.12E+00**	**2.63E-01**
Ackley	Standard FA	1.18E-02	4.42E+00	2.11E+00
	Scouting FA	**2.67E-03**	**4.04E-01**	**2.63E-01**

As shown, the Scouting FA outperformed the standard FA in all test functions. Even if the worst values are considered, the Scouting FA is closer to the global optimum than the standard FA.

Fig. 1 to 5 compare convergence plots in the five benchmark functions between the standard FA and the Scouting FA.

Fig. 1 demonstrates that when optimizing the Sphere function, the solution in the standard FA is trapped in the local optimum close to 0.20 of the fitness value in the early iterations. On the other hand, the Scouting FA performs better because it avoided premature convergence and gradually moved closer to the global optimum, *f(x)=0 at x=(0,0)*, in just more than 50 iterations.

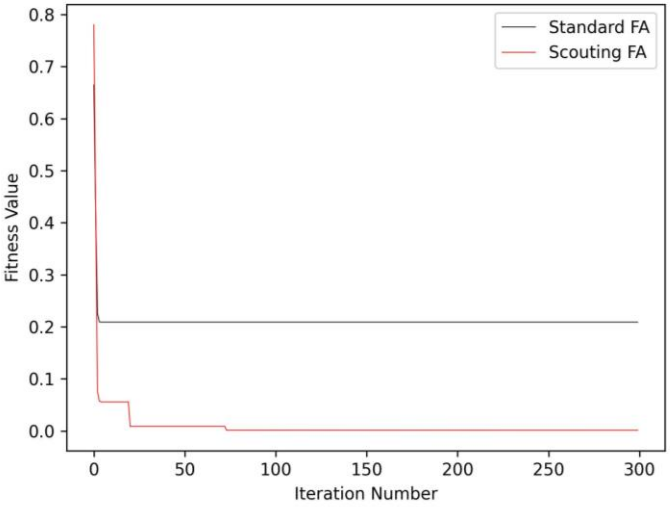


Fig. 1. Convergence plot for the Sphere function.

As shown in Fig. 2, early runs of the standard FA reveal an early convergence nearing 0.20 of the fitness value. When tested using the Booth function, the application of the Scouting FA is more effective. The graph demonstrates the solution progresses towards convergence at the global optimum, *f(x)=0 at x=(1,3)*, in more than 150 iterations.

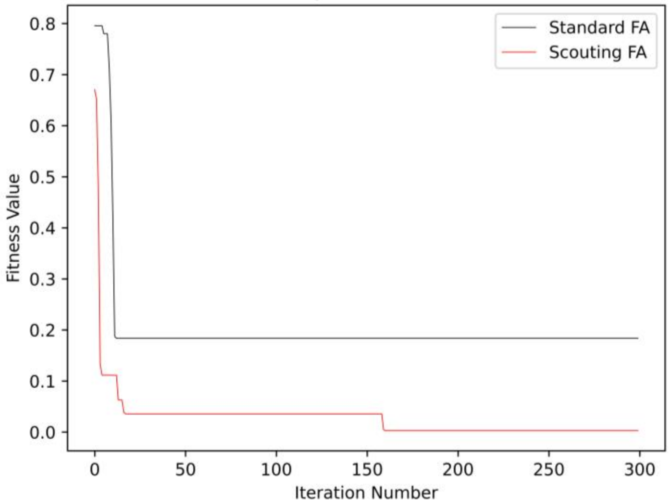


Fig. 2. Convergence plot for the booth function.

As depicted in Fig. 3, in terms of the Easom function, the premature convergence occurs very quickly at nearly -0.80 in the standard FA. Contrary to that, Scouting FA yielded a better result, as escaping from the local optimum is evident. In this problem, the global minimum, *f(x)*=-1 at *x*=(π,π), is nearly achieved at below 50 iterations.

When the Scouting FA optimized the difficult Rosenbrock function, as shown in Fig. 4, there was a significant improvement in achieving a near-optimal result. As illustrated, the Scouting FA's extra random walks slowly improve the solution until it converges close to the global optimum at *f(x)=0 at x=(1,1)* after the 200th iteration.

For a complex optimization problem like the Ackley function, the graph depicted in Fig. 5 revealed that the Scouting FA is considerably better than its original implementation. While standard FA prematurely converges above the 1.5 fitness value, the extra random walks the Scouting FA provides have progressively improved the fitness value until it achieves close to *f(x)=0 at x=(0,0)* before it reaches the 300th iteration.

Fig. 6 shows how 30 fireflies in the five test functions converged after 300 iterations when the standard FA was used. In Fig. 7, the Scouting FA was utilized. As depicted, indeed, with Scouting FA, fireflies can escape being trapped in the sub-optimal solution by taking directed extra random walks.

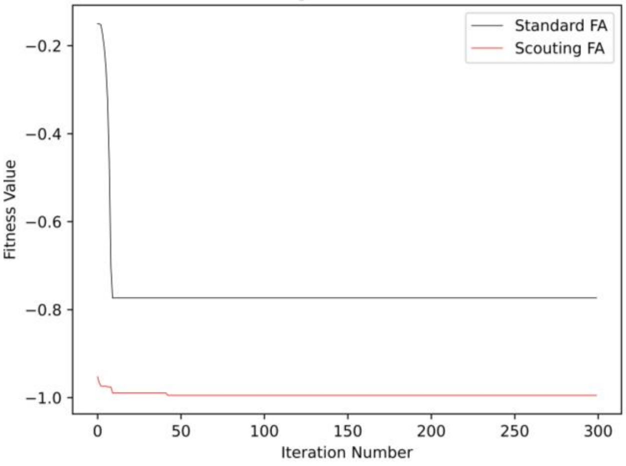


Fig. 3. Convergence plot for the Easom function.

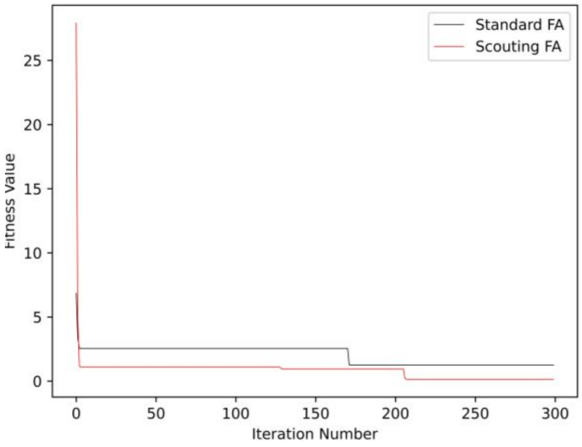


Fig. 4. Convergence plot for the rosenbrock function.

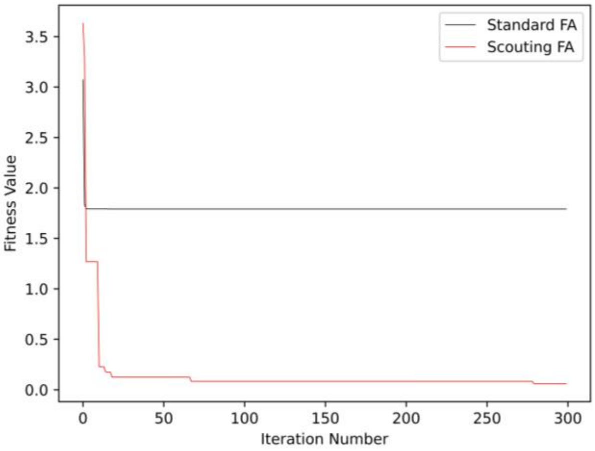
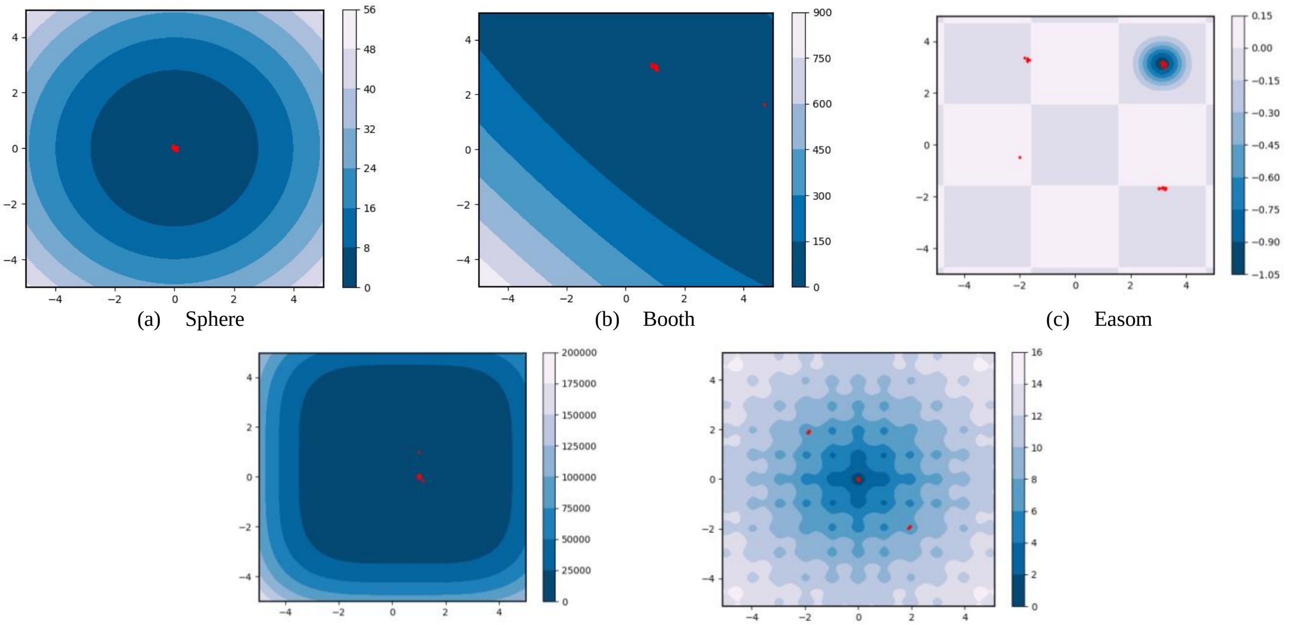


Fig. 5. Convergence plot for the ackley function.



(a) Sphere

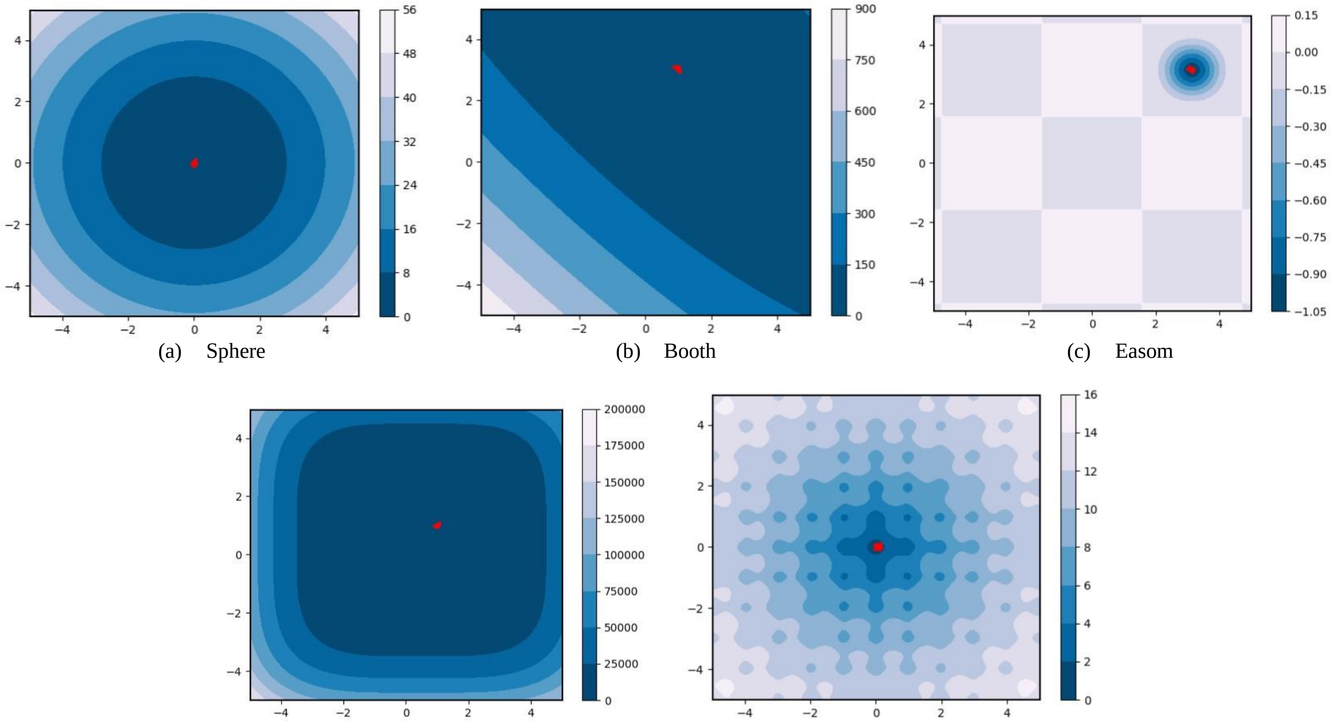
(b) Booth

(c) Easom

(d) Rosenbrock

(e) Ackley

Fig. 6. The convergence of 30 fireflies in the five test functions after 300 iterations by implementing the standard FA.



(a) Sphere (b) Booth (c) Easom

(d) Rosenbrock (e) Ackley

Fig. 7. The convergence of 30 fireflies in the five test functions after 300 iterations by implementing the Scouting FA.

VI. Conclusions and Future Works

This study improved the exploration of the standard FA by adding the behavior of the scout bee from the ABC algorithm. When the novel search method was used to solve global optimization problems, it made the search more precise, significantly improving convergence accuracy. This result implies that the Scouting FA is more powerful than its original implementation.

In the future, researchers will investigate how well the Scouting FA works when used to tune the hyperparameters of machine learning-based methods for solving real-world optimization problems.

Acknowledgment

The authors acknowledge Aklan State University for the scholarship grant that made this work possible. The recommendations and expertise shared by Mr. Kirk Hilario are also greatly appreciated.

References

[1] X.-S. Yang, “Firefly algorithms for multimodal optimization,” in *International symposium on stochastic algorithms*, 2009, pp. 169–178.

[2] J. Liu, J. Shi, F. Hao, M. Dai, and X. Zhang, “A novel enhanced exploration firefly algorithm for global continuous optimization problems,” *Eng Comput*, vol. 38, pp. 4479–4500, Dec. 2022, doi: 10.1007/s00366-021-01477-6.

[3] A. Sharma, R. Chaturvedi, and A. Bhargava, “A novel opposition based improved firefly algorithm for multilevel image segmentation,” *Multimed Tools Appl*, vol. 81, no. 11, pp. 15521–15544, 2022.

[4] A. O. Gamao and B. D. Gerardo, “Prediction-based model for student dropouts using modified mutated firefly algorithm,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3461–3469, Nov. 2019, doi: 10.30534/ijatcse/2019/122862019.

[5] J. Wang, Y. Liu, and H. Feng, “IFACNN: efficient DDoS attack detection based on improved firefly algorithm to optimize convolutional neural networks,” *Mathematical Biosciences and Engineering*, vol. 19, no. 2, pp. 1280–1303, 2022.

[6] T. Thepphakorn and P. Pongcharoen, “Modified and hybridised bi-objective firefly algorithms for university course scheduling,” *Soft comput*, pp. 1–38, 2023.

[7] S. Bazi, R. Benzid, Y. Bazi, and M. M. al Rahhal, “A fast firefly algorithm for function optimization: Application to the control of bldc motor,” *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165267.

[8] H. Peng *et al.*, “Multi-strategy firefly algorithm with selective ensemble for complex engineering optimization problems,” *Appl Soft Comput*, vol. 120, p. 108634, 2022.

[9] D. B. Balmadrid, J. N. Mallorca, B. D. Gerardo, and R. P. Medina, “IoT-based LED Lighting System with Variable Pulsing Frequency and Dark Periods for Sunflower Microgreens,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 8, pp. 136–143, Aug. 2022, doi: 10.46338/ijetae0822_17.

[10] M. J. D. Viñas, B. D. Gerardo, and R. P. Medina, “Forecasting PM2. 5 and PM10 Air Quality Index using Artificial Neural Network,” *Journal of Positive School Psychology*, pp. 6863–6871, 2022.

[11] S. G. Aribe, B. D. Gerardo, and R. P. Medina, “Time Series Forecasting of HIV/AIDS in the Philippines Using Deep Learning: Does COVID-19 Epidemic Matter?,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 9, pp. 144–157, Sep. 2022, doi: 10.46338/ijetae0922_15.

[12] D. M. Barrios, R. G. Lumauag, and J. A. Villaruz, “Machine vision-based dried danggit sorter,” in *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, 2019. doi: 10.1109/CCOMS.2019.8821634.

[13] M. Sababha, M. Zohdy, and M. Kafafy, “The enhanced firefly algorithm based on modified exploitation and exploration mechanism,” *Electronics (Switzerland)*, vol. 7, no. 8, Aug. 2018, doi: 10.3390/electronics7080132.

[14] M. J. Goldanloo and F. S. Gharehchopogh, “A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems,” *J Supercomput*, vol. 78, no. 3, pp. 3998–4031, 2022.

[15] [15] K. Rezaei and H. Rezaei, “An improved firefly algorithm for numerical optimization problems and it’s application in constrained optimization,” *Eng Comput*, vol. 38, no. 4, pp. 3793–3813, 2022.

[16] X.-S. Yang, *Nature-inspired Algorithms and Applied Optimization*, vol. 744. 2018. [Online]. Available: http://link.springer.com/10.1007/978-3-319-67669-2

[17] M.-R. Chen, L.-Q. Yang, G.-Q. Zeng, K.-D. Lu, and Y.-Y. Huang, “IFA-EO: An improved firefly algorithm hybridized with extremal optimization for continuous unconstrained optimization problems,” *Soft comput*, pp. 1–22, 2022.

[18] A. Abbaszadeh Shahri, M. Khorsand Zak, and H. Abbaszadeh Shahri, “A modified firefly algorithm applying on multi-objective radial-based function for blasting,” *Neural Comput Appl*, pp. 1–17, 2022.

[19] T. Bezdan, D. Cvetnic, L. Gajic, M. Zivkovic, I. Strumberger, and N. Bacanin, “Feature selection by firefly algorithm with improved initialization strategy,” in *7th conference on the engineering of computer based systems*, 2021, pp. 1–8.

[20] N. Bacanin, M. Zivkovic, T. Bezdan, K. Venkatachalam, and M. Abouhawwash, “Modified firefly algorithm for workflow scheduling in cloud-edge environment,” *Neural Comput Appl*, vol. 34, no. 11, pp. 9043–9068, 2022.

[21] D. Jovanovic, M. Antonijevic, M. Stankovic, M. Zivkovic, M. Tanaskovic, and N. Bacanin, “Tuning Machine Learning Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection,” *Mathematics*, vol. 10, no. 13, Jul. 2022, doi: 10.3390/math10132272.

[22] S. Talatahari, H. Bayzidi, and M. Saraee, “Social network search for global optimization,” *IEEE Access*, vol. 9, pp. 92815–92863, 2021.

[23] N. Bacanin, T. Bezdan, K. Venkatachalam, and F. Al-Turjman, “Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade,” in *Journal of Real-Time Image Processing*, Aug. 2021, vol. 18, no. 4, pp. 1085–1098. doi: 10.1007/s11554-021-01106-x.

[24] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical report-tr06, Erciyes university, engineering faculty, computer …, 2005.

[25] L. D. Austero, A. M. Sison, J. B. Matias, and R. P. Medina, “Solving course timetabling problem using Whale Optimization Algorithm,” in *2022 8th International Conference on Information Technology Trends (ITT)*, 2022, pp. 160–166.

[26] J. C. M. Bustillo, R. P. Medina, A. M. Sison, and M. Y. Orong, “Predictive Hybridization Model integrating Modified Genetic Algorithm (MGA) and C4.5,” in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, 2022, pp. 1500–1507. doi: 10.1109/ICECA55336.2022.10009532.

[27] I. Brajević, P. S. Stanimirović, S. Li, and X. Cao, “A hybrid firefly and multi-strategy artificial bee colony algorithm,” *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 810–821, 2020, doi: 10.2991/ijcis.d.200612.001.

[28] X.-S. Yang, “Firefly algorithm, stochastic test functions and design optimisation,” *arXiv preprint arXiv:1003.1409*, 2010.

[29] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

A Robust Steganographic Algorithm based on Linear Fractional Transformation and Chaotic Maps

Steganographic Algorithm based on S-Boxes

Muhammad Ramzan[1], Muhammad Fahad Khan[2]
Department of Computer Science-College of Computing and Informatics,
Saudi Electronic University, Riyadh 11673, Saudi Arabia[1]
Department of Software Engineering, Foundation University Islamabad, 44000, Pakistan[2]

***Abstract*—The fundamental objectives of a steganographic technique are to achieve both robustness and high-capacity for the hidden information. This paper proposes a steganographic algorithm that satisfies both of these objectives, based on enhanced chaotic maps. The algorithm consists of two phases. In the first phase, a cryptographic substitution box is constructed using a novel fusion technique based on logistic and sine maps. This technique overcomes existing vulnerabilities of chaotic maps, such as frail chaos, finite precision effects, dynamical degradation, and limited control parameters. In the second phase, a frequency-domain-based embedding scheme is used to transform the secret information into ciphertext by employing the substitution boxes. The statistical strength of the algorithm is assessed through several tests, including measures of homogeneity, correlation, mean squared error, information entropy, contrast, peak signal-to-noise ratio, energy, as well as evaluations of the algorithm's performance under JPEG compression and image degradation. The results of these tests demonstrate the algorithm's robustness against various attacks and provide evidence of its high-capacity for securely embedding secret information with good visual quality.**

Keywords—Steganography; information security; chaotic map vulnerabilities; enhanced chaotic maps; S-box Design

I. INTRODUCTION

History has demonstrated that secret communication has always been an essential requirement in human society. As time progressed, more advanced techniques have been introduced. In the last two decades, the rapid development of digital communication systems has significantly increased the demand for secure data exchange through digital multimedia. Innovative strategies to protect confidential information from intruders have become the focus of recent research. In this regard, modern techniques of cryptography, watermarking, and steganography have gained unusual importance in the last few years [1-5]. Cryptography involves converting useful information into dummy data to protect it from unintended recipients, while watermarking is associated with protecting the data's copyright. Steganography, on the other hand, conceals confidential information into other information [6-8]. With the advent of modern computer technology, remarkable skills for surreptitious communication have been developed. Steganography involves embedding secret information in either the spatial or transform (frequency) domain. In spatial domain embedding, the LSB-substitution technique is most commonly used. However, in the transform domain, invertible transforms such as (DCT) and (DWT) are typically applied to transform the image into its frequency representation [9-11]. While both domains have advantages and disadvantages, frequency domain embedding is robust, while spatial domain offers increased capacity for hiding data. This motivates researchers to deploy a combination of both domains. The transforms DCT and DWT are frequently used in image compression applications due to their favorable features. DCT is the most widely used, requiring fewer computational resources; however, DWT is considered more efficient in quality. Many multimedia applications and algorithms in recent literature are based on the joint applications of DCT and DWT [12-15]. In our proposed method, we apply a hybrid of these transforms for improved and robust outcomes.

In the last decade, chaos has been widely used to enhance the security level of confidential communication. Chaotic systems possess prime features such as irregularity, butterfly effect and unpredictability which making them well-suited for multimedia security applications. Consequently, the study and analysis of chaos-based steganographic techniques have gained popularity in recent years. However, it has been observed that some chaos-based methods are vulnerable to statistical analysis because of the limited chaotic range of the used maps [16-20]. To overcome this issue, authors of [19] proposed a nonlinear combination of one-dimensional chaotic maps that enhances the chaotic range of the resulting system. Such systems are applied in image encryption applications, but to the best of our knowledge, they have not been applied in steganographic methods yet.

The fundamental objectives of a steganographic technique are to achieve both robustness and high-capacity for the hidden information. This paper proposes a steganographic algorithm that satisfies both of these objectives, based on enhanced chaotic maps. The algorithm consists of two phases. In the first phase, a cryptographic substitution box is constructed using a novel fusion technique based on logistic and sine maps. This technique overcomes existing vulnerabilities of chaotic maps, such as frail chaos, finite precision effects, dynamical degradation, and limited control parameters. In the second phase, a frequency-domain-based embedding scheme is used to transform the secret information into ciphertext by employing the substitution boxes. In addition, we exploit the combination of the spatial and transform domains to achieve a significantly high capacity level for embedding secret data. Our technique uses a hybrid of DCT and DWT, and we observe that the combined effect of DCT and DWT increases robustness against several image processing attacks. The strength of the proposed method is evaluated through the most frequently used analysis techniques, and we prove that our technique produces coherent results. The use of enhanced chaotic systems in steganography has been demonstrated to be effective in various studies, highlighting the potential for further development in this field. The structure of the paper is organized as follows: Section II presents the chaotic map fusion technique. Section III describes the design of the S-box. Section IV presents a novel steganographic technique and Section V provides the security analysis and simulation results. Finally, Section VI presents the conclusions.

II. Fusion of 1-Dimension Chaotic Maps

The study of security protocols has demonstrated the extensive applications of one-dimensional chaotic maps due to their simple structure and computational convenience. In our research, we use the logistic map and the sine map to develop a stronger chaotic system for our problem. In the upcoming sections, we explored the essential features of these maps and their implementation in our research.

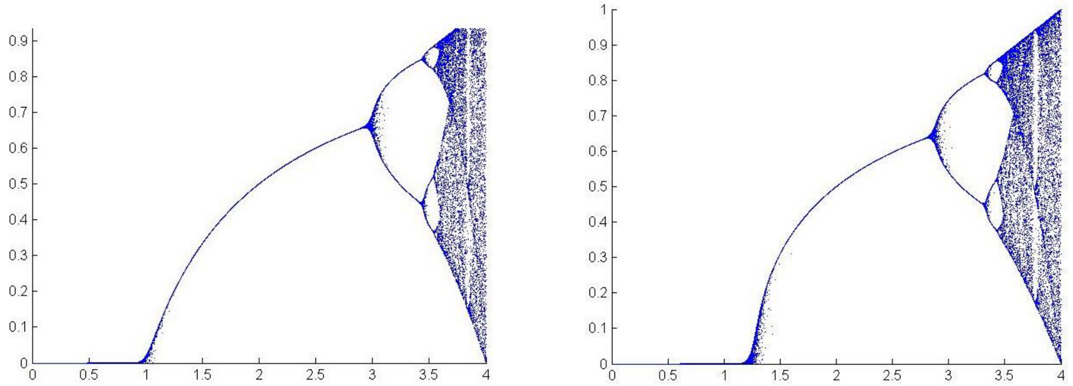
A. The Logistic Map

It is a mathematical function that has been used in various applications in information security due to its ability to generate chaotic and pseudorandom sequences. The logistic map is defined by a quadratic recurrence, which can be written as:

$$\mathcal{C}_{\mathcal{L}}(\vartheta_{\mathcal{L}}, x_i\ ) = \vartheta_{\mathcal{L}} x_i(-x_i); \qquad (1)$$

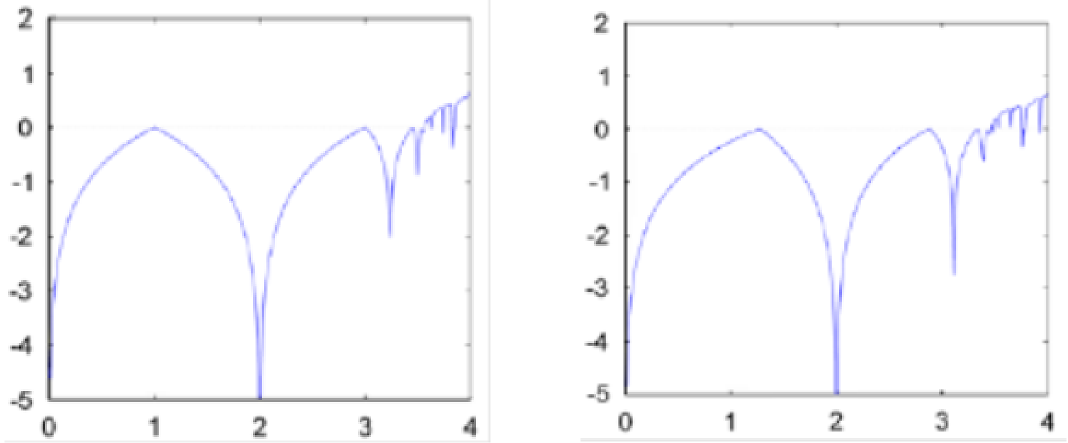
Where $\vartheta_{\mathcal{L}} \in (0, 4]$ is called the control parameter. It is also known *as a catalyst* for chaos as the behavior of the map varies when the value of parameter $\vartheta_{\mathcal{L}}$ changes. It is clear from the bifurcation diagram (as shown in Fig. 1(a)) that $\mathcal{C}_{\mathcal{L}}$ produces the chaotic effect only when $\vartheta_{\mathcal{L}} \in [3.57, 4]$. Although the Logistic map is widely used, it has been noted that its chaotic range is limited, which can be a drawback in some applications.

Another important characteristic of a dynamical system is Lyapunov Exponent, which is a quantitative measure of chaos. A system is chaotic if the value of the Lyapunov exponent is strictly positive, as then a minor disturbance in the initial conditions may cause exponential divergence. The larger the Lyapunov exponent, the better is chaotic performance. Fig. 2(a) shows the variations of the Lyapunov exponent of the logistic map.



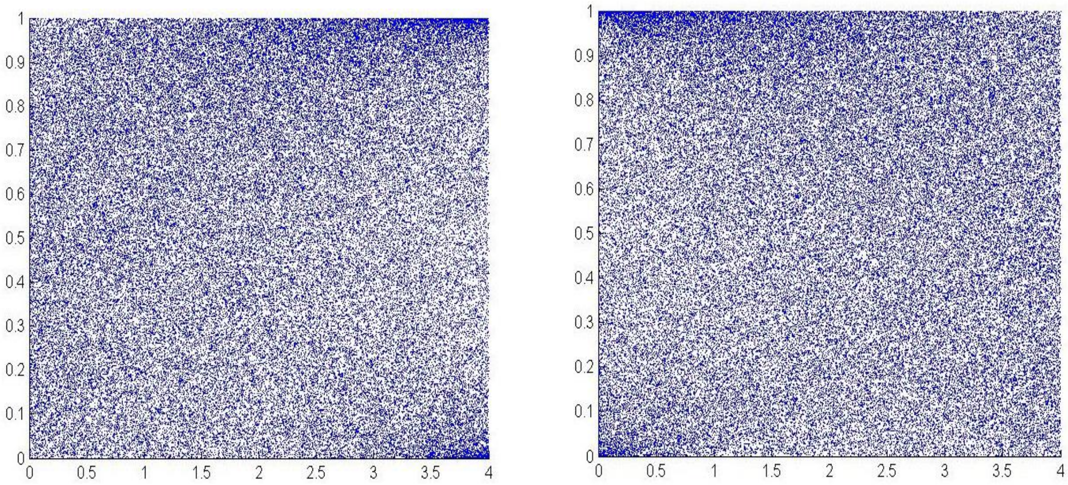
(a) (b)

Fig. 1. (a) Bifurcation diagrams of logistic map, (b) Bifurcation diagrams of sine map.



(a) (b)

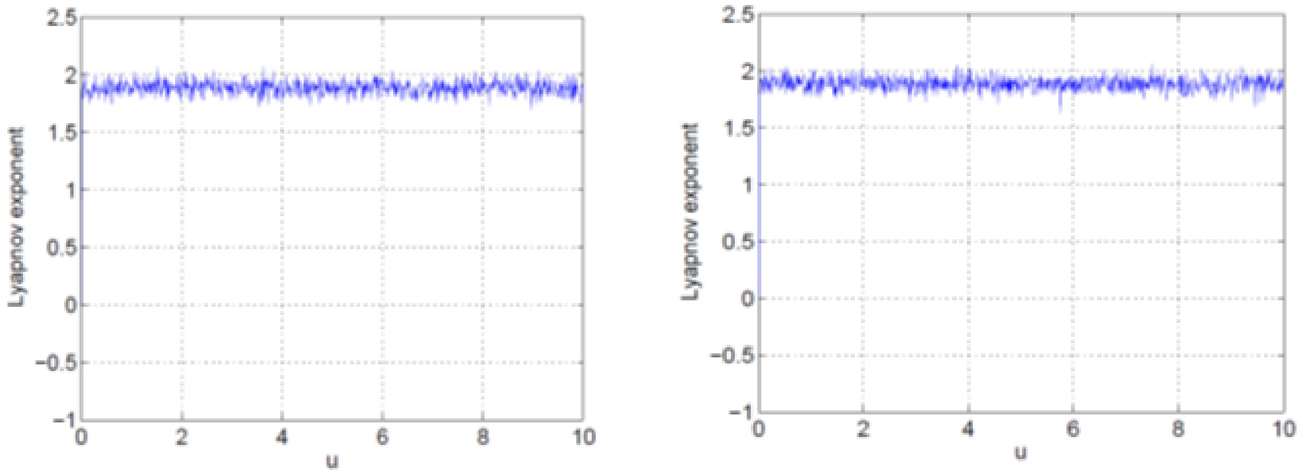
Fig. 2. (a) Lyapunov exponents of logistic maps, (b) Lyapunov exponents of sine maps.



(a) LLS

(b) SSS

Fig. 3. Bifurcation diagrams of LLS and SSS.



(a) LLS

(b) SSS

Fig. 4. The Lyapunov exponent of LLS and SSS.

B. The Sine Map

It is a type of chaotic system that has received considerable attention in the area of nonlinear dynamics. It is a straightforward mathematical model that is capable of displaying intricate and unpredictable patterns. Sine map equation is represented as:

$$\mathcal{C}_{\mathcal{S}}(\vartheta_{\mathcal{S}}, x_i) = \frac{\vartheta_{\mathcal{S}} Sin(\pi x_i)}{4}; \qquad (2)$$

Where $\vartheta_{\mathcal{S}} \in (0,4)$. Like the logistic map, the sine map also exhibits chaotic behavior. However, it has been observed that the chaotic range of the sine map is also limited, as shown in the bifurcation diagram presented in Fig. 1(b). The variations of the Lyapunov exponent of the sine map are shown in Fig. 2(b) Study regarding the combinations of one-dimensional chaotic maps shows that by introducing suitable combinations of such maps, the chaotic range can be enhanced [19]. In the next section, we discuss such combinations in detail.

C. Combinations of Chaotic Maps

Our approach involves creating new chaotic systems that have chaotic properties across their entire domain. We accomplish this by constructing nonlinear combinations of the underlying seed maps, which includes the LLS and SSS. The mathematical expression used for the nonlinear combination LLS is given below:

$$x_{i+1} = \mathcal{C}_{\mathcal{L}}(\vartheta_{\mathcal{L}}, x_i) \times \psi(n) - \mathcal{C}_{\mathcal{L}}(\vartheta_{\mathcal{L}}, x_i) \times \psi(n) - \lfloor \mathcal{C}_{\mathcal{L}}(\vartheta_{\mathcal{L}}, x_i) \times \psi(n) \rfloor; \qquad (3)$$

Where $\psi(n) = 2^n$; $8 \le n \le 14$, is called an adjustment function. The larger the value of n, the better is the chaotic performance. For both LLS and SSS we choose n = 14. Eq. (3) can be rewritten as:

$$x_{i+1} = \vartheta_{\mathcal{L}} x_i(1 - x_i) \times 2^{14} - \lfloor \vartheta_{\mathcal{L}} x_i(1 - x_i) \times 2^{14} \rfloor \quad (4)$$

The figures depicting the bifurcation diagram and the Lyapunov exponent of the LLS chaotic system can be observed in Fig. 3(a) and Fig. 4(a), respectively. A comparison of the bifurcation diagram of LLS with that of the logistic map in Fig. 2(a) reveals that the former exhibits a significantly wider chaotic range than the latter. Additionally, the improved uniform distribution of the density function in LLS compared to its seed maps ensures better chaotic behavior of the newly generated system and its efficient application in information security. The other two chaotic systems discussed in the following subsections exhibit similar properties. The expression for the Sine-Sine system can be obtained by using Eq. (1).

$$\mathcal{C}_{\mathcal{S}}(\vartheta_{\mathcal{S}}, x_i) == \frac{\vartheta_{\mathcal{S}} Sin(\pi x_i)}{4} \times 2^{14} - \left\lfloor \frac{\vartheta_{\mathcal{S}} Sin(\pi x_i)}{4} \times 2^{14} \right\rfloor \quad (5)$$

The bifurcation diagram and the variations of Lyapunov exponent are exhibited in Fig. 3(b) and Fig. 4(b). Just like the previous case, one may observe the improved performance of the new chaotic system.

III. Construction of Cryptographic S-Box

Block ciphers are cryptographic algorithms that convert plaintext into ciphertext by breaking it up into fixed-length blocks and applying a series of substitution and permutation operations. They are widely used to secure data in various applications, such as electronic payments, online transactions, and communication systems. S-boxes, or substitution boxes, are a key component of block ciphers, as they determine the nonlinear substitution of input bits with output bits in the encryption process. The quality of the S-box plays a vital role in the overall security of the cryptosystem, as any weakness or vulnerability in the S-box can be exploited by attackers to break the encryption. To achieve resistance against differential and linear cryptanalysis in an encryption scheme, the selection of a suitable S-box is crucial. Differential and linear cryptanalysis are two common techniques used by attackers to break ciphers, and a well-designed S-box can help to mitigate these attacks and provide stronger security. To ensure the highest level of security, cryptographers must carefully design and test S-boxes to ensure they are resistant to attacks and produce high-quality encryption output. The selection and design of S-boxes is a complex and ongoing research area, as the security requirements and threats to cryptosystems are constantly evolving. Despite the challenges, the use of strong and secure S-boxes is crucial for the development of robust and reliable cryptosystems.

Where $GL(n;\ \mathbb{F})$ are a group of invertible matrices and PGL of degree n over a field $\mathbb{F}$ is as $GL(n;\ \mathbb{F})$ by its center. We also form the 8×8 S-box through $GF(2^8)$ on $PGL\big(2, GF(2^8)\big)$:

$$f\ :\ PGL\big(2, GF(2^8)\big) \times GF(2^8) \rightarrow\ GF(2^8)$$

Defined as:

$$\tau(z)\ = \frac{\alpha z + \beta}{\gamma z + \delta} \quad (6)$$

Here τ is LFT of $\in GF(2^8)$, which satisfying the non-degeneracy condition $\alpha\delta - \beta\gamma \neq 0$. Where $\alpha = 21$, $\beta = 8$ and $\gamma = 3$ and $\delta = 17$.

TABLE I. CONSTRUCTED S-BOX

103	157	39	10	87	238	191	15	141	229	166	70	243	119	61	24
72	50	216	183	36	85	144	249	42	225	68	196	55	178	104	129
156	174	47	204	111	80	124	29	132	254	65	236	53	223	27	84
162	91	62	146	33	79	247	54	107	120	40	221	232	78	22	250
31	203	19	69	211	133	23	86	231	240	76	95	165	197	159	41
18	137	251	21	44	235	75	209	28	206	239	142	92	57	16	46
182	11	152	118	56	234	60	89	71	194	99	191	73	90	149	67
153	98	246	222	58	97	170	145	227	83	161	204	93	129	35	14
186	25	212	77	148	32	112	127	66	102	49	125	38	215	223	64
81	107	226	115	233	117	52	219	96	17	228	214	48	150	200	45
175	30	181	130	88	110	186	173	51	164	63	192	235	108	100	168
209	206	43	121	220	117	222	37	245	124	20	208	82	163	151	250
26	148	13	119	59	94	201	158	12	160	74	155	179	97	189	238
50	191	58	34	65	144	77	53	213	95	137	105	21	67	166	111
110	171	81	253	42	196	61	246	70	236	71	237	214	24	199	132
66	57	234	91	48	28	131	99	192	22	222	254	63	114	16	172

IV. Steganographic Scheme

The proposed scheme highlights some most essential steps for information embedding process. First, substituting the information through a highly nonlinear S-box. Secondly, instead of depending on either spatial or transform domain only, it employs a combination of both the domains to reach the acceptable level of embedding capacity as well as robustness. Thirdly, the scheme is based on stronger chaotic combinations which depict extraordinary features when compared with the individual seed maps. Lastly, it involves both DCT and DWT. The combined effect of both these transforms contributes to the robustness of the proposed method.

1) The detailed steganographic strategy is explained through the flowchart (Fig. 5). We explain the whole process in the following steps.

2) Take the host image (sized 512×512) in the spatial domain and shape it into a vector of length m.

3) Take the secret image $\mathcal{I}$ (to be embedded), that is $1/5$ of the host image size and substitute it using the S-box shown in table 1.

4) Break the substituted image into two parts $\mathcal{I}_{\infty}$ and $\mathcal{I}_{\in}$ in the ratio 70% and 30% respectively.

5) First, $\mathcal{I}_{\infty}$ is embedded at random positions of the host image, using the chaotic system *LLS*. This gives the partial stego image in the spatial domain.

6) Reshape this partial stego image into a matrix form and convert into the frequency domain by applying the combination of DWT and DCT.

7) Revamp the obtained image into a vector of length *m* and pick the largest frequency components (30% of the whole).

8) Use the chaotic system *SSS* to embed $\mathcal{I}_{\in}$ at the random positions of the selected largest values. This produces the frequency domain version of the stego image.

9) At this stage, apply the inverse of *DCT* and *DWT* to reach the final version of the stego image.

*10)*The original embedded image can be extracted from the stego image by applying the reverse of the above-explained method.

The simulation results are expressed in Fig. 6, Fig. 7. We apply the steganographic algorithm on two 512×512 images of baboon and Lena.

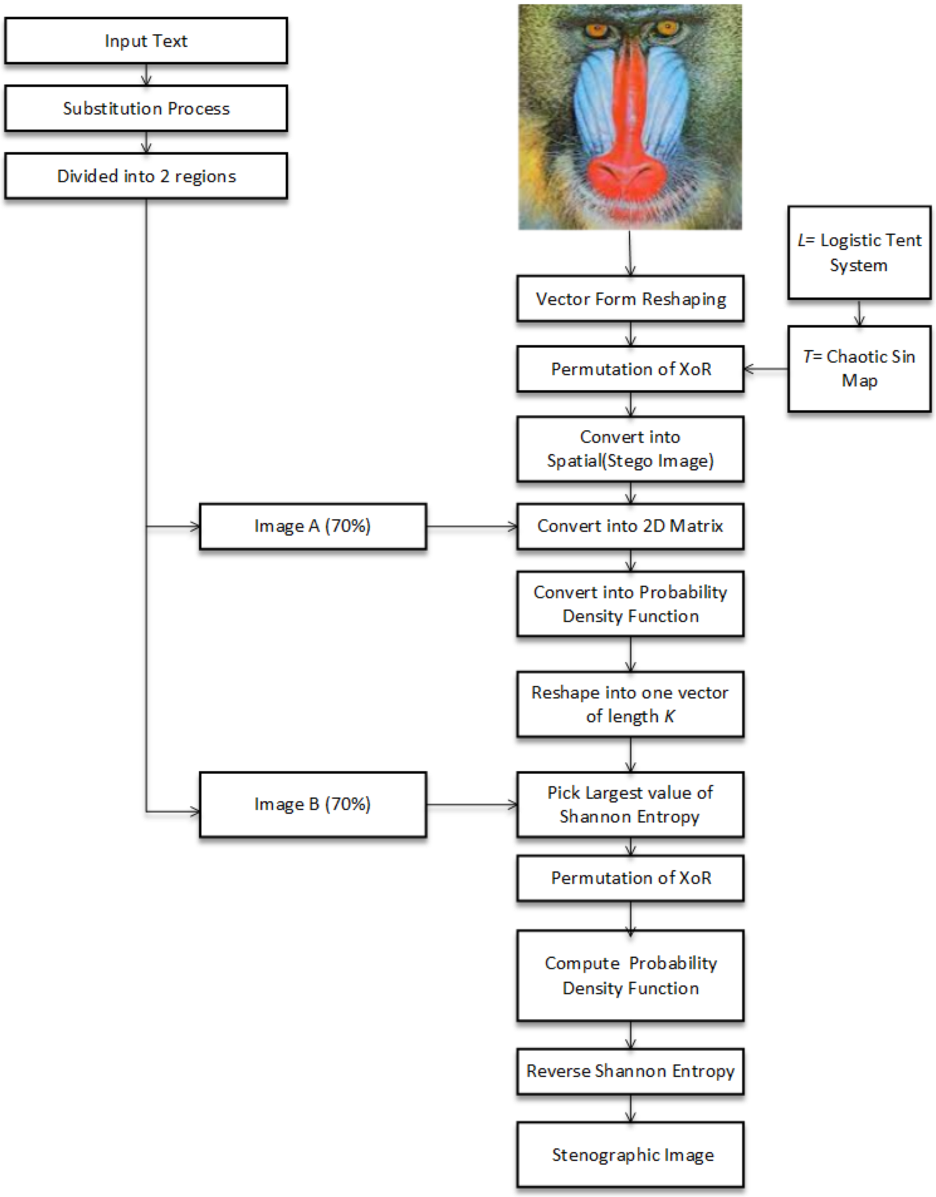
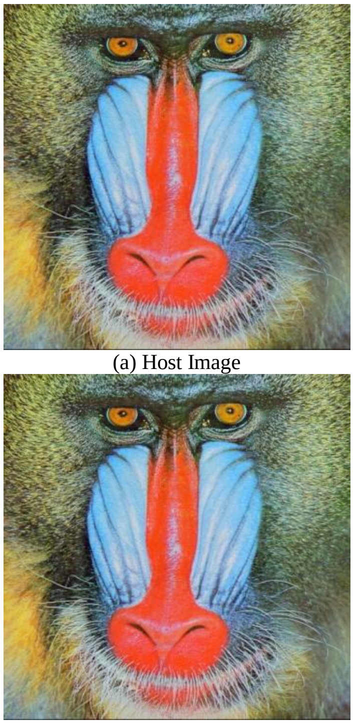


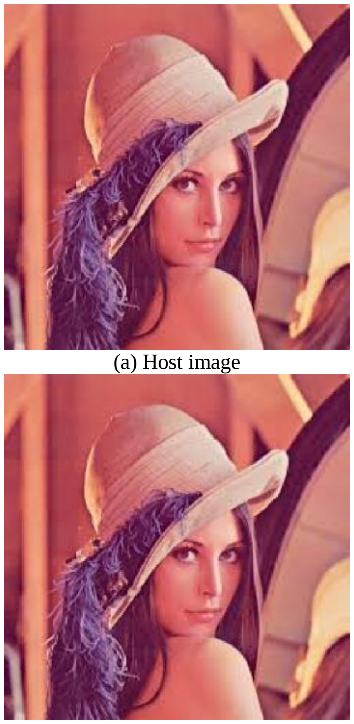
Fig. 5. Steganographic scheme.



(a) Host Image

(b) Stego Image

Fig. 6. Stego images.



(a) Host image

(b) Stego image

Fig. 7. Host and stego images.

V. Statistical Security Analysis

In this section, we measure the cryptographic strength of the proposed scheme with the help of some useful analysis such as entropy, contrast, correlation, homogeneity, peak signal to noise ratio (PSNR) and mean squared error (MSE). We analyzed two benchmark images, baboon and Lena, using the proposed scheme. Notably, the scheme exhibits a remarkable feature of producing steganographic images that have high resemblance to their original counterparts. Moving on, let's discuss these security parameters one by one.

A. *Contrast*

The analysis of contrast is a method employed to evaluate the degree of sensitivity of image textures to variations in intensity. In simpler terms, it measures how much the texture of the image changes when the intensity of the image is changed. The contrast of an image is directly related to the texture of the image, and is an important factor in determining the quality of an image. Images with high contrast are usually considered to be of higher quality, since they have a greater range of intensity and a more distinct texture. The mathematical expression for the contrast is given as:

$$\text{Contrast} = \sum_i \sum_j | -j|^2 p(i,j), \qquad (7)$$

Where Table II and III present the values of contrast for each image, which are computed from the (i,j) th-element of GLCM represented by P(i,j).

B. *Information Entropy*

Entropy analysis is used to measure the randomness or the degree of disorder of a system. Entropy is a critical security parameter used to evaluate the strength of a cryptographic scheme. It is a measure of randomness or uncertainty in the data, indicating the degree of unpredictability in the scheme. In the context of steganography, entropy measures the level of randomness in the distribution of pixels in the steganographic image. A higher entropy value implies that the distribution of pixel values in the steganographic image is more random and unpredictable, making it harder for an attacker to identify the hidden data. In general, higher entropy values are desirable as they indicate a higher level of security. A scheme with a low entropy value is more predictable, and hence less secure, as it makes it easier for an attacker to detect the presence of hidden data. Therefore, a higher entropy value indicates a higher degree of randomness in the scheme and, consequently, a more secure scheme. However, achieving a high entropy value may not always be practical, as it can negatively impact the quality of the steganographic image. A scheme that produces high entropy values but produces a visually distinguishable steganographic image may not be useful in practice. Thus, a balance between high entropy values and image quality needs to be struck to ensure that the scheme is both secure and practical. The mathematical formula for information entropy is given by:

$$\text{Entropy} = -\sum_i \sum_j p(i,j) \log P(i,j), \qquad (8)$$

Tables II and III present the numerical outcomes for both the original and steganographic images. The algorithm we employed demonstrated favorable outcomes for entropy.

TABLE II. Original Image: Results of Majority Logic Criterion

Images	Entropy	Contrast	Correl.	Homog.
Baboon	5.4598	0.3389	0.9689	0.8753
Lena	5.3721	0.1342	0.9752	0.8654

TABLE III. STEGO IMAGE: RESULTS OF MAJORITY LOGIC CRITERION

Images	Entropy	Contrast	Correl.	Homog.
Baboon	5.6790	0.3409	0.9573	0.8845
Lena	4.9612	0.1217	0.9703	0.9603

C. Correlation

Correlation is another important security parameter used to evaluate the strength of a steganographic scheme. It measures the relationship between adjacent pixels in an image, indicating how much the pixel values are dependent on one another. A high correlation value implies that the adjacent pixel values are similar, and hence predictable, whereas a low correlation value indicates that adjacent pixels are less dependent on one another and more unpredictable. In steganography, a lower correlation value is desirable as it indicates a less predictable steganographic image, making it harder for an attacker to detect the hidden data. This is because the insertion of hidden data modifies the pixel values, disrupting the natural correlation between adjacent pixels. Therefore, a low correlation value indicates that the scheme is successful in hiding the data in the image, and thus more secure. However, it is important to note that achieving a low correlation value may not always be feasible, as it may come at the cost of image quality. If the scheme introduces too much noise or distortion into the image to reduce the correlation value, the resulting steganographic image may be of poor quality, making it distinguishable from the original image. Hence, a trade-off between low correlation and image quality needs to be achieved to ensure that the scheme is both secure and practical. Mathematical expression for correlation is given by:

$$Correlation = \sum_i \sum_j \frac{(i-E_X)(j-E_Y)}{D_X D_Y} p(i,j), \quad (9)$$

Where E_X, E_Y, D_X, D_Y represent the expected values and standard deviations of X and Y.

D. Homogeneity

Homogeneity is another security parameter that measures the uniformity of the intensity distribution in an image. It is a measure of the smoothness of the image, indicating how closely the pixel values are clustered around the average value. A higher homogeneity value indicates a more uniform distribution of pixel intensities in the steganographic image, making it more difficult for an attacker to detect the presence of hidden data. In the context of steganography, a high homogeneity value implies that the steganographic image closely resembles the original image, with minimal variation in pixel intensities. This is desirable as it indicates that the hidden data has been successfully embedded into the image without significantly altering its appearance. Thus, a high homogeneity value suggests a more secure scheme. However, as with other security parameters, achieving high homogeneity values while maintaining acceptable image quality can be challenging. A scheme that produces high homogeneity values but results in poor quality steganographic images may not be practical. Hence, a balance between high homogeneity and image quality needs to be achieved to ensure that the scheme is both secure and practical. The numerical results of the homogeneity analysis of the steganographic images are presented in Table II, providing a quantitative measure of the scheme's homogeneity. Its mathematical formula is:

$$Homogeneity = \sum_i \sum_j \frac{P(i,j)}{1+|i-j|} \quad (10)$$

E. Mean Squared Error

It is a security parameter used to evaluate the quality of a steganographic image. It measures the average squared difference between the pixel values of the original and steganographic images. A lower MSE value indicates a smaller difference between the original and steganographic images, and hence a higher quality steganographic image. In the context of steganography, a lower MSE value implies that the scheme has successfully embedded the hidden data into the image without significantly altering its appearance. This is because the scheme has introduced minimal distortion or noise into the image, resulting in a steganographic image that is visually similar to the original image. Thus, a low MSE value suggests a more secure scheme. However, achieving a low MSE value while maintaining high security can be challenging. A scheme that produces low MSE values but results in a visually distinguishable steganographic image may not be practical. Hence, a balance between low MSE values and image quality needs to be achieved to ensure that the scheme is both secure and practical. The numerical results of the MSE analysis of the steganographic images are presented in Table IV, providing a quantitative measure of the scheme's quality.

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} X(i,j) - Y(i,j))^2, \quad (11)$$

Where $X(i,j)$ and $Y\,(i,j)$ represent the reference image and the steganographic image and i,j represents the pixel's position in an $M \times N$ image.

TABLE IV. MSE AND PSNR

Image	MSE	PSNR
Baboon	0.0024	58.2405
Lena	0.0017	57.6436

F. Peak Signal to Noise Ratio

It is a security parameter that measures the quality of a steganographic image by comparing the maximum possible signal value with the level of noise or distortion introduced by the embedding of hidden data. A higher PSNR value indicates a higher quality steganographic image with less noise or distortion. In steganography, a higher PSNR value indicates that the scheme has successfully hidden the data in the image without significantly degrading its quality. This is because a high PSNR value implies that the steganographic image is visually similar to the original image, with minimal distortion or noise. Thus, a high PSNR value suggests a more secure scheme. However, achieving high PSNR values while maintaining acceptable security can be challenging. A scheme that produces high PSNR values but results in a visually distinguishable steganographic image may not be practical. Hence, a balance between high PSNR values and security needs to be achieved to ensure that the scheme is both secure and practical. The numerical results of the PSNR analysis of the steganographic images are presented in Table II, providing a quantitative measure of the scheme's quality. Where, a maximum pixel value in the image is given by MAX_I . PSNR is defined by:

$$\mathrm{PSNR} = 20\log_{10}\frac{\mathrm{MAX_I}}{\sqrt{\mathrm{MSE}}} \tag{12}$$

G. Robustness Analysis

To assess the resilience of the suggested steganographic technique, we perform several tests such as applying JPEG compression, adding noise, and cropping effect on the steganographic images. We determine the resemblance between the extracted steganographic image and the original one. Our algorithm's robustness is demonstrated by the high correlation between these two images. Mathematically, the estimation of similarity can be expressed as:

$$\text{Similarity measure} = \frac{\sum \mathrm{r_i s_i}}{\sqrt{\mathrm{r_i}^2 \mathrm{s_i}^2}} \tag{13}$$

Where r_i and s_i represent the corresponding elements. In the subsequent subsections, we will examine the impact of individual image processing operations one by one.

TABLE V. SIMILARITY UNDER VARIOUS ATTACKS

Attacks	Baboon	Lena
Compression	1.1124×10^{-4}	1.1125×10^{-4}
Noise	1.0906×10^{-4}	1.0957×10^{-4}
Cropping	1.1087×10^{-4}	1.1087×10^{-4}

H. JPEG Compression

In steganography, it is important to evaluate the robustness of the proposed scheme under different scenarios, including image compression. JPEG compression is a widely used lossy compression algorithm that can introduce significant distortion in the image, which may result in loss of hidden data. Therefore, it is crucial to test the proposed scheme's robustness under JPEG compression.

To evaluate the effect of JPEG compression on the steganographic images, we subjected the steganographic images of baboon and Lena to JPEG compression and studied the resultant images. The results of the study are presented in Fig. 8(a) and Fig. 9(a), which show the steganographic images before and after JPEG compression. Additionally, Table V presents the numerical results of the study, which prove the robustness of our proposed scheme. The results of the study demonstrate that the proposed scheme is robust to JPEG compression, as the hidden data remains intact even after compression. This is evidenced by the high values of security parameters such as entropy, correlation, homogeneity, PSNR, and low values of MSE for the compressed steganographic images. Thus, the proposed scheme is suitable for secure data transmission over channels that may introduce compression, such as the internet, where image compression is commonly used to reduce transmission time and bandwidth requirements.

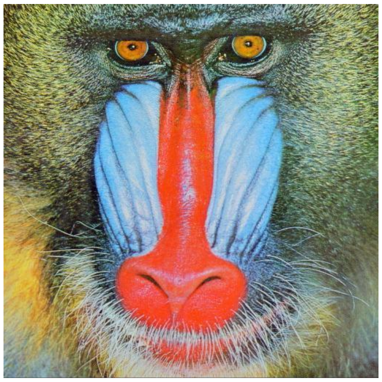
I. Noise Addition

In addition to evaluating the robustness of the proposed scheme under JPEG compression, it is important to assess its performance under other forms of image degradation, such as noise. Salt and pepper noise is a common form of noise that can affect the steganographic images during transmission, and it is important to ensure that the proposed scheme is robust against such attacks.

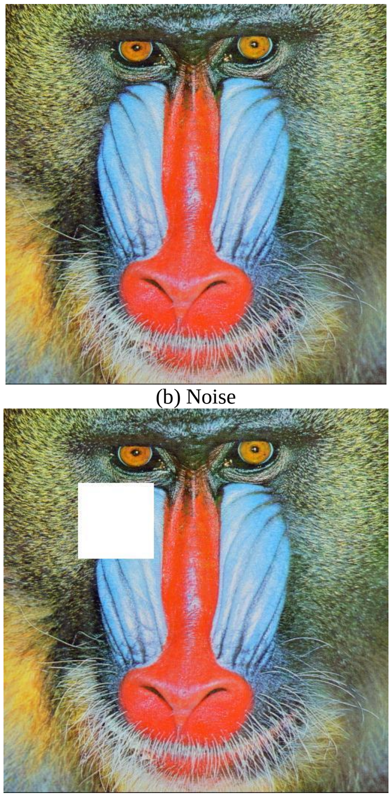
To evaluate the effect of salt and pepper noise on the steganographic images, we subjected the steganographic images of baboon and Lena to varying levels of noise and studied the resultant images. The results of the study are presented in Fig. 8(b) and Fig. 9(b), which show the steganographic images before and after applying salt and pepper noise. Additionally, Table V presents the numerical results of the study, which demonstrate the proposed scheme's robustness against noise attacks. The results of the study indicate that the proposed scheme is quite robust against salt and pepper noise attacks, as the hidden data remains intact even after the introduction of noise. The high values of security parameters such as entropy, correlation, homogeneity, and PSNR, and the low values of MSE for the steganographic images with noise, suggest that the scheme can effectively protect the hidden data from being compromised.

J. Cropping Effect

The cropping effect is an important consideration in evaluating the effectiveness of a steganographic scheme. The cropping of an image refers to the removal of a portion of the image, which can be used to hide the secret data. Therefore, it is essential to assess the robustness of the steganographic scheme against cropping attacks. To evaluate the cropping effect on the proposed scheme, we cropped the steganographic images of baboon and Lena left side. Cropped image is shown in Fig. 8(c) and Fig. 9(c) respectively. Table V presents the numerical results of the study, which demonstrate the robustness of the proposed scheme against cropping attacks. The results of the study indicate that the proposed scheme is quite robust against cropping attacks, as the hidden data remains intact even after the removal of a significant portion of the image. The high values of security parameters such as entropy, correlation, homogeneity, and PSNR, and the low values of MSE for the steganographic images with different cropping percentages, suggest that the scheme can effectively protect the hidden data from being compromised.



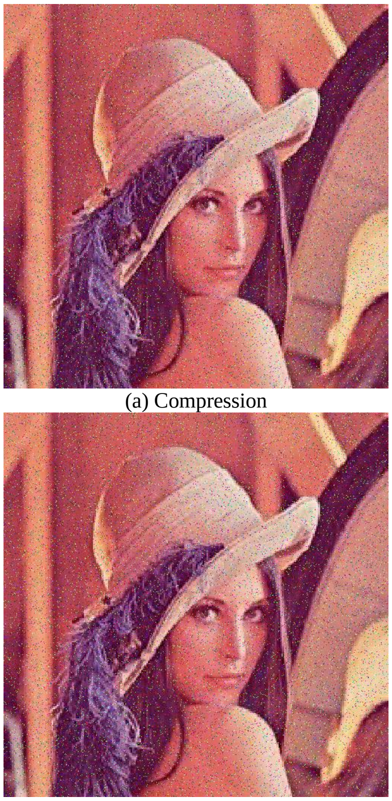
(a) Compression



(b) Noise

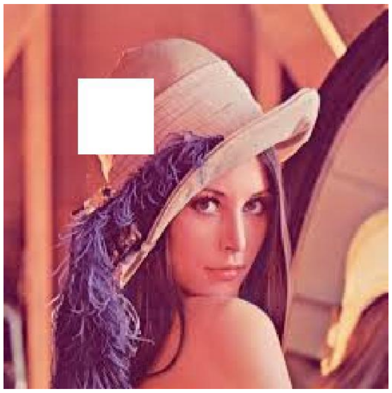
(c) Cropping

Fig. 8. Baboon's image subjected to different image processing techniques.



(a) Compression

(b) Noise



(c) Cropping

Fig. 9. Lena's image subjected to different image processing techniques.

VI. Conclusion

Based on the proposed steganographic algorithm using enhanced chaotic maps, the results of the statistical security analysis demonstrate its robustness and high capacity for securely embedding secret information with good visual quality. The use of a novel fusion technique based on logistic and sine maps in constructing the cryptographic substitution box overcomes existing vulnerabilities of chaotic maps such as frail chaos, finite precision effects, dynamical degradation, and limited control parameters. The algorithm's statistical strength was assessed through several tests, including measures of information entropy, correlation, contrast, energy, homogeneity, peak signal-to-noise ratio, mean squared error, as well as evaluations of the algorithm's performance under JPEG compression and image degradation. These tests demonstrate the algorithm's ability to resist various attacks while maintaining good visual quality. In summary, the proposed steganographic algorithm satisfies the fundamental objectives of achieving both robustness and high-capacity for hidden information, and it offers a secure and effective means of embedding secret information in digital images. The algorithm's strengths in terms of security and visual quality make it a promising tool for applications where the protection of sensitive information is critical.

References

[1] Khan, Muhammad Fahad, et al. "A novel design of cryptographic SP-network based on gold sequences and chaotic logistic tent system." IEEE Access 7 (2019): 84980-84991.

[2] M. Fahad Khan, K. Saleem, M. Alotaibi, M. Mazyad Hazzazi, E. Rehman et al., "Construction and optimization of trng based substitution boxes for block encryption algorithms," Computers, Materials & Continua, vol. 73, no.2, pp. 2679–2696, 2022.

[3] Khan, Muhammad Fahad, Faisal Baig, and Saira Beg. "Steganography between silence intervals of audio in video content using chaotic maps." Circuits, Systems, and Signal Processing 33 (2014): 3901-3919.

[4] Khan, Muhammad Fahad, Adeel Ahmed, and Khalid Saleem. "A novel cryptographic substitution box design using Gaussian distribution." IEEE Access 7 (2019): 15999-16007.

[5] S. T., & Arivazhagan, S. (2019). Universal secret payload location identification in spatial LSB stego images. Annals of Telecommunications, 74(5-6), 273-286.

[6] Siddiqui, Ghazanfar Farooq, et al. "A dynamic three-bit image steganography algorithm for medical and e-healthcare systems." IEEE Access 8 (2020): 181893-181903.

[7] Martín, Alejandro, et al. "Evolving Generative Adversarial Networks to improve image steganography." Expert Systems with Applications (2023): 119841.

[8] Martín, Alejandro, et al. "Evolving Generative Adversarial Networks to improve image steganography." Expert Systems with Applications (2023): 119841.

[9] Daoui, Achraf, et al. "Color stereo image encryption and local zero-watermarking schemes using octonion Hahn moments and modified Henon map." Journal of King Saud University-Computer and Information Sciences 34.10 (2022): 8927-8954.

[10] Gakam Tegue, Gabriel Armand, et al. "A Novel Image Encryption Scheme Combining a Dynamic S-Box Generator and a New Chaotic Oscillator with Hidden Behavior." Arabian Journal for Science and Engineering (2023): 1-20.

[11] Alkhayyat, Ahmed, et al. "A novel 4D hyperchaotic system assisted josephus permutation for secure substitution-box generation." Journal of Signal Processing Systems 94.3 (2022): 315-328.

[12] Tanveer, Muhammad, et al. "Towards a secure and computational framework for internet of drones enabled aerial computing." IEEE Transactions on Network Science and Engineering (2022).

[13] Baig, Faisal, et al. "Onion steganography: a novel layering approach." Nonlinear Dynamics 84 (2016): 1431-1446.

[14] Shah, Tariq, Ayesha Qureshi, and Muhammad Fahad Khan. "DESIGNING MORE EFFICIENT NOVEL S 8 S-BOXES." International Journal on Information Technologies & Security 12.2 (2020).

[15] Kaur, Ishleen, et al. "An integrated approach for cancer survival prediction using data mining techniques." Computational Intelligence and Neuroscience 2021 (2021).

[16] Manzoor, Atif, Amjad Hussain Zahid, and Malik Tahir Hassan. "A new dynamic substitution box for data security using an innovative chaotic map." IEEE Access 10 (2022): 74164-74174.

[17] Khan, Muhammad Fahad, et al. "Human Psychological Disorder towards Cryptography: True Random Number Generator from EEG of Schizophrenics and Its Application in Block Encryption’s Substitution Box." Computational Intelligence and Neuroscience 2022 (2022).

[18] Ahmad, Musheer, et al. "An image encryption algorithm based on new generalized fusion fractal structure." Information Sciences 592 (2022): 1-20.

[19] Khan, Muhammad Fahad, et al. "Block Cipher’s Substitution Box Generation Based on Natural Randomness in Underwater Acoustics and Knight’s Tour Chain." Computational Intelligence and Neuroscience 2022 (2022).

[20] Khan, Muhammad Fahad, et al. "Multilevel information fusion for cryptographic substitution box construction based on inevitable random noise in medical imaging." Scientific reports 11.1 (2021): 1-23.

Implementing Bisection Method on Forex Trading Database for Early Diagnosis of Inflection Point

Agustinus Noertjahyana[1], Zuraida Abal Abas[2], Zeratul Izzah Mohd. Yusoh[3], M. Zainal Arifin[4]
Informatics Department, Petra Christian University, Surabaya, Indonesia[1]
Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia[1, 2, 3]
Informatics Department, State University of Malang, Malang, Indonesia[4]

***Abstract*—Many people are trading in the forex market during the COVID-19 pandemic with the hope of earning money, but they are experiencing shortages due to the lack of information and technology-based tools for existing daily data. Sometimes traders only use moving averages in trading data, even though this information needs to be processed again to get the right inflection point. The objective of this research is to find inflection points based on Forex trading database. Another algorithm can also be used to determine the inflection point between two points on a moving average. This can be supported by the Bisection method used because it can guarantee that convergence will occur. The results show that the points resulting from the bisection calculation on the moving average provide a fairly accurate decision support for the location where the inflection point is located. From 10,000 data there is a standard deviation of 0.71 points which is very small compared to an average of 20 pips (points used as the difference in price values in forex). The use of the bisection method provides an accuracy of the results in seeing the inflection point of 87%.**

Keywords—Bisection method; moving average; inflection point; forex trading; decision support

I. Introduction

Since the COVID-19 pandemic, the value of trade transactions has decreased, this is due to unstable economic conditions and tends to decline and so many speculators do wait and see [1–4]. Based on these problems, this research is focused on finding a solution by processing a database containing transaction data and forex value movements obtained from the meta trader application connected to the broker server [5–7]. From this problem arises the question, how to know the inflection point on a curve that can be seen by the user in real time [8].

Many researchers use the moving average method to assist in making buy or sell decisions based on a certain time frame, but often moving averages are less accurate because errors often occur, especially when there is a change in trend. For this reason, an additional method is needed that can overcome the problems that occur in moving averages, especially in determining inflection points.

The bisection method is a way to determine the value of a curve by dividing the interval in half; this method is used to find the equation of a straight line [9–11]. Because the downward movement of the curve tends to be linear, the researcher uses this method to determine the inflection point [12-13]. The bisection method could help estimate the value of past data so that the results are expected to determine the inflection point properly.

In this research, it will be proved that by looking for inflection points, it can speed up the decision-making process accurately, because the method to be used in this study uses the Bisection Method, which can speed up the search for the midpoint of a line. So, it is expected that between these inflection points an up or down curve can be found which has an important role in decision making.

II. Literature Review

A. Forex Database

To be able to observe price movements in forex trading, a connection is needed to the account of one of the brokers that serves forex trading. To be able to get a database related to forex prices, you can choose a currency pair such as USD and JPY. After selecting a currency pair, price movements can be observed directly depending on the desired timeline. Starting from every 1 minute, 15 minutes, 30 minutes, 1 hour, 4 hours, daily, weekly, and even monthly [14].

To be able to accurately perform formula calculations accurately, of course you cannot use all time frames. For this reason, this study uses a 1-hour and 4-hour and daily time frame to check currency price movements [15-16].

The data obtained from previous researchers is the dataset that will be used in this study. The dataset is not directly used in research but it is necessary to clean the data using SPSS to find out that the data is free from outliers. SPSS helps find statistics from a data set, i.e., statistics that include median, mean, mode, variance, and standard deviation.

The resulting statistical data will be processed by Python to remove invalid data, usually invalid data due to redundancy, political factors that affect foreign exchange rates, certain major factors such as explosions, wars, and others. Many researchers do data cleaning with various parameters that are felt to interfere with research, but with a statistical approach it is expected that the dataset will be of better quality.

Data on currency price movements can be downloaded using the .csv format [17-19] so that later it can be used for further calculation processes. Forex price movement data in the form of candlesticks can be seen in Fig. 1.

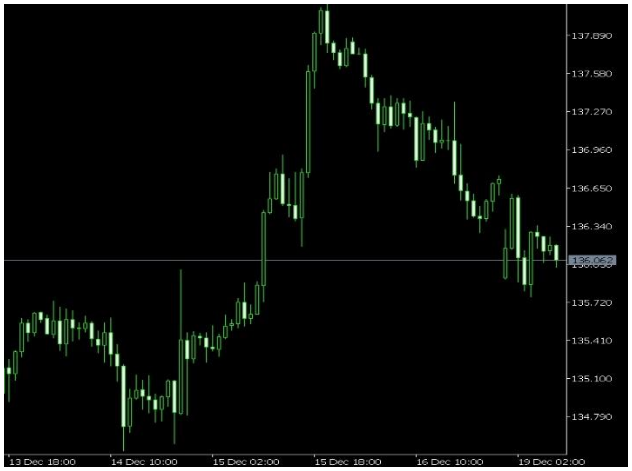
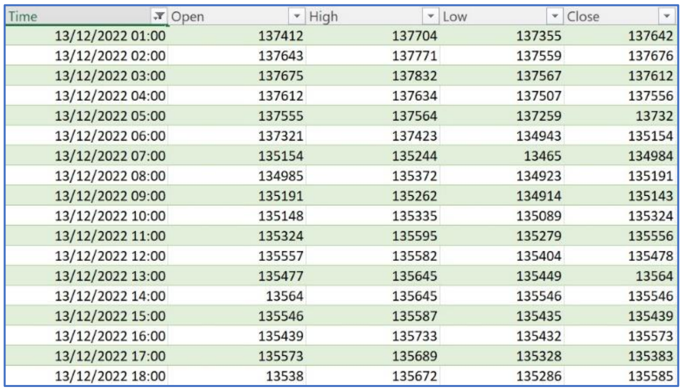


Fig. 1. Candlestick chart for an hourly chart.

To get hourly price data both the price at open, close, high and low can be seen in detail in Fig. 2.



Time	Open	High	Low	Close
13/12/2022 01:00	137412	137704	137355	137642
13/12/2022 02:00	137643	137771	137559	137676
13/12/2022 03:00	137675	137832	137567	137612
13/12/2022 04:00	137612	137634	137507	137556
13/12/2022 05:00	137555	137564	137259	13732
13/12/2022 06:00	137321	137423	134943	135154
13/12/2022 07:00	135154	135244	13465	134984
13/12/2022 08:00	134985	135372	134923	135191
13/12/2022 09:00	135191	135262	134914	135143
13/12/2022 10:00	135148	135335	135089	135324
13/12/2022 11:00	135324	135595	135279	135556
13/12/2022 12:00	135557	135582	135404	135478
13/12/2022 13:00	135477	135645	135449	13564
13/12/2022 14:00	13564	135645	135546	135546
13/12/2022 15:00	135546	135587	135435	135439
13/12/2022 16:00	135439	135733	135432	135573
13/12/2022 17:00	135573	135689	135328	135383
13/12/2022 18:00	13538	135672	135286	135585

Fig. 2. Forex price data per 1 hour in .CSV form.

Open price data indicates the price at the beginning of the new time frame. High data indicates the highest value the price reached during that time frame. Low data indicates the lowest value the price reached during that time frame. Close data indicates the price at the end of the time frame [20-21].

B. *Inflection Point*

The inflection point is the point where a curve will change sign, for example from a curve that tends to decrease in value to change with a value that tends to increase [20]. This change is usually marked with an acceleration that is zero. The acceleration itself is obtained from the second derivative of a function equation; in this study the function is the movement of exchange rate data over time.

In this study, the inflection points where *f''(x)=0* has a relationship with the bisection method because both will stop at a point with a zero value, zero for the acceleration at the inflection point and for the middle value of the Bolzano method [22-23]. The value of *f(x)* in this study is the lowest or highest value of the exchange rate as shown in Fig. 3.

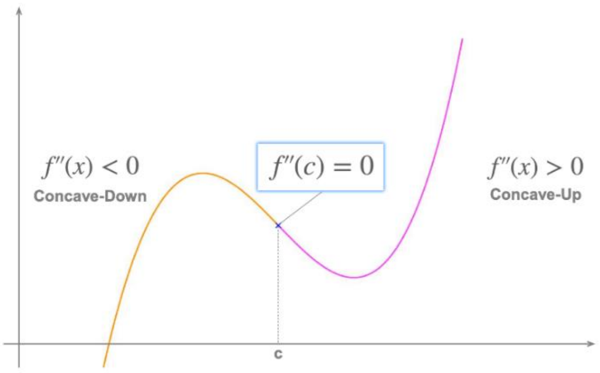


Fig. 3. Example of an inflection point.

C. *Bisection Method*

The Bisection method is a root search algorithm at an interval [24-26]. The interval divides into two parts, then chooses from these two parts which part contains roots and which part does not contain roots is discarded. This is done repeatedly until the roots of the equation are obtained or close to the roots of the equation [27] This method is applicable when you want to solve the equation *f(x)=0* where *f(x)* is a continuous function. Bisection Method can be seen in Fig. 4.

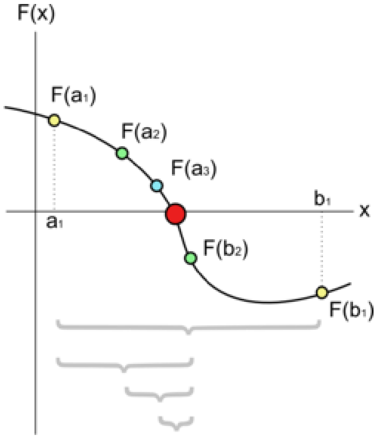


Fig. 4. Bisection method.

Suppose it is guaranteed that f(x) is a continuous function on the interval [a, b] and *f(a)f(b)* < 0. This means that f(x) must at least have roots on the interval [a, b]. Then define the midpoint of the interval [a, b] namely c = (a+b)/2. From this we obtain two subintervals namely [a, c] and [c, b]. After that, check whether *f(a)f(c)* < 0 or *f(b)f(c)* < 0. If *f(a)f(c)* < 0 then b = c (meaning point b is replaced by point c which serves as point b in the next iteration), otherwise a = c. From the first iteration we obtain a new interval [a, b] and a new midpoint c. Then check again as before until you get a small enough error.

III. Methodology

This research uses a simple research method that can be seen in the picture. Before the research was carried out, the researcher collected datasets obtained from meta traders. This dataset is then processed using python to look for data patterns, outliers and dataset variances that are beyond reasonable limits.

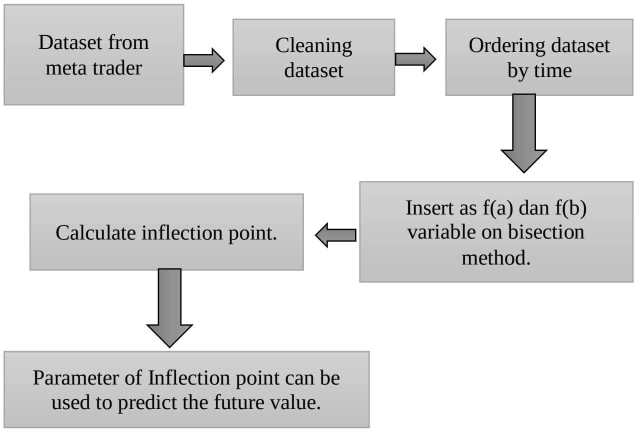

Dataset from meta trader
Cleaning dataset
Ordering dataset by time
Insert as f(a) dan f(b) variable on bisection method.
Calculate inflection point.
Parameter of Inflection point can be used to predict the future value.

Fig. 5. Flowchart of research stages.

After the dataset has been successfully cleaned, the dataset is sorted by time, this aims to make it easier to take points that will be used as points f(a) and f(b). This point is then entered into the bisection formula by determining the new point c=(a+b) / 2, where point c is the midpoint or inflection point. Of course, there are many midpoints, so it is necessary to look for several possibilities from several existing datasets. From this point, it can be predicted that the forex price will go up or down so that better decisions can be made. The flowchart of the various stages as explained can be seen in Fig. 5.

To calculate this inflection point, researchers use the Python programming language which makes it easy to calculate quickly. This algorithm is validated by experts in mathematics and when the validation is appropriate then it is tested on a server, the researcher uses a 10-core server with 16GB RAM. The data entered in the algorithm consists of a one-month timeframe.

TABLE I. VALIDATOR QUESTIONNAIRE

Question Number	Question
1	Is the bisection formula made in accordance with the journal?
2	is the transformation of the formula to the python program appropriate?
3	Have the computational results been compared with the original formula?
4	Are the results of the estimated value in accordance with statistical calculations?

In Table I, the validator fills in the maximum value or approves it for the next stage.

IV. RESULT AND DISCUSSION

The dataset in this study was taken from secondary data on export results in the form of .csv from the Metatrader software. This dataset consists of four data types, namely: High, Low, Open, Close. The next process is to clean the data by not entering datasets that exceed the upper and lower limits based on statistical analysis which includes median, mean, standard deviation and variance.

After the dataset is already cleaned, the next step is to run the dataset in an application made using Python. Python program code can be seen in Algorithm 1.

Algorithm 1:

```
def bisection (f, a, b, tol=1e-6, max_iter=100):
   if f(a) * f(b) >= 0:
     raise ValueError("Interval is not valid: f(a) and f(b) has similar sign.")
  c = (a + b) / 2
  iter_count = 0
  while abs(f(c)) > tol and iter_count < max_iter:
    if f(a) * f(c) < 0:
      b = c
    else:
      a = c
    c = (a + b) / 2
    iter_count += 1

  if iter_count == max_iter:
    print("Outbound Maximum Values")
  return c
```

From the dataset point entered in the program code, it will produce a curve as displayed in Fig. 6.

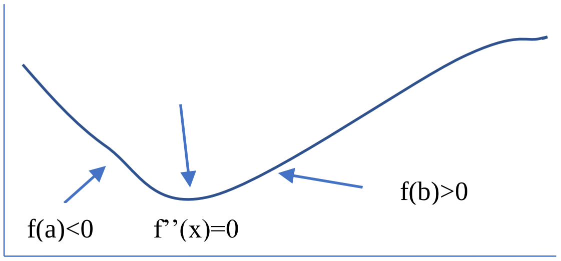


Fig. 6. Graph showing how the bisection program code work in python.

After the application is run, the results will look like in Table II.

TABLE II. RESULTS OF RUNNING MULTIPLE DATASETS

f(a)	f(b)	c (Previous Research)	Predicted (Current Research)
1914.760	1905.340	1910.050	1912.900
1913.450	1909.780	1911.615	1912.250
1918.530	1912.060	1915.295	1916.490
1921.720	1915.360	1918.540	1919.550
1925.080	1920.590	1922.835	1921.090
1923.610	1920.090	1921.850	1922.480
1922.870	1918.920	1920.895	1921.590
1925.440	1921.410	1923.425	1922.870
1925.400	1922.300	1923.850	1924.700
1934.970	1924.520	1929.745	1930.030
1933.780	1930.490	1932.135	1932.080
1934.430	1927.820	1931.125	1931.440
1936.400	1932.260	1934.330	1933.390
1934.760	1929.960	1932.360	1931.830
1941.920	1929.700	1935.810	1935.740

From the table it appears that the predicted value using bisection has a small error range when compared between the 'c' column and the 'predicted' column. With accuracy of an inflection point is 87% by SPSS and standard deviation about 0.71 and 20 pips moving. This will help users when they get information that after inflection there has been an increase in forex prices.

V. Conclusion

The research resulted in several conclusions, namely: a combination of bisection formulas on moving averages that work well. This is because both have the same scalar properties. In addition, the error rate of both algorithms is less than 3%. Furthermore, including optimization in determining future prediction values has been carried out by the bisection method even though it still has an error, but the error is still below 5%; this gives a sense of optimism for users to make decisions on buying or selling forex trading. This study was conducted in several experiments and ran well in the Python language environment with the objective achieved.

Acknowledgment

Special thanks to validator Zainal Arifin from State University of Malang, thanks to Petra Christian University, and Universiti Teknikal Malaysia Melaka.

References

[1] I. O. Fasanya, O. Oyewole, O. B. Adekoya, and J. Odei-Mensah, “Dynamic spillovers and connectedness between COVID-19 pandemic and global foreign exchange markets,” Taylor & Francis, vol. 34, no. 1, pp. 2059–2084, 2020, doi: 10.1080/1331677X.2020.1860796.

[2] C. Elleby, I. P. Domínguez, M. Adenauer, and G. Genovese, “Impacts of the COVID-19 Pandemic on the Global Agricultural Markets,” Environmental and Resource Economics, vol. 76, no. 4, pp. 1067–1079, Aug. 2020, doi: 10.1007/S10640-020-00473-6.

[3] D. Yu et al., “Nationwide lockdown, population density, and financial distress brings inadequacy to manage covid-19: Leading the services sector into the trajectory of global depression,” Healthcare (Switzerland), vol. 9, no. 2, Feb. 2021, doi: 10.3390/HEALTHCARE9020220.

[4] S. M. Jowitt, “COVID-19 and the Global Mining Industry,” SEG Discovery, no. 122, pp. 33–41, Jul. 2020, doi: 10.5382/SEGNEWS.2020-122.FEA-02.

[5] D. Yıldırım, I. Toroslu, U. F.-F. Innovation, and undefined 2021, “Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators,” Springer.

[6] X. Meng, C. H.-T. N. A. J. of E. and, and undefined 2019, “The time-frequency co-movement of Asian effective exchange rates: A wavelet approach with daily data,” Elsevier.

[7] S. Seifollahi and M. Shajari, “Word sense disambiguation application in sentiment analysis of news headlines: an applied approach to FOREX market prediction,” Journal of Intelligent Information Systems, vol. 52, no. 1, pp. 57–83, Feb. 2019, doi: 10.1007/S10844-018-0504-9.

[8] O. Amadeo Vilca-Huayta, U. Nacional, A. Puno, P. Ubaldo, and Y. Tito, “Efficient Function Integration and a Case Study with Gompertz Functions for Covid-19 Waves Professor Department of System Engineering,” IJACSA) International Journal of Advanced Computer Science and Applications, vol. 13, no. 8, p. 2022, Accessed: Mar. 01, 2023. [Online]. Available: www.ijacsa.thesai.org.

[9] I. F. D. Oliveira and R. H. C. Takahashi, “An Enhancement of the Bisection Method Average Performance Preserving Minmax Optimality,” ACM Transactions on Mathematical Software, vol. 47, no. 1, Jan. 2021, doi: 10.1145/3423597.

[10] Q. Xia et al., “Safety risk assessment method for thermal abuse of lithium-ion battery pack based on multiphysics simulation and improved bisection method,” Elsevier.

[11] M. Singh, V. Rajendran, J. P.-C. M. Science, and undefined 2020, “A novel bisection method based algorithm to quantify interphase in epoxy alumina nanocomposites,” Elsevier.

[12] E. J. Prentice et al., “The inflection point hypothesis: The relationship between the temperature dependence of enzyme-catalyzed reaction rates and microbial growth rates,” Biochemistry, vol. 59, no. 38, pp. 3562–3569, Sep. 2020, doi: 10.1021/ACS.BIOCHEM.0C00530.

[13] W. T. Xu, J. Liu, T. J. Gao, and Z. K. Guo, “Gravitational waves from double-inflection-point inflation,” Physical Review D, vol. 101, no. 2, Jan. 2020, doi: 10.1103/PHYSREVD.101.023505.

[14] H. Pan, “Strength Investing - A Computable Methodology for Capturing Strong Trends in Price-Time Space in Stocks, Futures, and Forex,” Advances in Intelligent Systems and Computing, vol. 1009 AISC, pp. 53–60, 2020, doi: 10.1007/978-3-030-38227-8_7.

[15] N. Weeraddana, … A. S.-… on A. in I. for, and undefined 2018, “Detection of black regions in the forex market by analyzing high-frequency intraday data,” ieeexplore.ieee.org.

[16] M. Raimundo, J. O. J.-I. J. of M. and, and undefined 2018, “Application of Hurst Exponent (H) and the R/S Analysis in the Classification of FOREX Securities,” ijmo.org, doi: 10.7763/IJMO.2018.V8.635.

[17] P. Vuong, T. Dat, … T. M.-… S. S. and, and undefined 2022, “Stock-price forecasting based on XGBoost and LSTM,” digital.lib.ueh.edu.vn.

[18] J. Carapuço, R. Neves, N. H.-A. S. Computing, and undefined 2018, “Reinforcement learning applied to Forex trading,” Elsevier.

[19] D. C. Yıldırım, I. H. Toroslu, and U. Fiore, “Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators,” Financial Innovation, vol. 7, no. 1, Dec. 2021, doi: 10.1186/S40854-020-00220-2.

[20] B. Donnelly, “The Art of Currency Trading: A Professional’s Guide to the Foreign Exchange Market,” 2019.

[21] G. J. Berman, G. A. Kane, G. alo Lopes, J. L. Saunders, A. Mathis, and M. W. Mathis, “Real-time, low-latency closed-loop feedback using markerless posture tracking,” elifesciences.org, doi: 10.7554/eLife.61909.

[22] G. Mahesh, G. Swapna, K. V.-J. Math. Comput. Sci., and undefined 2020, “An iterative method for solving non-linear transcendental equations,” scik.org, vol. 10, no. 5, pp. 1633–1642, 2020, doi: 10.28919/jmcs/4723.

[23] I. Azure, G. Aloliga, L. D.-Math. Lett, and undefined 2019, “Comparative study of numerical methods for solving non-linear equations using manual computation,” researchgate.net.

[24] H. Li, PhD, “Finding Roots of Equations,” Numerical Methods Using Java, pp. 207–228, 2022, doi: 10.1007/978-1-4842-6797-4_3.

[25] G. Levkine, “Topological bisection method for solving of system of two equation,” 2020.

[26] C. Aviles-Ramos, K. T. Harris, and A. Haji-Sheikh, “A hybrid root finder,” Integral methods in science and engineering, pp. 41–45, Jun. 2019, doi: 10.1201/9780429123634-5/HYBRID-ROOT-FINDER-AVILES-RAMOS-HARRIS-HAJI-SHEIKH.

[27] F. N. Zghoul, “An Extension of the Bisection Theorem to Symmetrical Circuits with Cross-Coupling,” International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, 2016, doi: 10.14569/IJACSA.2016.070658.

A Novel Approach to Network Forensic Analysis: Combining Packet Capture Data and Social Network Analysis

Irwan Sembiring[1], Suharyadi[2], Ade Iriani[3], Jenni Veronika Br Ginting[4], Jusia Amanda Ginting[5]
Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia[1, 2, 3]
Institut Teknologi dan Bisnis Indonesia, Medan, Indonesia[4]
Univeristas Bunda Mulia, Jakarta, Indonesia[5]

***Abstract*—Log data from computers used for network forensic analysis is ineffective at identifying specific security threats. Log data limitations include the difficulty in reconstructing communication patterns between nodes and the inability to identify more advanced security threats. By combining traditional log data analysis methods with a more effective combination of approaches, a more comprehensive view of communication patterns can be achieved. This combined approach can then help identify potential security threats more effectively. It's difficult to determine the specific benefits of combining Packet Capture (PCAP) and Social Network Analysis (SNA) when performing forensics. This article proposes a new approach to forensic analysis that combines PCAP and social network analysis to overcome some of the limitations of traditional methods. The purpose of this discovery is to improve the accuracy of network forensic analysis by combining PCAP and social network analysis to provide a more comprehensive view of network communication patterns. Network forensics, which combines pcap analysis and social network analysis, provides more comprehensive results. PCAP analysis is used to analyze network traffic, conversation statistics, protocol distribution, packet content and round-trip times. Social network analysis maps communication patterns between nodes and identifies the most influential key players within the network. PCAP analysis efficiently captures and analyzes network packets, and SNA provides insight into relationships and communication patterns between devices on the network.**

Keywords—PCAP analysis; social network analysis; network forensic; network communication pattern

I. Introduction

Network forensic analysis is an important tool for identifying and tracking malicious activities on the network [1][2]. Traditional methods of network forensic analysis using log data are not effective in identifying specific security threats [3]. The fundamental causes of these issues are the log data's limits in recreating communication patterns between nodes and its inability to recognize more sophisticated threats. [4]. In recent years, there has been increasing focus on combining packet capture analysis and social network analysis in network forensic analysis to improve the accuracy and completeness of the analysis. PCAP analysis involves capturing and analyzing data packets transmitted over a network, while social network analysis involves visualizing and analyzing the relationships and communication patterns between devices on the network [1][5][6][7]. The research question is how social network analysis can be used in network forensics to identify potential suspects and their relationships within a network.

The combination of these two approaches has the potential to provide a more comprehensive view of network communication patterns and more effectively identify potential security threats. However, little is known about the specific benefits of combining PCAP and social network analysis in network forensic analysis. In this paper, the main contribution is to propose a new approach to network forensic analysis combining PCAP and social network analysis to address the challenges and limitations of traditional approaches.

The goal of this discovery is to combine PCAP and social network analysis to improve the accuracy of network forensic analysis by providing a more comprehensive view of network communication patterns. By combining social network analysis and PCAP analysis, it is possible to gain a deeper knowledge of network activity and communication patterns. Investigators can use this to find unusual or suspect activities on the network, such as secret or encrypted communication.

II. Related Work

A. Research Related to Packet Capture (PCAP) Analysis

Sikos [1] made a comprehensive comparison of Carnivore, Snort, Windump, Wireshark, dsniff, tcpdump, Omnipeek, Solarwind and other packet analysis tools when analyzing PCAP data. The method used is AI-based deep learning inspection combined with semi-supervised machine learning. The research aims to compare the ability to recognize patterns in different packet analyzer applications in order to find the most suitable tool for network forensic analysis activities. The results of DPI (Deep Packet Inspection), a packet analysis tool with machine learning capabilities are valid [8][9][10].

Cappers et al. [5] focused on data reduction and visualization techniques using the EventPad tool. The purpose of this study was to conduct a safety analysis in a behavioral pattern study using PCAP data. This study presents a case study of the EventPad visual analysis tool to obtain attack profiles and traffic analysis using rules and aggregations. The study did not describe any communication patterns at the application level.

Shrivastava et al. [11] focused on capturing attacks on IoT devices using Cowrie honeypots and using machine learning to classify attack types. They apply various machine learning algorithms namely Naive Bayes, J48 Decision Trees, Random Forest, and Support Vector Machines (SVM) to classify attacks such as malicious payloads, SSH attacks, XOR DDoS, espionage, suspicious and clean attacks. Perform feature selection using subset evaluation and best-first search. The training results achieved an accuracy rate ranging from 67.7% to 97.39%.

B. Network Analysis Research using Social Network Analysis (SNA)

Chakraborty et al. [12] conducted a study that advances the understanding of 5G-COVID-19 conspiracy theories. This paper conducts a social network analysis to analyze the content of Twitter data over a seven-day period (the #5GCoronavirus hashtag became trending on Twitter in the UK. The content analysis revealed that 34.8% (n=81) of a sample of 233 tweets contained references to 5G and COVID-19-related opinions, 32.2% (n=75) were critical of conspiracy theories, 33.0% (n=77) were general tweets, not disclosing views or personal opinions) tweets were from non-conspiracy theory supporters, indicating that despite interest in the topic is high, but only a small percentage of users actually believe in the conspiracy theory. Liu et al. [13] provided a large-scale group decision-making model based on the process of propagating beliefs; the process of conflict detection and resolution; and the process of selection using social network analysis methods. In the first procedure, we propose a relation strength-based belief propagation operator, which allows building a complete social network while considering the effect of relation strength on propagation efficiency. In the second procedure, we define the notion of degree of conflict and measure the degree of collective conflict in conjunction with assessments of information and belief relationships among large groups of decision makers. SNA is a modeling of users represented by nodes and interactions between users are represented by lines (edges). This analysis is needed because it brings new opportunities to understand individuals or communities regarding their social interaction patterns [20] [21]. SNA can be used to study network patterns of organizations, ideas, and people who are connected in various ways in an environment [22] [23]. Degree centrality counts the number of connections or interactions a node has. To calculate the value of centrality degree (CD), we use Eq. (1) [24].

$$CD(i) = \sum_{\substack{j=1 \\ i\neq j}}^{N} Xij \tag{1}$$

Closeness centrality (CC) calculates the average distance between a node and all other nodes on the network. This measure describes the proximity of this node to other nodes [24] as in Eq. (2).

$$cc(i) = \frac{N-1}{\sum_{J=1}^{N} dij} \tag{2}$$

Betweenness centrality (BC) calculates how often a node is passed by another node to go to a certain node in the network. This value serves to determine the role of the actor who is the bridge that connects the interaction in the network. To calculate the value of degree centrality we use Eq. (3) [24].

$$Cb(i) = \sum_{J=1}^{N} \sum_{k=1}^{j-1} \frac{gjk(i)}{gjk} \tag{3}$$

One of the most important processes at the digital forensic stage is data integrity in the preservation section. Message-digest algorithms MD5 and SHA-1 as one-way cryptographic hash functions are used in integrity validation [23] [24]. Four non-linear functions in 512bit blocks in the MD5 Algorithm as Eq. (4).

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$

$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$

$$H(B,C,D) = B \oplus C \oplus D$$

$$I(B,C,D) = C \oplus (B \vee \neg D) \tag{4}$$

C. PCAP Data and Network Forensics Analysis

PCAP analysis in cyber forensics can be performed using a variety of methods including using software, PCAP software analysis. The software can be used to view and analyze packet content, including headers and payloads, and look for signs of malicious activity [14]. The next approach is statistical analysis, which extracts statistics from PCAP, such as packet count, network traffic and protocol statistics [15]. A rather important approach is packet and payload analysis to extract information from packet headers such as: B. Source and destination IP addresses, protocol and port used, and payload from the packet [10][11] [17][18]. Other findings indicate that computer network traffic results provide a variety of valuable information in graphical form to help identify routine banking transactions (pooled accounts, straw men, smurfing) used to hide movement of prohibited resources or obfuscation, thereby enhancing the visualization of financial analysis aspect. Packet analysis of internet network traffic is an important backtracking technique in network forensics, if the captured packet details are detailed enough, it can even show all network traffic at a specific point in time. This can be used to detect traces of malicious online behavior, data breaches, unauthorized website access, malware infections, and infiltration attempts, and to reconstruct image files, documents, email attachments, and other content sent over the network [1][15] [16][19].

We recommend combining PCAP analysis and social network analysis. This combination will demonstrate advanced network forensic analysis by revealing communication patterns between specific nodes in social media interactions. Combining PCAP and SNA to analyze network forensic activity can be a powerful method for identifying and analyzing malicious activity on the network. The integration of these two technologies can take advantage of the detailed information provided by PCAP data and the broader network-level view provided by SNA. An example of how this combination could be used is to use PCAP data to identify specific patterns or anomalies in network traffic that are consistent with known malicious activity, such as: B. Botnet command and control traffic or data exfiltration. Once these patterns are identified, SNA can be used to identify nodes and edges in the network that match these patterns, which helps identify the source and destination of malicious activity and the relationships between these entities.

III. Combining PCAP Data and Social Network Analysis for Network Forensic Activity

This research uses an experimental method in a laboratory where the independent variable is the number of nodes in social media capturing, and the dependent variable is the result of network forensic analysis, shown in Fig. 1.

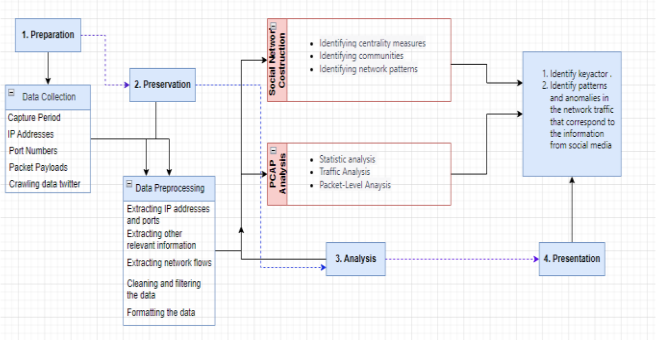

1. Preparation
2. Preservation
Data Collection
Capture Period
IP Addresses
Port Numbers
Packet Payloads
Crawling data twitter
Data Preprocessing
Extracting IP addresses and ports
Extracting other relevant information
Extracting network flows
Cleaning and filtering the data
Formatting the data
Social Network Costruction
• Identifying centrality measures
• Identifying communities
• Identifying network patterns
PCAP Analysis
• Statistic analysis
• Traffic Analysis
• Packet-Level Anaysis
1. Identify keyactor .
2. Identify patterns and anomalies in the network traffic that correspond to the information from social media
3. Analysis
4. Presentation

Fig. 1. Combining PCAP data and Social Network Analysis (SNA) for network forensic methodology.

A. Data Collection

The first step is to collect PCAP data from the network using an application such as Wireshark or tcpdump. This data includes information about network traffic that occurred during the collection period, including details such as source and destination IP addresses, port numbers, and packet payloads. During the collection process, the keyword "Manchester United" was used to crawl data from Twitter, and 1000 nodes were obtained. Twitter scraping was performed on January 4, 2023 using the netlytic application.

B. Data Preprocessing

This step is performed to filter out packets that are not relevant to the current investigation. For example, only save packets from suspicious IP addresses or use protocols considered important. Parsing, this step is performed to extract relevant information from the PCAP packet. Information that can be extracted includes source and destination IP addresses, protocols used, timestamps, and payload data. Anonymization, this step is performed to remove information that could be used to identify individuals participating in the communication. Information that can be removed includes IP addresses, MAC addresses, and personal information contained in payload data. Normalization, this step is performed to convert the data extracted from the PCAP packets into a format that is easier to use for analysis. For example, converting timestamps to a more readable format or converting used logs to a simpler format. Aggregation, this step is performed to combine data from multiple PCAP packets into a larger unit. For example, combining multiple packets from the same IP address into a larger unit. Enrichment, this step is performed to add additional information to the data extracted from PCAP packets. Additional information can be in the form of IP geolocation information, WHOIS information, or IP reputation information.

C. PCAP Analysis

Traffic Analysis, this step involves analyzing data to identify patterns and anomalies in network traffic. This may include identifying unusual traffic destinations, unusual traffic patterns, or specific protocols used. Packet level analysis, this step involves examining individual packets in the PCAP data such as: B. Source and destination IP addresses, source and destination ports, and packet payload. This can be used to identify specific keywords, extract files, or extract other information from the payload. Statistical Analysis, this is the process of analyzing and interpreting the data contained in a PCAP file using statistical methods. This can include identifying patterns, trends, and anomalies in network traffic, as well as estimating various metrics such as traffic volume, packet size, and packet arrival time.

D. Social Network Construction

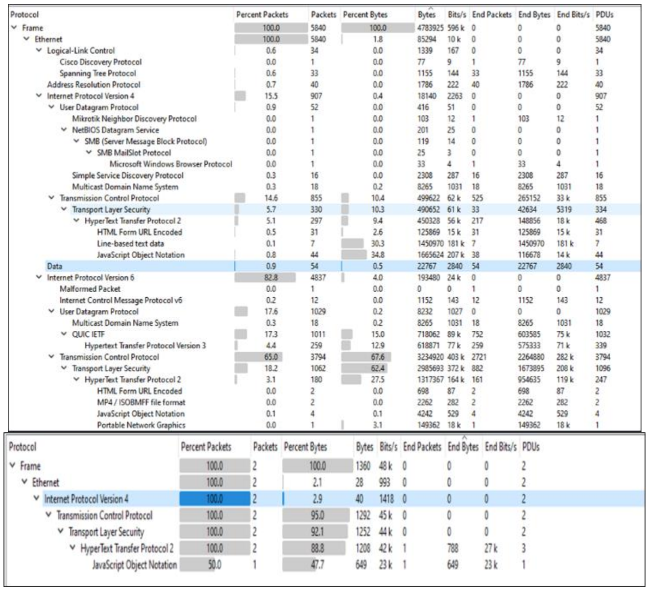
Social network construction creates representations of relationships between individuals or entities in a social network. This may involve identifying relationships between individuals, such as B. friendships, family ties, or professional relationships, and the strength of these relationships, such as B. frequency of personal interaction or communication. These centrality measures can be applied to social networks created from PCAP data and help identify key IP addresses or ports that may be important for understanding communication patterns and how information or malware spreads in the network. Degree centrality, this measure is based on the number of edges (connections) a node has. Nodes with high centrality are those that have many connections in the network. Betweenness centrality, this measure is based on the number of shortest paths through a node. Nodes with high betweenness centrality are those located on many shortest paths, considered as bridges or gatekeepers between different communities. Closeness centrality this measure is based on the average distance of a node from all other nodes in the network. Nodes with high proximity centrality are those that are close to many other nodes in the network. Community discovery is an important task in social network analysis that aims to identify groups of nodes (communities) that are more connected to each other than to the rest of the network. There are various methods for discovering communities on the web. Network pattern recognition is the process of identifying structural patterns in a network, which can provide insight into network organization and function. There are several methods for identifying patterns in networks; some of the most popular are subgraph counts and clustering coefficients.

IV. Experiment

Conversation statistics IP source 192.168.1.14 to 104.211.42.0 show the wireshark session statistics to see which devices are communicating with each other. This data also counts the traffic exchanged between these devices. It helps to understand the communication patterns in the network. Protocol Distribution Use Wireshark Hierarchical Protocol Statistics to see which protocols are used on the network and how much traffic they generate. It shows the distribution of logs and the number of packages present. Fig. 2 shows the number of packets, namely 5840, distributed among the different protocols and conversation statistics.

A. Authors and Affiliations

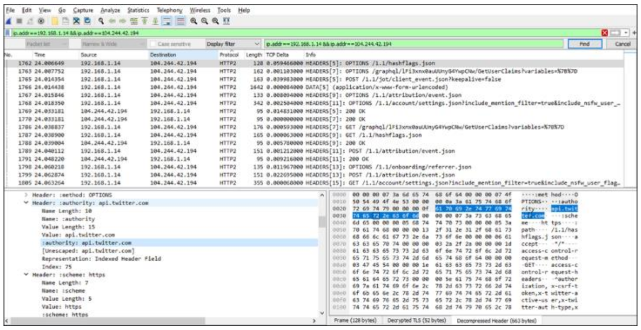
Fig. 3 shows the distribution of content packets in Wireshark's packet details panel to examine the contents of individual packets and see their structure. It can help understand the data exchanged in the network and identify potential problems. Fig. 4 shows that round trip time (RTT) is a measure of how long it takes to send a packet of data from one device to another and receive a response. RTT can use this value to measure network latency and identify potential delays. = 55 milliseconds.



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
⌄ Frame	100.0	5840	100.0	4783925	596 k	0	0	0	5840
⌄ Ethernet	100.0	5840	1.8	85294	10 k	0	0	0	5840
⌄ Logical-Link Control	0.6	34	0.0	1339	167	0	0	0	34
Cisco Discovery Protocol	0.0	1	0.0	77	9	1	77	9	1
Spanning Tree Protocol	0.6	33	0.0	1155	144	33	1155	144	33
Address Resolution Protocol	0.7	40	0.0	1786	222	40	1786	222	40
⌄ Internet Protocol Version 4	15.5	907	0.4	18140	2263	0	0	0	907
⌄ User Datagram Protocol	0.9	52	0.0	416	51	0	0	0	52
Mikrotik Neighbor Discovery Protocol	0.0	1	0.0	103	12	1	103	12	1
⌄ NetBIOS Datagram Service	0.0	1	0.0	201	25	0	0	0	1
⌄ SMB (Server Message Block Protocol)	0.0	1	0.0	119	14	0	0	0	1
⌄ SMB MailSlot Protocol	0.0	1	0.0	25	3	0	0	0	1
Microsoft Windows Browser Protocol	0.0	1	0.0	33	4	1	33	4	1
Simple Service Discovery Protocol	0.3	16	0.0	2308	287	16	2308	287	16
Multicast Domain Name System	0.3	18	0.2	8265	1031	18	8265	1031	18
⌄ Transmission Control Protocol	14.6	855	10.4	499622	62 k	525	265152	33 k	855
⌄ Transport Layer Security	5.7	330	10.3	490652	61 k	33	42634	5319	334
⌄ HyperText Transfer Protocol 2	5.1	297	9.4	450328	56 k	217	148856	18 k	468
HTML Form URL Encoded	0.5	31	2.6	125869	15 k	31	125869	15 k	31
Line-based text data	0.1	7	30.3	1450970	181 k	7	1450970	181 k	7
JavaScript Object Notation	0.8	44	34.8	1665624	207 k	38	116678	14 k	44
Data	0.9	54	0.5	22767	2840	54	22767	2840	54
⌄ Internet Protocol Version 6	82.8	4837	4.0	193480	24 k	0	0	0	4837
Malformed Packet	0.0	1	0.0	0	0	1	0	0	1
Internet Control Message Protocol v6	0.2	12	0.0	1152	143	12	1152	143	12
⌄ User Datagram Protocol	17.6	1029	0.2	8232	1027	0	0	0	1029
Multicast Domain Name System	0.3	18	0.2	8265	1031	18	8265	1031	18
⌄ QUIC IETF	17.3	1011	15.0	718062	89 k	752	603585	75 k	1032
Hypertext Transfer Protocol Version 3	4.4	259	12.9	618871	77 k	259	575333	71 k	339
⌄ Transmission Control Protocol	65.0	3794	67.6	3234920	403 k	2721	2264880	282 k	3794
⌄ Transport Layer Security	18.2	1062	62.4	2985693	372 k	882	1673895	208 k	1096
⌄ HyperText Transfer Protocol 2	3.1	180	27.5	1317367	164 k	161	954635	119 k	247
HTML Form URL Encoded	0.0	2	0.0	698	87	2	698	87	2
MP4 / ISOBMFF file format	0.0	2	0.0	2262	282	2	2262	282	2
JavaScript Object Notation	0.1	4	0.1	4242	529	4	4242	529	4
Portable Network Graphics	0.0	1	3.1	149362	18 k	1	149362	18 k	1

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
⌄ Frame	100.0	2	100.0	1360	48 k	0	0	0	2
⌄ Ethernet	100.0	2	2.1	28	993	0	0	0	2
⌄ Internet Protocol Version 4	100.0	2	2.9	40	1418	0	0	0	2
⌄ Transmission Control Protocol	100.0	2	95.0	1292	45 k	0	0	0	2
⌄ Transport Layer Security	100.0	2	92.1	1252	44 k	0	0	0	2
⌄ HyperText Transfer Protocol 2	100.0	2	88.8	1208	42 k	1	788	27 k	3
JavaScript Object Notation	50.0	1	47.7	649	23 k	1	649	23 k	1

Fig. 2. Conversation statistic and protocol distribution.



```
ip.addr==192.168.1.14 && ip.addr==104.244.42.194

No.  Time       Source          Destination     Protocol Length TCP Delta   Info
1762 24.006649  192.168.1.14    104.244.42.194  HTTP2     128 0.059466000 HEADERS[5]: OPTIONS /1.1/hashflags.json
1763 24.007752  192.168.1.14    104.244.42.194  HTTP2     162 0.001103000 HEADERS[7]: OPTIONS /graphql/lFi3xnx0auUUnyG4YwpCNw/GetUserClaims?variables=%7B%7D
1765 24.014354  192.168.1.14    104.244.42.194  HTTP2     163 0.039983000 HEADERS[5]: POST /1.1/jot/client_event.json?keepalive=false
1766 24.014438  192.168.1.14    104.244.42.194  HTTP2    1642 0.000084000 DATA[5] (application/x-www-form-urlencoded)
1767 24.015846  192.168.1.14    104.244.42.194  HTTP2     133 0.000894000 HEADERS[9]: OPTIONS /1.1/attribution/event.json
1768 24.018350  192.168.1.14    104.244.42.194  HTTP2     342 0.002504000 HEADERS[11]: OPTIONS /1.1/account/settings.json?include_mention_filter=true&include_nsfw_user_…
1769 24.033181  104.244.42.194  192.168.1.14    HTTP2      95 0.014831000 HEADERS[5]: 200 OK
1770 24.033181  104.244.42.194  192.168.1.14    HTTP2      95 0.000000000 HEADERS[7]: 200 OK
1786 24.038837  192.168.1.14    104.244.42.194  HTTP2     176 0.000593000 HEADERS[7]: GET /graphql/lFi3xnx0auUUnyG4YwpCNw/GetUserClaims?variables=%7B%7D
1787 24.038900  192.168.1.14    104.244.42.194  HTTP2     165 0.000063000 HEADERS[9]: GET /1.1/hashflags.json
1788 24.039004  104.244.42.194  192.168.1.14    HTTP2      95 0.005780000 HEADERS[9]: 200 OK
1789 24.040112  192.168.1.14    104.244.42.194  HTTP2     151 0.001212000 HEADERS[11]: POST /1.1/attribution/event.json
1791 24.048220  104.244.42.194  192.168.1.14    HTTP2      95 0.009216000 HEADERS[11]: 200 OK
1798 24.060218  192.168.1.14    104.244.42.194  HTTP2     135 0.011967000 HEADERS[13]: OPTIONS /1.1/onboarding/referrer.json
1799 24.062874  192.168.1.14    104.244.42.194  HTTP2     151 0.022695000 HEADERS[13]: POST /1.1/attribution/event.json
1805 24.063264  192.168.1.14    104.244.42.194  HTTP2     355 0.000068000 HEADERS[15]: GET /1.1/account/settings.json?include_mention_filter=true&include_nsfw_user_flag…
```

```
 > Header: :method: OPTIONS
 v Header: :authority: api.twitter.com
     Name Length: 10
     Name: :authority
     Value Length: 15
     Value: api.twitter.com
     :authority: api.twitter.com
     [Unescaped: api.twitter.com]
     Representation: Indexed Header Field
     Index: 75
 v Header: :scheme: https
     Name Length: 7
     Name: :scheme
     Value Length: 5
     Value: https
     :scheme: https
```

```
0000  00 00 00 07 3a 6d 65 74  68 6f 64 00 00 00 07 4f   ····:met hod····O
0010  50 54 49 4f 4e 53 00 00  00 0a 3a 61 75 74 68 6f   PTIONS·· ··:autho
0020  72 69 74 79 00 00 00 0f  61 70 69 2e 74 77 69 74   rity···· api.twit
0030  74 65 72 2e 63 6f 6d 00  00 00 07 3a 73 63 68 65   ter.com· ···:sche
0040  6d 65 00 00 00 05 68 74  74 70 73 00 00 00 05 3a   me····ht tps····:
0050  70 61 74 68 00 00 00 13  2f 31 2e 31 2f 68 61 73   path···· /1.1/has
0060  68 66 6c 61 67 73 2e 6a  73 6f 6e 00 00 00 06 61   hflags.j son····a
0070  63 63 65 70 74 00 00 00  03 2a 2f 2a 00 00 00 1d   ccept··· ·*/*····
0080  61 63 63 65 73 73 2d 63  6f 6e 74 72 6f 6c 2d 72   access-c ontrol-r
0090  65 71 75 65 73 74 2d 6d  65 74 68 6f 64 00 00 00   equest-m ethod···
00a0  03 47 45 54 00 00 00 1e  61 63 63 65 73 73 2d 63   ·GET···· access-c
00b0  6f 6e 74 72 6f 6c 2d 72  65 71 75 65 73 74 2d 68   ontrol-r equest-h
00c0  65 61 64 65 72 73 00 00  00 5e 61 75 74 68 6f 72   eaders·· ·^author
00d0  69 7a 61 74 69 6f 6e 2c  78 2d 63 73 72 66 2d 74   ization, x-csrf-t
00e0  6f 6b 65 6e 2c 78 2d 74  77 69 74 74 65 72 2d 61   oken,x-t witter-a
00f0  63 74 69 76 65 2d 75 73  65 72 2c 78 2d 74 77 69   ctive-us er,x-twi
0100  74 74 65 72 2d 61 75 74  68 2d 74 79 70 65 2c 78   tter-aut h-type,x

Frame (128 bytes)   Decrypted TLS (52 bytes)   Decompressed Header (663 bytes)
```

Fig. 3. Packet content on Wireshark app.

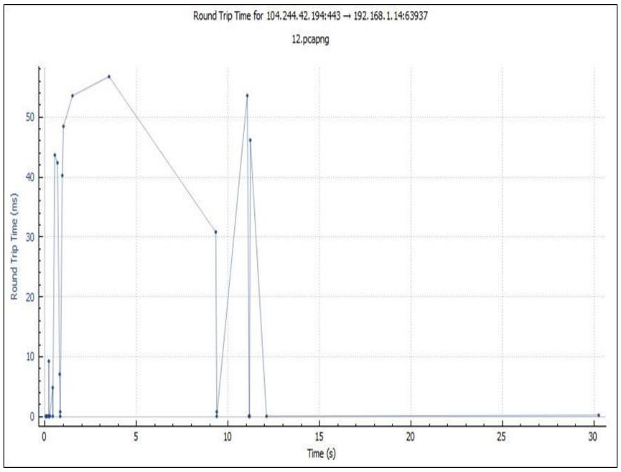
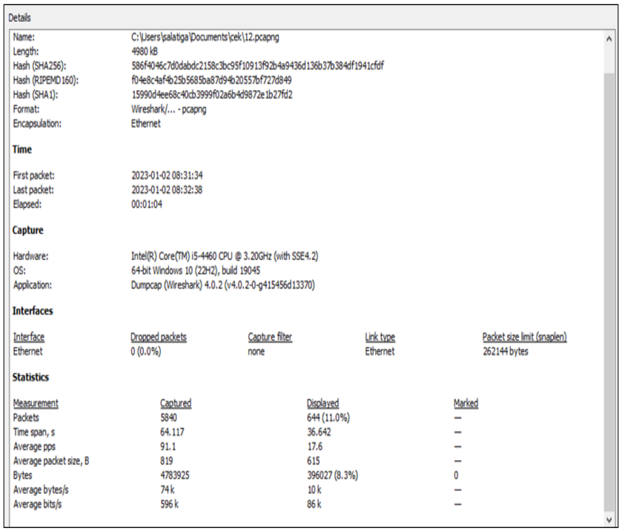


Fig. 4. Twitter server communication round trip time value.



Details

Name:	C:\Users\salatiga\Documents\cek\12.pcapng
Length:	4980 kB
Hash (SHA256):	586f4046c7d0dabdc2158c3bc95f10913f92b4a9436d136b37b384df1941cfdf
Hash (RIPEMD160):	f04e8c4af4b25b5685ba87d94b20557bf727d849
Hash (SHA1):	15990d4ee68c40cb3999f02a6b4d9872e1b27fd2
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2023-01-02 08:31:34
Last packet:	2023-01-02 08:32:38
Elapsed:	00:01:04

Capture

Hardware:	Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz (with SSE4.2)
OS:	64-bit Windows 10 (22H2), build 19045
Application:	Dumpcap (Wireshark) 4.0.2 (v4.0.2-0-g415456d13370)

Interfaces

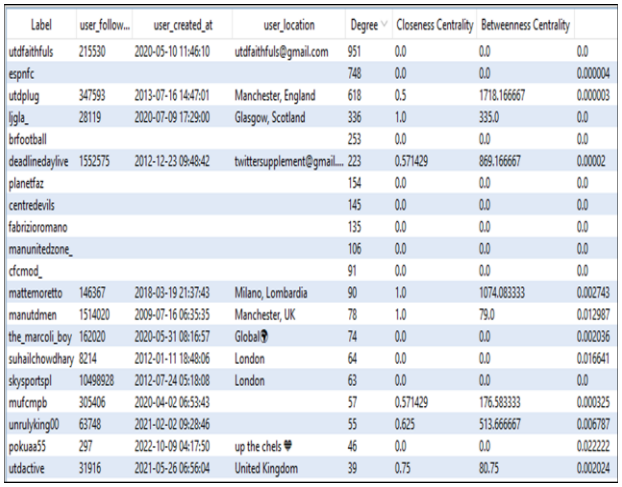
Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Ethernet	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	5840	644 (11.0%)	—
Time span, s	64.117	36.642	—
Average pps	91.1	17.6	—
Average packet size, B	819	615	—
Bytes	4783925	396027 (8.3%)	0
Average bytes/s	74 k	10 k	—
Average bits/s	596 k	86 k	—

Fig. 5. The results of data acquisition and PCAP file integrity check.

Fig. 5 shows data collection verification and file integrity verification in network forensics refers to the process of collecting and verifying the authenticity and integrity of digital evidence from the network. This process is critical to ensuring that evidence collected is accurate and available for investigation or trial. Data collection involves capturing and copying data from the network, while file integrity checking involves checking hashes or digital signatures of captured data to ensure it has not been tampered with. These steps are important to maintain the chain of custody and maintain the authenticity of the evidence. Digital forensic analysis using SNA includes identifying primary and secondary actors. This assessment makes the investigative process more focused on specific actors.



Label	user_follow...	user_created_at	user_location	Degree ⌄	Closeness Centrality	Betweenness Centrality	
utdfaithfuls	215530	2020-05-10 11:46:10	utdfaithfuls@gmail.com	951	0.0	0.0	0.0
espnfc				748	0.0	0.0	0.000004
utdplug	347593	2013-07-16 14:47:01	Manchester, England	618	0.5	1718.166667	0.000003
ljgla_	28119	2020-07-09 17:29:00	Glasgow, Scotland	336	1.0	335.0	0.0
brfootball				253	0.0	0.0	0.0
deadlinedaylive	1552575	2012-12-23 09:48:42	twittersupplement@gmail....	223	0.571429	869.166667	0.00002
planetfaz				154	0.0	0.0	0.0
centredevils				145	0.0	0.0	0.0
fabrizioromano				135	0.0	0.0	0.0
manunitedzone_				106	0.0	0.0	0.0
cfcmod_				91	0.0	0.0	0.0
mattemoretto	146367	2018-03-19 21:37:43	Milano, Lombardia	90	1.0	1074.083333	0.002743
manutdmen	1514020	2009-07-16 06:35:35	Manchester, UK	78	1.0	79.0	0.012987
the_marcoli_boy	162020	2020-05-31 08:16:57	Global🌍	74	0.0	0.0	0.002036
suhailchowdhary	8214	2012-01-11 18:48:06	London	64	0.0	0.0	0.016641
skysportspl	10498928	2012-07-24 05:18:08	London	63	0.0	0.0	0.0
mufcmpb	305406	2020-04-02 06:53:43		57	0.571429	176.583333	0.000325
unrulyking00	63748	2021-02-02 09:28:46		55	0.625	513.666667	0.006787
pokuaa55	297	2022-10-09 04:17:50	up the chels 💙	46	0.0	0.0	0.022222
utdactive	31916	2021-05-26 06:56:04	United Kingdom	39	0.75	80.75	0.002024

Fig. 6. Degree, closeness and betweeness Centrality.

Fig. 6 generates degree centrality as the number of connections or edges a node has to other nodes in the network. Nodes with high centrality have many connections to other nodes. In this study, 10,000 nodes were obtained using the manchester united keyword. The main actor has 951 degrees on the node labeled: utdfaithfuls. Perform key player identification to determine the degree of influence a participant has in the network. It is important to identify who are the actors who play the most important role in the communication model. This betweenness centrality takes into account the number of shortest paths between other nodes passing through a given node. High centrality nodes are "bridges" between other nodes in the network. The degree, proximity, and betweenness centrality metrics tables will show the following information for each node (or vertex) in the network Degree centrality, this is a measure of how many direct connections a node has to other nodes in the network. This is usually expressed as the number of edges intersecting the node. Closeness centrality, this is a measure of how close a node is to all other nodes in the network. This is usually expressed as the sum of the shortest distances between a node and all other nodes in the network. Betweenness centrality, this is a measure of how often a node acts as a bridge between other nodes in the network. This is usually expressed as the number of shortest paths between pairs of nodes passing through a given node.

For this experiment, the tag **utdfaithfuls** Beetweenness centrality: 1718 nodes with utdplug. Graphical visualization of the top three communication modes at the highest level is shown in Fig. 7. Graph visualization is a commonly used technique in social network analysis to show the relationships between individuals or entities in the network. It visualizes the network as a graph or graph, with nodes representing individuals or entities and edges representing the relationships between them. The size and color of nodes and the thickness and direction of edges can be used to represent attributes or measures, such as relationship strength or centrality measures. This helps visualize patterns, relationships, and communities in the network and makes it easier to understand the network structure and its properties.

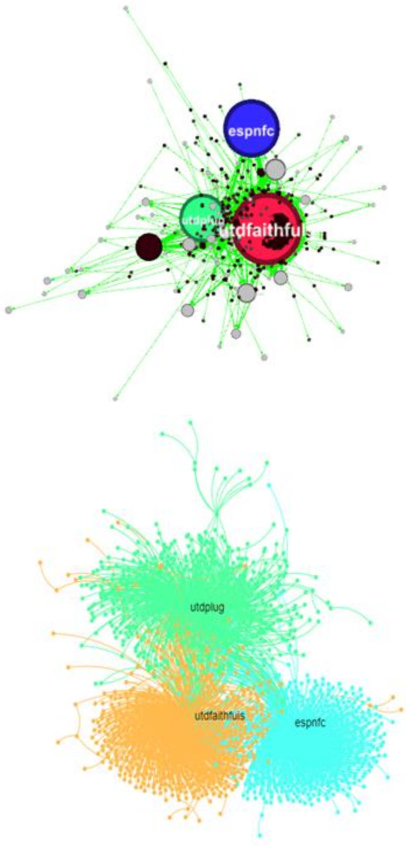


Fig. 7. Graph visualization of the top 3 pad communication patterns with the highest degree.

TABLE I. (A). RESULTS OF NETWORK FORENSIC ANALYSIS OF PCAP. AND SNA COMBINATION

Source IP	Destination IP	Packet	A >B	B>A	Date/Time	Dura Tion (s)	Name server
192.168.1.16	224.0.0.251	18	18	0	2023-01-02T08:31:34.517	4,74	
192.168.1.5	239.255.255.250	12	12	0	2023-01-02T08:31:37.513	60,00	
192.168.1.14	20.198.119.143	3	2	1	2023-01-02T08:31:40.793	0,12	
192.168.1.10	255.255.255.255	1	1	0	2023-01-02T08:31:54.895	-	
192.168.1.14	104.244.42.129	129	35	94	2023-01-02T08:31:56.481	7,37	
192.168.1.14	152.199.43.83	26	12	14	2023-01-02T08:31:57.878	0,69	
192.168.1.14	104.244.42.194	644	255	389	2023-01-02T08:31:58.043	36,60	api.twitter.com
192.168.1.14	104.244.42.5	11	5	6	2023-01-02T08:31:58.107	0,07	api.twitter.com
192.168.1.14	104.244.43.131	42	19	23	2023-01-02T08:31:59.670	11,45	api.twitter.com
192.168.1.14	192.168.1.255	1	1	0	2023-01-02T08:32:24.230	-	
192.168.1.14	239.255.255.250	4	4	0	2023-01-02T08:32:32.322	3,03	
192.168.1.13	192.168.1.14	16	16	0	2023-01-02T08:32:32.326	3,23	

(B). RESULTS OF NETWORK FORENSIC ANALYSIS OF SNA

Social Network Analysis	1.	Network Size	10000 node
	2.	Centrality / Keyactor	UtdFaithfuls
		a. User created at	10/05/20 11.46
		b. Follower	215530
		c. Pub date	04/01/23 16.54
	3	Clustering coefficient	0.005
	4	Density	0.001
	5	Network Diameter	71
	6	Cluster	3

As Table I (A), (B) in network forensics, combining PCAP analysis with social network analysis can have a big impact on research in a number of ways:

1) Better suspect identification. Based on their connections to and interactions with other network members, prospective suspects in a network can be found via social network analysis. This can give a more complete picture of the network and increase the precision with which suspects are identified.

2) An improved comprehension of network behavior can be obtained by combining social network analysis and PCAP analysis, which can give a more in-depth understanding of communication patterns and network behavior. Investigators can use this to spot unusual or suspect activities on the network, such as secret or encrypted communication.

3) Enhanced incident response. By integrating PCAP analysis with social network analysis, detectives can more precisely pinpoint the origin and consequences of a crime.

The outcomes of benchmarking the integration of PCAP analysis with social network analysis in network forensics can give important insights into the most efficient ways to carry out investigations, uncover potential security concerns, and improve network security generally.

V. Conclusion

Network forensics using PCAP analysis combined with social network analysis shows more comprehensive results. PCAP analysis is used to analyze network traffic, conversation statistics, protocol distribution, packet content and round-trip time. Social network analysis maps communication patterns between nodes and identifies the most influential key players in the network. PCAP analysis efficiently captures and analyzes network packets, while SNA provides insight into the relationships and communication patterns between devices on the network. The availability of data sources is also a factor to consider when deciding to combine PCAP and SNA. The combination of PCAP and SNA can provide a more complete view of the network when limited log data is available. When the nature of the threat includes both technical and social aspects, a combination of these two approaches may be required. PCAP and SNA can provide a more comprehensive view of the network and improve the accuracy of forensic analysis. This is because SNA can provide context about captured packets and help identify relationships and patterns that might not be apparent from packet analysis alone. Potential future suggestions for improving forensic cyber analysis include artificial intelligence and machine learning, which includes AI and machine learning algorithms that can help automate the data analysis process, making it more efficient and accurate.

References

[1] L. F. Sikos, “Packet analysis for network forensics: A comprehensive survey,” Forensic Sci. Int. Digit. Investig., vol. 32, p. 200892, Mar. 2020, doi: 10.1016/J.FSIDI.2019.200892.

[2] N. Koroniotis, “Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,” Futur. Gener. Comput. Syst., vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.

[3] J. Lian, “Implementation of computer network user behavior forensic analysis system based on speech data system log,” Int. J. Speech Technol., vol. 23, no. 3, 2020, doi: 10.1007/s10772-020-09747-2.

[4] H. Munkhondya, “Digital forensic readiness approach for potential evidence preservation in software-defined networks,” 14th Int. Conf. Cyber Warf. Secur. ICCWS 2019, pp. 268–276, 2019.

[5] B. C. M. Cappers, “Eventpad: Rapid Malware Analysis and Reverse Engineering using Visual Analytics,” 2018 IEEE Symposium on Visualization for Cyber Security, VizSec 2018. 2019. doi: 10.1109/VIZSEC.2018.8709230.

[6] I. Yaqoob, “Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges,” Futur. Gener. Comput. Syst., vol. 92, pp. 265–275, 2019, doi: 10.1016/j.future.2018.09.058.

[7] A. Ulmer, “NetCapVis: Web-based progressive visual analytics for network packet captures,” 2019 IEEE Symposium on Visualization for Cyber Security, VizSec 2019. 2019. doi: 10.1109/VizSec48167.2019.9161633.

[8] C. Yin, H. Wang, and J. Wang, “Network data stream classification by deep packet inspection and machine learning,” in Lecture Notes in Electrical Engineering, 2019, vol. 518. doi: 10.1007/978-981-13-1328-8_31.

[9] B. Indira, K. Valarmathi, and D. Devaraj, “An approach to enhance packet classification performance of software-defined network using deep learning,” Soft Comput., vol. 23, no. 18, 2019, doi: 10.1007/s00500-019-03975-8.

[10] J. Yoon and M. DeBiase, “Real-time analysis of big network packet streams by learning the likelihood of trusted sequences,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, vol. 10968 LNCS. doi: 10.1007/978-3-319-94301-5_4.

[11] R. K. Shrivastava, “Attack detection and forensics using honeypot in IoT environment,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2019, vol. 11319, pp. 402–409. doi: 10.1007/978-3-030-05366-6_33.

[12] K. Chakraborty, S. Bhattacharyya, and R. Bag, “A Survey of Sentiment Analysis from Social Media Data,” IEEE Trans. Comput. Soc. Syst., vol. 7, no. 2, pp. 450–464, 2020, doi: 10.1109/TCSS.2019.2956957.

[13] B. Liu, “Large-scale group decision making model based on social network analysis: Trust relationship-based conflict detection and elimination,” Eur. J. Oper. Res., vol. 275, no. 2, pp. 737–754, 2019, doi: 10.1016/j.ejor.2018.11.075.

[14] F. L. Aryeh, “Graphical analysis of captured network packets for detection of suspicious network nodes,” 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cyber SA 2020. 2020. doi: 10.1109/CyberSA49311.2020.9139672.

[15] S. M. Farjad, “Cluster Analysis and Statistical Modeling: A Unified Approach for Packet Inspection,” 1st Annual International Conference on Cyber Warfare and Security, ICCWS 2020 - Proceedings. 2020. doi: 10.1109/ICCWS48432.2020.9292396.

[16] M. Marchetti, “READ: Reverse engineering of automotive data frames,” IEEE Trans. Inf. Forensics Secur., vol. 14, no. 4, pp. 1083–1097, 2019, doi: 10.1109/TIFS.2018.2870826.

[17] S. M. Hosseini, “Digesting Network Traffic for Forensic Investigation Using Digital Signal Processing Techniques,” IEEE Trans. Inf. Forensics Secur., vol. 14, no. 12, pp. 3312–3321, 2019, doi: 10.1109/TIFS.2019.2915190.

[18] P. Białczak, “Hfinger: Malware http request fingerprinting,” Entropy, vol. 23, no. 5, 2021, doi: 10.3390/e23050507.

[19] S. Ali, “Effective Multitask Deep Learning for IoT Malware Detection and Identification Using Behavioral Traffic Analysis,” IEEE Trans. Netw. Serv. Manag., 2022, doi: 10.1109/TNSM.2022.3200741.

[20] S. Sen Zhang, X. Liang, YD Wei, and X. Zhang, “On Structural Features, User Social Behavior, and Kinship Discrimination in Communication Social Networks,” IEEE Trans. Comput. soc. syst., vol. 7, no. 2, pp. 425–436, 2020, doi:10.109/TCSS.2019.2962231.

[21] A. Matakos, C. Aslay, E. Galbrun, and A. Gionis, “Maximizing the Diversity of Exposure in a Social Network,” IEEE Trans. knowl. Data

Eng., vol. 34, no. 9, pp. 4357–4370, 2022, doi:10.109/TKDE.2020.3038711.

[22] M. Mirtaheri, S. Abu-El-Haija, F. Morstatter, G. Ver Steeg, and A. Galstyan, “Identifying and Analyzing Cryptocurrency Manipulations in Social Media,” IEEE Trans. Comput. soc. syst., vol. 8, no. 3, pp. 607–617, 2021, doi:101109/TCSS.2021.3059286.

[23] D. Vimalajeewa, S. Balasubramaniam, B. O'Brien, C. Kulatunga, and DP Berry, “Leveraging Social Network Analysis for Characterizing Cohesion of Human-Managed Animals,” IEEE Trans. Comput. soc. syst., vol. 6, no. 2, pp. 323–337, 2019, doi:10.109/TCSS.2019.2902456.

[24] AA Al-Shargabi and A. Selmi, “Social Network Analysis and Visualization of Arabic Tweets During the COVID-19 Pandemic,” IEEE Access, vol. 9, pp. 90616–90630, 2021, doi:101109/access.2 021.3091537.

Validating the Usability Evaluation Model for Hearing Impaired Mobile Application

Shelena Soosay Nathan[1], Nor Laily Hashim[2], Azham Hussain[3], Ashok Sivaji[4], Mohd Affendi Ahmad Pozin[5]
ITecH Focus Group-Center for Diploma Studies, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia[1]
School of Computing, Universiti Utara Malaysia, Kedah, Malaysia[2, 3]
Malaysia Institute of Microelectronics Systems (MIMOS), Kuala Lumpur, Malaysia[4]
Faculty of Business and Communication, Universiti Malaysia Perlis, Perlis, Malaysia[5]

***Abstract*—Usability is an important element that enables the identification of the efficiency for application or product. However, many applications have been developed for general users' needs and are unable to provide adequate applications usage for disabled people. This study focuses on the development of usability evaluation model and the validation process on the proposed model through experts. The developed model later evaluated by group of experts through focus group method. Focus group method enables to identify the 13 variables derived to develop the model are appropriately placed and useful in the evaluation process. The results shows that the selected variables are appropriate to identify usability of mobile application for the hearing impairment through three variables tested namely, gain satisfaction with the model, satisfaction with the model presentation, and support for tasks. Conclusively, the developed model can identify usability of mobile applications for hearing impairment and enable in identifying useful criteria to be included during application development process in real life process. As future study, the model can be tested among the hearing impairment people and practitioner to establish the results obtained which contributes to usability practitioners and application developers for the disabled.**

***Keywords*—Usability; hearing impaired; validation; evaluation; MAEHI**

I. Introduction

Usability is an important element in any system or application to analyse any incurring usability issues. Usability commonly referred as a guideline for measuring usability of the system using models. These models provide insight into measurements to be used for usability analysis data collection [1]. Models such as International Standard Organization (ISO) 9241-11 [2] and Nielsen's (2005) [3] model are among the common usability evaluation models that will be used in user evaluation to identify the issues [1]. However, when a specific targeted user application is developed, the user's requirement must be incorporated into the application. If the requirement is missing, the application will fail to satisfy the user and becomes difficult to use.

Different disabilities having different levels of cognitive and mental strength [4][5][6]. Thus, applications developed for these people should consider these issues to ensure the usefulness of such application. Mobile devices are proliferating at an incredible rate. Statista (2022) [7] releases that currently, smartphone users have increased up to seven billion worldwide whereas almost 80% of the population in the world is owning a smartphone. These rates are extraordinary, given to relative introduction of many applications in mobile that eases people's daily life.

As such, this mobile application industry is growing remarkably, and the penetration of mobile apps can be seen growing in foreseeable future. Mobile phones once have only been used for answering calls now have grown up for muti use among the users. Many have gained benefits from various use of the mobile phone [1]. As this is the case, mobile phone does not bound only for the normal people but benefits the disabled people around the world as well. The compatibility of these mobile phones and its applications are being studied continuously as an objective to enhance the usability of all type of people as said in [1] to be more useable.

Usability has been a pivotal part of the discussion in many domains [8][9]. Usability evaluation is being conducted to measure an application satisfaction achieved by the users and commonly measured subjectively which is a collection of satisfaction rating of application among users [10]. However, this method is less defined [10]. Few renowned usability models are used as references to measure the applications such as ISO 9241-11 (1998) and Nielsen's (2005) [3] model although many other usability models have been constantly studied thereafter.

The focused issue is about the application developed for special people which is rather challenging compare with applications for normal people. Measurements used for general purpose application are unable to measure important features of the needs of these disabled users [1][11].

According to a survey conducted in the United States, the number of hearing-impaired users using smartphone applications is in the second highest percentage (i.e. 31%) after physically disabled people [12]. This shows that the hearing-impaired are one of the major disabled users of the smartphone and its applications. The hearing-impaired are those whose hearing is impaired to some degree at the time of birth [13]. These people are unable to produce coherent speech due to the lack of auditory input and thus the inability to monitor their voices [1] [14]. They rely on sign languages or any methods with gestures for communication purposes [15] [1].

As such, a study was conducted and a usability assessment model for the hearing - impaired, called Model for Mobile

Application Evaluation for Hearing Impaired (MAEHI), has been proposed to overcome these deficiencies [1]. This model proposed consists of six dimensions, 15 criteria and 47 metrics. This study will conduct the expert review methods for the developed MAEHI model for verification process to ensure the reliability of the proposed MAEHI model that has been developed.

The verification of the model performed through the method of expert review. The expert review was conducted to verify the proposed MAEHI for usability assessment of mobile applications with hearing impairment. One of the important ways to detect and remove defects is by expert review [16]. This study therefore adopted this technique to verify the proposed MAEHI. All the components developed, and the appropriate organization and presentation can be confirmed by verification [17].

This paper consists of the verification process of the model that was conducted to ensure that the model needs to be met for the deaf user. As such, the next section consists of discussing the focus group method for the assessment of the model followed by section three discussing the results of the focus group and section four discussing the conclusion.

II. Methods

The main aspect to be verified in the proposed MAEHI is the use of the appropriate dimension, criteria and metrics and the overall applicability, originality, and comprehensibility of the proposed dimension in MAEHI model [1] as shown in the Fig. 1, which is categorized into five components: consistency, ease of use, comprehensible, verifiable, and overall impression.

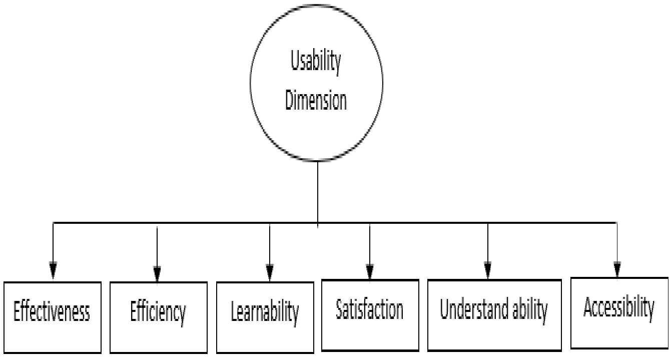


Fig. 1. MAEHI model dimension [1].

Among the 15 dimensions that were shortlisted, only six were chosen for final model development of MAEHI [1]. These criteria with the dimensions are designed in the model form as full model version and presented in [1].

Potential usability experts were therefore identified for the validation process of the model in Fig. 1, particularly researchers, academics, application developers or practitioners, as well as disability experts, particularly on the hearing impaired. As suggested by Hallowell, and John Gambatese (2009) [18] and Rogers, Margaret, and Emilia Lopez (2002) [19], the expert was selected. Once the model was verified, the validation was carried out to ensure that the model developed can provide adequate results in evaluating the deaf application's usability.

Domain users and experts from Malaysia National Research and Development, User Experience Lab, Malaysian Institute of Microelectronic Systems (MIMOS) participated in the MAEHI evaluation. This is a well-known software development organisation with numerous technology-focused areas and the only usability laboratory certified in Malaysia with MS ISO / IEC17025:2005 [20]. As part of this study, the model evaluation with experts and usability testing was conducted in cooperation with MIMOS. Experts validated the MAEHI in terms of its ability to be used in real testing environment through the focus group discussion.

The next section is implementing the focus group activity which includes planning, conducting, and analysing the outcome [21][22]. The objective of the focus group is to evaluate and validate the MAEHI in terms of its ability in producing reliable results to be implemented in real-world environments. As such, conduct planning is made up of a few steps as it is important to ensure that the focus group is properly conducted.

Planning was carried out thoroughly to ensure a good implementation of the focus group. There are four activities for the focus group implementation, adapted from Mazza, Riccardo, and Berre [23] to define focus group objectives, identify participants, prepare materials, and schedule meetings. Each activity is further explained in the section below.

A. *Defining Objective of Focus Group*

In particular, the focus group's objective is to evaluate and validate the MAEHI in terms of its ability in producing reliable results to be implemented in real-world environments.

B. *Identification of Participants Defining Objective of Focus Group*

At the MIMOS National Research and Development Center, Kuala Lumpur, participants for the focus group were selected from a group of application developers and usability practitioners. This study, however, only considers practitioners with more than three years of experience in this field [24]. Through emails and phone calls, 11 software developers and usability practitioners were approached. However, only seven were willing to participate as the others were involved in the organization's projects. The seven participants are within the recommended range for a focus group [25] (i.e., between six to ten members). The focus group was organised for the convenience of the expert group at MIMOS User Experience Lab in Kuala Lumpur.

C. *Meeting Scheduling for Focus Group*

The meeting of the focus group was scheduled at the Kuala Lumpur MIMOS User Experience Lab. The meeting place was chosen within the organization itself to accommodate its busy schedule with organizational projects. Since the venue was within the organization itself, it is convenient for the participants [26].

D. Material Preparation for Focus Group

Based on the activity to be carried out, the material used for the focus group was prepared. Materials such as participant documentation for evaluation have been prepared earlier to ease the focus group process.

The focus group was conducted with the developer's group and usability practitioners as scheduled. Upon arrival, all the participants were warmly and informally welcomed, as some arrived earlier than the scheduled time. This creates a space for knowing much better about the field in which they work and makes them feel relaxed before the formal discussion begins. Next, the materials needed for the session of the expert focus group were given. A brief introduction to the focus group objective was mentioned and the focus group purpose was reminded to experts. They were briefed on the model for validation purposes to be evaluated. The explanation given was quite easy for them as they were from the usability domain. It took nearly two hours to discuss the entire model assessment process, which is a common and acceptable duration [23][24].

All participants tested the developed MAEHI in the mobile application they downloaded from the app store on their smartphones during the evaluation process. Experts filled out the evaluation form to indicate their satisfaction with the ability of the developed model to be implemented in usability measurements in the real-world environment.

E. Instrument Development

The experts validated the developed model through an evaluation with predefined criteria adapted from past studies [27]. These criteria include gain satisfaction with the model, satisfaction with the Model presentation, and support for tasks. In the first criterion that is gain satisfaction with the model variables were 'Relevance for the intended application,' 'Perceived usefulness,' and 'Clarity'. This criterion is to reveal a model or framework's accuracy. The second criterion that is satisfaction with the Model presentation is represented by two variables that are "Ease of Use" and "Organization". This allows the experts to determine whether the model developed is practical and easy to evaluate and well organized for usability [28].

In the third criterion, four variables are used to measure task support satisfaction, where they were tested on "Practicality," "Completeness," "Incomprehensibility" and "Ability to produce expected results." These variables allow the model's understandable relevance to imply an evaluation and completeness of the application [29]. Each criterion will be assessed based on its own variables with two scales in, "Agree" and "Disagree."

F. Data Analysis

Once the evaluation is completed by the experts, the forms are collected, and its data is analysed by obtaining the ratio value for each item and an overall proportion of the measures to identify the experts' level of agreement on the model. The model evaluation findings will be further discussed in the next section.

III. Results and Discussion

After testing the developed model and discussing its ability to be used in the real-world environment, The results of these measures were the endorsement in the real-world environment of the validity of the proposed model. Consequently, each criterion and the results will be discussed as below.

Through this, the model has been validated by the experts in executing for usability assessment use on its relevance. Table I shows the experts' feedback.

TABLE I. Expert Agreement on Gain Satisfaction with the Model

Variables	%	Results
Relevancy to the intended application	86%	Experts agreed that the model developed is relevant since it focuses on targeted users for accessibility. They also found that the developed model appropriately highlighted accessibility.
Perceived usefulness	86%	Experts agreed that the usability model developed is suitable for the evaluation of hearing-impaired mobile apps. Furthermore, all experts agreed that this usability assessment made it possible to identify problems for hearing impaired users as the existing usability models are not able to meet their needs.
Clarity	100%	All the experts agreed unanimously that the phases provided for the developed model were clear in all the evaluations and tasks. The method provided for both data calculation and collection were agreed for use as clear.

The results of the expert evaluation validation were shown in Table I for their satisfaction gain with the model. Based on the experts' view, the developed model is identified as being useful for evaluating the usability of the mobile application with hearing impairment [30]. In addition, expert feedback also revealed that the model developed is clear for use with real users and can identify usability - related issues especially for hearing impaired users.

The expert feedback based on the, Satisfaction with the Model presentation is outlined in Table II.

TABLE II. Expert Agreement on Satisfaction with the Model

Variables	%	Results
Perceived Ease of use	86%	Experts agreed on the model's ease of use for the intended application. However, one expert was not convinced that this evaluation could be carried out on the hearing impaired as they would face communication problems during testing as they would need a translator if they communicated with normal hearing people.
Organization	86%	Experts agreed that the model developed was well organized and satisfied with the way all the procedures to be evaluated were organized in terms of their structure sequence and understandability arrangement.

According to Table II, recorded by the experts' feedback, they found that the usability model was easy to implement in the real environment, apart from well-organized measurements. The experts were pleased to find the model suitable for the evaluation of hearing-impaired mobile apps. There was, however, a slight reluctance to collect data with the hearing impaired as communication barrier is an important issue for usability practitioners to understand [31].

Table III shows that the model is practical for the hearing-impaired usability assessment to be conducted. The model is also comprehensive and easy to understand and implement. In addition, the results show agreeable views that the model developed can produce adequate results.

TABLE III. EXPERT AGREEMENT ON TASK SUPPORT SATISFACTION OF THE MODEL

Variables	%	Results
Practicality	100%	Experts showed some reluctance to practice with disabled people on the real environment. They agreed, however, if the evaluation was carried out using a controlled method, in which it would be possible. The other experts mentioned that the metrics and criteria were suitable and that expected results were easy to obtain.
Completeness	86%	Experts found the model to be appropriate in the use of usability assessment for hearing impaired mobile applications.
Understandability	100%	The proposed model was viewed in an understandable way as all the experts agreed that it was a well - organized evaluation and that it was easy to understand, especially for those with usability experience.
		All the experts agreed on the proposed model that it could produce expected results. They expressed satisfaction with the conduct of the evaluation and the dimensions covering the requirements of users with hearing impairment. In addition, experts also agreed that the model could produce expected results and identified usability issues in the evaluation of the application.

A summary of all the percentages obtained and calculated on average identifies the overall score for each criterion. For all variables, the overall percentage of agreement for each criterion is above 86%, indicating a high percentage of agreement on the model and acceptance of the model's applicability in the real-world environment. Thus, these results show that the proposed model is practical for the usability assessment of hearing-impaired mobile applications to be implemented in the real-world environment.

IV. CONCLUSION

Experts delightfully wanted to participate with the actual hearing-impaired users in the usability test to further evaluate the model. This also demonstrates the experts' eagerness to directly test the model developed in the real environment. They also shared the intention to collaborate with the hearing-impaired with this developed model in future testing and usability evaluation as they found that the model is particularly useful when the user is disabled. Emphasis has been given to the fact that usability assessment model is less important for disabled user applications, which many developers tend to ignore. The developed model will be used for usability testing for future study to demonstrate the practicality of use in the real environment and to identify the model's ability to collect useful analytical data.

ACKNOWLEDGMENT

Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216. The author would also like to thank MIMOS Berhad for their utmost cooperation.

REFERENCES

[1] S. Nathan, A. Hussain, and N. L. Hashim, "Usability evaluation of DEAF mobile application interface: A systematic review," *Journal of Engineering and Applied Sciences*, vol. 13, no. 2, pp. 291–297, 2018, doi: 10.3923/jeasci.2018.291.297.

[2] International Organization for Standardization, *International Standard: ISO 9241-11(Guidance on Usability)*. Geneva., 1998.

[3] J. Nielsen, "Durability of usability guidelines," 2005.

[4] A. Kobayashi, K. Yasu, H. Nishizaki, and N. Kitaoka, "Corpus Design and Automatic Speech Recognition for Deaf and Hard-of-Hearing People," in *2021 IEEE 10th Global Conference on Consumer Electronics, GCCE 2021*, 2021, pp. 17–18. doi: 10.1109/GCCE53005.2021.9621959.

[5] K. Mack, D. Bragg, M. R. Morris, M. W. Bos, I. Albi, and A. Monroy-Hernández, "Social App Accessibility for Deaf Signers," *Proc ACM Hum Comput Interact*, vol. 4, no. CSCW2, Oct. 2020, doi: 10.1145/3415196.

[6] J. P. Bigham and P. Carrington, "Learning from the Front: People with Disabilities as Early Adopters of AI." [Online]. Available: https://www.bloomberg.com/news/articles/2018-03-19/uber-crash-is-nightmare-the-driverless-world-feared-but-expected.

[7] Statista, "Smartphones - Statistics & Facts.," 2022.

[8] A. Riegler and C. Holzmann, "Measuring Visual User Interface Complexity of Mobile Applications with Metrics," *Interact Comput*, vol. 30, no. 3, pp. 207–223, May 2018, doi: 10.1093/iwc/iwy008.

[9] C. K. Coursaris and D. J. Kim, "A Meta-Analytical Review of Empirical Mobile Usability Studies," 2011.

[10] A. Abran, "A Measurement Design for the Comparison of Expert Usability Evaluation and Mobile App User Reviews Software Estimation View project A Measurement Design for the Comparison of Expert Usability Evaluation and Mobile App User Reviews View project Necmiye Genc-Nayebi École de Technologie Supérieure," 2018. [Online]. Available: https://www.researchgate.net/publication /327797105.

[11] R. Inostroza, C. Rusu, S. Roncagliolo, and V. Rusu, "Usability heuristics for touchscreen-based mobile devices: Update," in *ACM International Conference Proceeding Series*, 2013, pp. 24–29. doi: 10.1145/2535597.2535602.

[12] J. Poushter and R. Stewart, "Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies But advanced economies still have higher rates of technology use FOR MEDIA OR OTHER INQUIRIES," 2016. [Online]. Available: www.pewresearch.org.

[13] "10 facts about deafness," 2022.

[14] V. Nagalingam, "Communicative Themes and Features in SMS Messages of the Deaf ," University of Malaya., 2008.

[15] S. Ramachandran, Gururaj, K. N. Pallavi, and N. Rajan, “Text to Braille Converting Communication Device forthe Visual and Hearing Impaired Persons,” in *2021 International Conference on Computer Communication and Informatics, ICCCI 2021*, Jan. 2021. doi: 10.1109/ICCCI50826.2021.9402590.

[16] M. Komuro and N. Komoda, “An explanation model for quality improvement effect of peer reviews.,” in *International Conference on Computational Intelligence for Modelling Control & Automation.*, 2008, pp. 1159-1164.

[17] M. Chemuturi, Software Design: A Comprehensive Guide to Software Development Projects. , 1st ed. Chapman and Hall/CRC., 2018.

[18] M. R. Hallowell and J. A. Gambatese, “Qualitative Research: Application of the Delphi Method to CEM Research”, doi: 10.1061/ASCECO.1943-7862.0000137.

[19] M. R. Rogers and E. C. Lopez, “Identifying Critical Cross-Cultural School Psychology Competencies,” 2002.

[20] A. Sivaji *et al.*, “Measuring public value UX-based on ISO/IEC 25010 quality attributes: Case study on e-Government website,” in *Proceedings - 2014 3rd International Conference on User Science and Engineering: Experience. Engineer. Engage, i-USEr 2014*, Jan. 2015, pp. 56–61. doi: 10.1109/IUSER.2014.7002677.

[21] A. Martakis and M. Daneva, “Handling requirements dependencies in agile projects: A focus group with agile software development practitioners,” in *Proceedings - International Conference on Research Challenges in Information Science*, 2013. doi: 10.1109/RCIS.2013.6577679.

[22] M. Daneva and N. Ahituv, “What practitioners think of inter-organizational erp requirements engineering practices: Focus group results,” *International Journal of Information System Modeling and Design*, vol. 2, no. 3, pp. 49–74, Jul. 2011, doi: 10.4018/jismd.2011070103.

[23] R. Mazza and A. Berrè, “Focus Group Methodology for Evaluating Information Visualization Techniques and Tools,” 2007.

[24] P. Liamputtong, *Focus Group Methodology: Principles and Practice*. SAGE Publications Ltd, 2011.

[25] A. F. Newell, P. Gregor, M. Morgan, G. Pullin, and C. Macaulay, “User-Sensitive Inclusive Design,” *Univers Access Inf Soc*, vol. 10, no. 3, pp. 235–243, Aug. 2011, doi: 10.1007/s10209-010-0203-y.

[26] J. Lazar, J. Feng, and H. Hochheiser, *Research Methods in Human-Computer Interaction*, 2nd ed. John Wiley & Sons. , 2017.

[27] H. I. Abubakar, L. Hashim, and A. Hussain, “Usability Evaluation Model for Mobile Banking Applications Interface: Model Evaluation Process using Experts’ Panel”.

[28] S. Farvin and P. Mohamed, “A PROCESS BASED APPROACH SOFTWARE CERTIFICATION MODEL FOR AGILE AND SECURE ENVIRONMENT,” 2015.

[29] P. Bharati and D. Berg, “People and Information Matter: Task Support Satisfaction from the Other Side,” Winter, 2002. [Online]. Available: https://www.researchgate.net/publication/254693791.

[30] A. Hussain and M. Kutar, “Usability Evaluation of SatNav Application on Mobile Phone Using mGQM.” [Online]. Available: www.mirlabs.net/ijcisim/index.html.

[31] N. Mohamad and N. Laily Hashim, “UX Testing for Mobile Learning Applications of Deaf Children.” [Online]. Available: www.ijacsa.thesai.org.

Online Signature Verification for Forgery Detection

Muhammad Rizwan[1], Farhan Aadil[2], Mehr Yahya Durrani[3], Rajermani Thinakaran[4]
Department of Computer Science, COMSATS University Islamabad, Attock Campus, Punjab, Pakistan[1, 2, 3]
Faculty of Data Science and Information Technology, INTI International University, Negeri Sembilan, Malaysia[4]

***Abstract*—The increasing trend of using e-versions of document transmission and storage requires the electronic verification of sender/author. This research presents an efficient and robust online handwritten signature verification system targeting verification rates better than the available state-of-the-art systems in the presence of skilled forgeries. Fourier analysis is employed on the signatures to represent feature vectors in higher dimensional space followed by Local Fisher Discriminant Analysis to obtain compress representation while enhancing inter-class scatter between signature patterns. Signature modeling is performed using m-mediod-based modeling approach where m-mediods are put on to represent data distribution in each class. Connected component labeling is applied to binarized images of Urdu text to extract ligatures which are separated into primary ligatures and diacritics. Fast Euclidean Distance is used as dis(similarity). A total of 2414 signature samples including skilled forgeries are considered in our study. The evaluation of the proposed system on Japanese signature dataset provided by SigWiComp2013 realized promising results than the competitors.**

Keywords*—Fast Euclidean distance; m-mediod; local fisher discriminant analysis

I. Introduction

A person’s authentication is much more demanding in the current era and requires more secure methods to serve the purpose. From the last few decades, there has been an increase in research interest aiming at the development of robust online handwritten signature verification systems. Biometrics can be defined as a process to identify an individual through some characteristics unique to him. Literature categorizes biometric traits into physiological traits and behavioral traits. Physiological methods include facial patterns, fingerprints, iris, hand geometry and retina. While behavioral verification employs traits like handwritten samples (signature, handwriting etc.) and the voice of a person.

Normally, the process of signing is unique to every individual. Two persons with the same name may present different signatures and signing process (inter-personal variation). On the other hand, depending upon the environment or conditions (physical health, fatigue, signing instrument/surface) during the signing process, the same individual’s sign may differ in some aspects (intra-personal variations).

Two approaches are widely researched in literature, which are offline/static signature verification systems and online/dynamic signature verification systems [1]. Offline verification systems [2] bank upon extracting features such as size, shape, signing time and rotation angle etc. from scanned signatures. The actual signing process is performed at paper using a pen and then transformed into digital form by scanning it into computer. Online handwritten signature verification (OHSV) systems [3], on the other hand, record dynamic signature properties such as pen pressure, angle, pen up/down time etc. along the entire signature length.

Online Signature Verification [4, 5] catches the attention of researchers from the past few decades and still is an enduring research area. An online signature is made up of a series of sample points. The features of an online signature can be represented as time series data. Time series is a series of values which are measured as a function of time [6]. Researchers have researched and proposed a variety of procedures and methodologies for the evolution of highly robust online signature verification systems to date including Template Matching Approaches, Structural Approaches and Statistical Approaches. Time series of two signatures compared with Euclidean Distance (ED) -based dynamic time warping (DTW) is shown in Fig. 1. Usually, time series data exhibits higher dimensionality which is sometime difficult to incorporate in its original form. There exists in variety of approaches that transform higher dimensional data to lower dimension and speed up the upcoming processes. These include Fourier Transform [7, 8], Discrete Wavelet Transform (DWT) [9, 10, 11], and Discrete Cosine Transform (DCT) [12, 13]. Using DWT in feature extraction from handwritten digital signatures yielding superior verification rate in comparison to time domain verification system is found in [14].

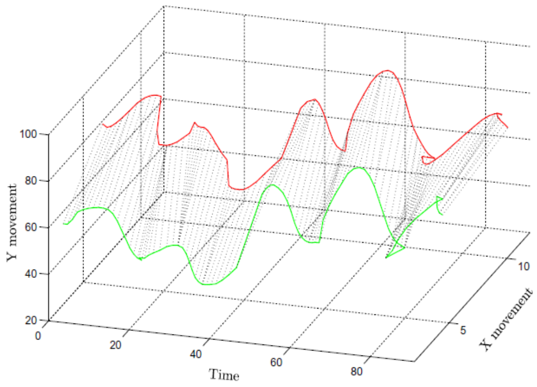


Fig. 1. Two signatures compared with ED-based DTW (Image Source: [15]).

Manjunatha et al. in [9] proposed a three step method (signal modeling, feature extraction and feature matching) for verifying the signature uses both genuine and forged signatures. The *x* position and the *y* position of all the signature points are extracted and represented, for each point, as one dimensional (1D) time domain signal along with pen moving angles as a third time domain signal. Due to length variability of these time domain signals, DWT is employed to reduce the dimensionality and extract features from these signals in a compact representation. However, system performance can further be improved by exploiting a different transformation. In [10], DWT is used to enhance the feature vectors in order to maximally separate genuine and forged signatures. More recently, Cpalka et al. in [13] used the combination of DWT and DCT for signatures' local information extraction. Wavelet packet with a fixed number of features and coefficients is employed to get better results than wavelet transform in [11].

Diaz et al. in [12] presented DCT based online signature verification approach in which a feature vector is created by applying DCT on 44 signatures features and extract the DCT coefficients to represent the feature vector in the reduced feature space. In [13], Cpalka et al. applied DCT on the coefficient vectors obtained from wavelet transform for dimensionality reduction.

Fourier Transform can also be seen as a promising approach for dimensionality reduction. As a proper transform can become an efficient tool for analyzing dynamic characteristics in time series patterns, Fast Fourier Transform also revealed many useful characteristics, in terms of signal frequency which was not the case in the actual signal [8].

II. Proposed Online Signature Verification Methodology

Our study is aimed at developing an OHSV system. Firstly, our emphasis was on selecting a suitable feature vector representation method for dimensionality reduction. Next, a feasible distance-oriented approach among the available state-of-the-art approaches for similarity measurement is identified for modeling and classification of signatures. The main contribution of our research work is the introduction of a new feature vector representation scheme to get more desirable results than the current results witnessed in different signature competitions held worldwide as well as handling forgeries at various levels.

A. *System Overview*

Mostly every static/dynamic signature verification system is trained by enrolling samples (reference signatures) which are then preprocessed. From the preprocessed signature samples, features are extracted which are useful in distinguishing these samples at the classification stage. Signature Modeling is then carried out for the registered individual’s signature samples and a threshold is computed. Classification phase requires a query signature (test signature) of user claiming to be a particular individual which is preprocessed, and same features are extracted as done in the training phase. If the user is already enrolled in the system, then the query signature is compared against the learnt model and accepted or rejected based on the threshold value [16]. For a new user, first its model of normality is learned from the provided set of reference signatures and a confidence or threshold value is computed and then compared with the corresponding model of normality to verify an individual. Dissimilarity score crossing a certain threshold rejects that user, otherwise authenticates him.

The process of OHSV can be sub-categorized into following steps namely data acquisition, preprocessing, feature extraction, and classification (training and verification) [17].

B. *Data Acquisition*

An OHSV system starts with some input to the system. For an OHSV system, input is captured at runtime (i.e., dynamic) which is usually taken by means of a digital tablet or alike devices. This captured information is then processed after digitization. In general, performance assessment of OHSV systems or algorithms is reported by the authors by exploiting their self-established databases, which are not accessible to other researchers. On the other hand, a variety of standard signature databases established by different institutes and research groups have been witnessed in prior research work in the domain of OHSV. In our research work, input comes from Japanese Online dataset taken from SigWiComp2013[17].

C. *Preprocessing*

Both the training and testing sets of signatures are inherent to noise and may also vary in length which makes preprocessing an important step. The degree of signature's preprocessing should be carefully done [10], [18]. Preprocessing should be performed with the objective to minimize loss to the signature temporal information, endpoints of strokes and points where the signature trajectory changes [19]. The most important function of preprocessing is to remove noise and additional jerks in the signatures [3], [20], [21]. We haven’t applied any preprocessing on the signature samples in our research work.

D. *Feature Extraction and Enhancement*

Feature extraction is the most pivotal step of verification process as the accuracy of system is highly relied upon over features used. Feature is any unique property or attribute that can be measured to represent signature effectively. Features for an online signature verification system are termed as (i) global features, which represent the whole signature; (ii) local features, which are extracted for each recorded point of signature sample; (iii) and segmental features, where features are extracted for each segment of signature sample unlike local or global features [22, 23]. A variety of features have been proposed and used in that falls within one of the stated categories. Total writing time, number of pen ups/downs and the number of strokes etc., are the examples of global features [18], [21], [24]. Some of the local features are speed, local curvature, pressure, tangential and centripetal acceleration [18], [21], [23]. Areas of high/low pen's pressure and high/low speed are the two common segmental features [3], [9], [25], [26].

E. *Dynamic Feature Vector*

The dynamic feature set refers to how the signature is signed than how it appears/looks. Signature dynamics are challenging for imposters to imitate (paper citation) because these not only captures the information of the signature’s overall shape, but also information of the individual sampling points (signature strokes) and other dynamics (speed) of the various signature strokes. Features are extracted from each point in OHSV as OHSV data requires to be represented in sequence of points. Dynamic features such as speed *Spi* and mean distance *MDi* for each sample *i*, are identified to have good discriminative potential. Selection of features plays an important role in the later processing and classification.

For this system, the dynamic information (raw data vector) obtained from the dataset comprises the following three-dimensional (3D) time-series data represented as in (1). Where *xts* and *yts* show position information of the signature and pts comprises pen down/up information at each sampling point at time *ts*. In this research, we are using pen down/up, speed and mean distance as our feature vector.

$$\mathrm{T(S)} = \{(\mathrm{xts}, \mathrm{yts}, \mathrm{pts})\} \quad (ts = 0{,}1{,}2, \dots, \mathrm{n}) \quad (1)$$

Firstly, we compute the derivatives of the original *x* and *y* signature time series. We exploit these derivatives to computed speed *sp* for each signature sample *i* as given in (2).

$$\mathrm{Sp}_i = \sum_{s=1,t=1}^{n} \sqrt{\left(\mathrm{xs} + 1 - \mathrm{xs}\right)2 + \left(\mathrm{yt} + 1 - \mathrm{yt}\right)2}\ (i = 1{,}2, \dots, \mathrm{n}) \quad (2)$$

Next, we have extracted the mean distance feature (*MDf*) by averaging the two dimensional (2D) raw data feature vector *V(S)* as represented in (3).

$$\mathrm{V(S)} = \{(\mathrm{xts}, \mathrm{yts})\} \quad (ts = 0{,}1{,}2, \dots, \mathrm{n}) \quad (3)$$

Now our feature vector for signature sample *S* takes the form given in (4).

$$\mathrm{T(S)} = \{(\mathrm{MDf}, \mathrm{Spf}, \mathrm{Pf})\} \quad (S = 1{,}2, \dots, \mathrm{n}) \quad (4)$$

After performing these basic operations on the acquired features, dimensionality of the signature samples is taken into account prior to signature model learning and classification.

F. *Dimensionality Reduction for Feature Vector Representation*

Dimensionality reduction is an important technique at this stage to deal with higher dimensionality problems in time series data. The goal of dimensionality reduction technique is to reduce the dimension of the samples while preserving most intrinsic and essential information even if multimodal scenarios exist within a dataset class.

It is possible to work in the raw point original coordinate space where signature sampling points are comparatively shorter. Conversely, direct manipulation of sampling point sequences for instance, greater than thousand or even more, seems to be impractical and unfeasible for feature extraction. The intention of applying dimensionality reduction is to come out with a feature extraction function *F* that decreases data dimensionality from *y* to *x* with *x* ≪ *y*. Similarity signature modeling and classification is then carried out in the reduced output feature space.

Time series data modeling and representation has also been carried out in prior research using a renowned transformation namely Discrete Fourier Transform. Time series is modeled compactly using a fixed number of coefficients [8], which results in quick signature modeling and classification. For each normalized signature, Fourier descriptors are computed and the selection of descriptors exhibiting highest magnitudes is made. Fisher discriminant analysis (FDA) is then employed with the imaginary as well as real part of the harmonics selected using empirical evaluation to identify the most suitable and appropriate features along with their associated weights. Traditionally, FDA [4] has been employed to serve the purpose but FDA fails to handle multimodality problem within classes. To cope with the multimodality issue, locality preserving projection is proposed [27] but it fails to handle labeled data due to its unsupervised nature. Local fisher discriminant analysis (LFDA) [28] is proposed and widely used to deal multimodality in time series data at localized level by taking into account the local structures of data.

In this research work, we employed Discrete Fourier Transform (DFT) to represent the data distribution in the higher dimensional space data given in (5) followed by LFDA to obtain reduced and compressed representation. The n-point fourier transform of $\{MD_i\}$, stated as a series $\{\overline{MD_f}\}$ of n complex numbers $f = 1, \dots.., n$ at discrete points. Where j is the imaginary unit $j = \sqrt{-1}$ and MD_f are complex numbers with the exception of CD_0 which is real. As the centroid distance-based time series is z-normalized, CD_0 which represents the mean of time series will always have a value of 0 and is ignored. Normally, DFT sequences are truncated after n terms. In our case, the feature vector is made up of $2(n-1)$ entries (from real and imaginary parts). More formally, let $x_i$and $\widehat{x_i}$be the real and imaginary part of $\overline{MD}$.

$$\overline{MD_f} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} MD_i \exp\left(-\frac{j2\pi f i}{n}\right) \quad f = 1, 2, \dots\dots, n \quad (5)$$

Signatures can be denoted in the reduced coefficient feature space by a $2(n-1)$ dimensional vector of DFT coefficients $F_{\boldsymbol{DFT}}$ as shown in (6). The given $\mathrm{F_{DFT}}$ can be efficiently employed for feature vector representation of signature samples.

$$F_{DFT} = [x_1, \widehat{x_1}, \dots\dots x_{n-1}, \widehat{a_{n-1}}] \quad (6)$$

Parodi et al. [29] proposed a template protection scheme which needs a fixed-length and compact feature vector representation of the signature time series. Liu et al. [8] exploited Fourier analysis to have fixed-length compact feature vector representation for their proposed individuality model for OHSV. Lagendijk et al. [30] acquired a fixed-length representation of fingerprint minutiae by exploiting fourier transform. Discrete cosine transformation has been used by Rashidi et al. [31] to get reduced feature vector representation.

G. *Signature Modeling*

It has been seen in the work of Liu et al. [8], and Lagendijk et al. [30] that DFT based dimensionality reduction is a suitable selection for compressed representation of signature time series data in the reduced space with data samples exhibiting length variation. DFT gives a uniform and unwavering features space representation of signature samples to cope with the issue of varying lengths in signature datasets. We bring DFT based coefficient feature space representation into play to accomplish the learning of signature patterns in online signature datasets. The resulting learnt output patterns from the learning process can then lead to the correct identification of a previously unseen signature pattern and assigning it to the class it belongs to. It needs much skill to realize a learning system to serve the purpose. The complexity of the realization of a learning system is directly dependent on the number of different signature patterns (inter-class and intra-class variability) exhibited by the enrolled users. In this study, signature modeling doesn't take larger dataset for training session. The learning process is capable of proficiently learning signature patterns in the presence of small training set where the membership count for previously unseen pattern is sufficient for not considering it as an abnormal pattern.

Multimodal m-mediods based modeling and classification approach [32] best suit the sample's estimated multi-modal distribution contained by a given signature pattern. Our model learning method works with coefficient feature space representation of training data (data) yielded from DFT as an input. Labeled information (labels) is also taken by the system as an input. The number of outputs mediods to be perceived (#output) along with the maximum iterations in training (train_iter) as the input parameters are taken by the system. The outcome of this method is the number of outputs mediods (#output) along with their associated weights. Given training samples *TS(i)* having enhanced and improved feature vector representation of signature samples be associated with signature class *i*, its normality model is generated as:

1) Initialize the Learning Vector Quantization (LVQ) network with the number of outputs mediods. LVQ is initialized with the number of outputs mediods (Moutput) empirically and taking the number of samples presented in a class as upper limit yield in (7). WhereꚌ is the number of samples in a given class and *m* is used to denote the number of values randomly thrown to get the desired number of outputs mediods in the dataset.

$$\text{Moutput} = \begin{cases} \text{Ꚍ}\ if\ \text{Ꚍ} < 3*m \\ 3*m\ if\ \text{Ꚍ} > 3*m \end{cases} \tag{7}$$

2) Initialize the weights *Wc* as per output mediod. A variety of methods exist for weights initialization. A very general and common approach is to assign random weight values, but it may slow down the training process of LVQ. Also, it may output some of the clusters with no representation associated with it. To cope with this problem, we approximate a multivariate Gaussian distribution function (PDF) by exploiting the training data. This PDF approximation generates samples in greater number maximizing the possibility of closeness of at least one sample from the generated samples with groups concealed in the training dataset. The weight vectors in our approach are estimated from the training data using a single multivariate Gaussian probability distribution function PDF $N(\mathrm{\mu},\sum)$ in (8). Where $M \in TS(\mathrm{i})$, μ and $\sum$ are the mean and covariance estimations to $TS(i)$. $PDF\ N(\mathrm{\mu},\sum)$ is then employed to get the number of outputs mediods Moutput along with the initialization of corresponding weights *Wi* with $1 \le i \le$ *Moutput*.

$$\text{PDF N}(\mathrm{\mu},\textstyle\sum) = \frac{1}{\sqrt{2\pi\sum}} exp\left[-\frac{(M-\mu)^2}{2\sum}\right] \tag{8}$$

3) Pass the feature vector (input) from $TS(i)$in succession and selection of the output mediod that is the nearest representative of provided input data during network training. That nearest output mediod is termed as the winning output mediod. Suppose DFT be the input feature vector and *Wc* denotes the associated weight of output mediod *i*, selection of the winning output mediod *i* is made in a way that the distance (Euclidean) between DFT and *Wc* is the smallest among all the output mediods, specified in (9) where *k* is the index of output mediod *i*.

$$\mathrm{k} = \mathrm{argmin}_i \|\mathrm{Wc}, \mathrm{F_{DFT}}\| \forall\, i \tag{9}$$

4) Adjust *Wc* to train LVQ so that it starts revealing the trend of the $TS(\mathrm{i})$. In this process, the neighboring mediods are also important. *Wc* of *i* and its neighboring mediods are adjusted to reveal topology preserving estimation of $TS(i)$. As a result, we have output mediods exhibiting similar trend which are very nearer in the LVQ network structural space. A subgroup of the weights comprising the wining neuron *i* with the center surrounded by its neighborhood is updated in (10) where $\alpha(t)$ is the learning rate of LVQ.

$$\mathrm{W_k(t+1)} = \mathrm{W_k(t)} + \mathrm{\alpha(t)}\|\mathrm{Wc}, \mathrm{F_{DFT}}\| \tag{10}$$

5) Converge the network training gradually from rough to refining of *Wc* by dropping down the learning rate. At the start, higher values are taken in the learning process to accomplish representation of input space by a quick adjustment of *Wc*. Convergence slows down after successive iterations resulting in lesser effect of new arriving data on LVQ. The LVQ network continues to learn and adjust itself to correctly describe the trends in signature data. Exponential decrease in $\alpha(t)$ over time *t* is given as in (11) where $train_{iter}$ are the maximum training iterations.

$$\mathrm{\alpha(t)} = 1 - e^{\frac{2(t-train_{iter})}{train_{iter}}} \tag{11}$$

6) Iterate through step 3-5 for all the training iterations and discard the output mediods holding no sample.

7) Determine the index *(x, y)* of the closest pair of *Wc* as in (12). Where Wi and *Wj* denotes the weight vector representations for output mediods *i* and *j* respectively.

$$(x,y) = \mathrm{argmin_{(i,j)}}\|\mathrm{Wi,Wj}\| \times \sqrt{|\mathrm{Wi}| + |\mathrm{Wj}}| \forall\, i,j \wedge i \neq\ j \tag{12}$$

8) Pairs which are closet are merged by using the weighted average. For instance, *Wx* and Wy are the weight vectors linked with output mediods indicating the most similar groups and *x* and *y* are the # of sample signatures mapped to these mediods respectively. *Wxy* (new weight vector) for the resultant merged group can be computed using (13).

$$\mathrm{W}_{xy} = \frac{|\mathrm{Wx}\ |\times\mathrm{Wx} + |\mathrm{Wy}\ |\times\mathrm{Wy}}{|\mathrm{Wx}\ | + |\mathrm{Wy}|} \tag{13}$$

9) Repeat Step 6-8 as late as weight vector *Wc* count add up to *m* and adjoin *W* to *M(i)* demonstrating the sample *i*.

Once we have done with mediods identification *M(k)*, the next step is to figure out set of normality ranges (NR) for each class. Normality ranges are identified to keep a set of samples to be a part of class that falls within the defined ranges and to maximally distinguish between normal and abnormal samples in the best possible way. However, defining an NR doesn't seem so simple as the work normality suggests because these ranges are defined in generalizations in terms of common patterns a signer exhibits. Since, a given class k may encompass different normality ranges *NR(k)* and each mediods within that class may comprises different *NR(k)*; a set of *NR(k)* for each class is identified after finding out the mediods as follows:

1) Start with NR(k) = {}.

2) From the list of identified mediods M(k), determine the index of the closest pair *(i, j)* at index *(x, y)* and update *NR(k)*. Suppose *Mi* and *Mj* are the two mediods indexed at *(x, y)*, closeness of mediod pair *(i, j)* is determined in a way that the Euclidean distance between *Mi* and *Mj* is the smallest among all the other identified mediods exist in the list, specified by (14) and (15).

$$(x, y) = \text{argmin}_{(\text{i,j})} \|\text{Mi}, \text{Mj}\|\big|\forall\, i, j \wedge i \neq\ j \qquad (14)$$

$$NR^{(k)} = NR^{(k)} \cup \left[\left|M_x, M_y\right|\right] \qquad (15)$$

3) Mediod Pairs which exhibit closeness are merged by using the weighted average. For instance, x and y are the weights linked with mediods *Mx* and *My* indicating the most similar pair. *Mx* and *My* are merged together to get a new vector *Mxy* which is computed using (16).

$$\text{M}_{xy} = \frac{|\text{Mx}\ |\times\text{Mx} + |\text{My}\ |\times\text{My}}{|\text{Mx}\ | + |\text{My}|} \qquad (16)$$

4) Repeat step 2 & 3 until M(k) converges to 1.

From the normality ranges *NR(k)* of a given class, normality ranges for each mediods *NR(m)* are identified. For each signature sample, traverse each class we have in the *TS(i)* in a continuous fashion and find out the closest match of mediods with that sample. We use *NR(k)* (attainable normality ranges) to get *NR(m)*. Given the *NR(m)* is the closest mediod for a given class; it results in the utmost samples of a given class to be kept in the NR of that mediod whereas allowing least possible samples from other classes to be contained within *NR(m)*. This process results in minimizing the false acceptance and false rejection of signature samples.

H. Signature Classification

After the identification of mediods *M(k)* and their equivalent normality ranges *NR(k)*, classification of signature samples *TS(i)* in a multimodal fashion is carried out. Classification involves grouping objects exhibiting similar behavior in their corresponding classes. The classification process for new signatures samples is carried out based on closeness of that previously seen/unseen sample to the learnt models of the available identified classes in the dataset. DFT based representation of feature vector for an unseen sample is computed and passed to the list of identified mediods of the entire set of dataset classes to serve the purpose as follows:

1) Calculate the query sample distance *QS(i)* with all the *M(k)* of different classes and ascending sort the outcomes.

2) Initialize index i of the mediod nearest to *QS(i)* to 1.

3) Label the i^{th} nearest mediod index with *Mi* and the equivalent class index to *Mc*. If the query sample *QS(i)* falls within the NR of i^{th} mediod, classify it in the equivalent class and finish the classification phase.

4) Increment the index from *i* to *i+1*.

5) Repeat step 3 and 4 until the value of index *i* exceeds the normality range of that mediod.

III. Experimental Studies of Proposed OHSV System

In this section, we present the experimental evaluation results to prove the effectiveness of our proposed methodology. Starting with an introduction about the dataset used in our experimental evaluations, we present the verification rates reported by the system at different levels of forgeries followed by comparisons with the competitors at hand.

A. Experimental Dataset

Japanese signature data collection is carried out using HP Elite Book 2730p tablet PC and a Microsoft INK SDK based developed data collection software with a sampling and resolution rate of 200Hz and 50 pixels/cm respectively. The online Japanese dataset contains ASCII files with the representation as: X, Y, and Z where X and Y denote position information and Z represent pen position information (Pen up (0), Pen down (100)). In the overall dataset collection process, 30 signers took part after a practice session to get acquainted with the signature capturing device. Forgers are allowed to see the genuine signatures of the authors whereas the original authors (signers) signed their signatures without having access to their previous signatures. The division of dataset into training and testing phases is as follows:

1) Training set: A total of 462 signatures (genuine) from 11 signers with 42 samples of each author and a total of 396 forgeries (skilled) with 36 samples per author are provided for the training session.

2) Testing set: The testing set contains 20 authors with 42 genuine signatures per author along with 36 forgeries each for both tasks. A detailed description is given in Table I.

The dataset reported is used in ICDAR2013[17] Competitions on Signature Verification and Writer Identification for Online and Offline Skilled Forgeries. Systems submitted by the online signature verification community that revealed best results under their experimental conditions are shown in Table I.

TABLE I. Results for Japanese Online Signature Verification (Source: [17])

Mode	Detection Accuracy	FAR	FRR
Online	70.55	30.22	29.56
Online	**72.55**	**27.36**	**27.56**
Online	72.47	27.50	27.56

B. Evaluation Criteria

Evaluation of Signature verification systems is based upon the error rate they yield. The lower the error rate, the higher the performance of the system. There might be two types of errors. When a system accepts the signature exhibited by a forger, it is termed as False Acceptance Rate (FAR). Conversely, if the system rejects the signature exhibited by a genuine signer, it is referred to as False Rejection Rate (FRR). Most authors report the performance in terms of FAR and FRR. FAR and FRR are inversely proportional which means that if we try to keep FAR down, FRR goes up and vice versa. Some authors have used the term Equal Error Rate (EER) which is the point at which both FAR and FRR are equal. Lower EER means higher accuracy. When it comes to comparison of the different biometric verification systems, EER is a widely used metric.

C. Experiment: Evaluation of Proposed OHSV System

The objective of training is 1) to detect the genuine signature class from the available n reference signatures; 2) to detect the forged signature imitated by some other author for each class.

D. Training

In the training phase, we have sub-divided the training set into training and cross-validation for model learning and parameter tuning. In the first step, we identify the system parameter and then perform model learning based on those parameters. Model learning and verification is performed by exploiting858 reference signatures including genuine signatures and skilled forgeries. By varying the parameters i.e., threshold, # of mediods, # of iterations etc. An ideal value of threshold (i.e., 1.055) is identified. The detailed methodology is discussed in section 3.6. After threshold identification for each class, cross-validation involves the tuning of parameters i.e., threshold and normality ranges with the intention of minimizing false positive and false negative by exploiting the model and labeled information.

E. Evaluation

This phase sub-divides the evaluation set into training set and test set. From the training set, model is learnt, which in conjunction with the tuned parameters from the first phase is used to classify the test signatures into genuine and forged ones. Model learning and verification is performed by exploiting 1560 test signatures including genuine signatures and skilled forgeries. The detection and the forgery accuracy obtained in the training and the evaluation along with the competitor results are given in Table II and III respectively.

TABLE II. DETECTION & FORGERY ACCURACY IN TRAINING

Mode	# of authors	# of signatures (Total)	Detection Accuracy %	FAR %	FRR %
ICDAR2013	11	858	72.55	27.36	27.56
Our proposed approach	11	858	84.23	15.77	17.54

TABLE III. DETECTION & FORGERY ACCURACY (EVALUATION)

Mode	# of authors	# of signatures (Total)	Detection Accuracy %	FAR %	FRR %
ICDAR2013	20	1560	72.55	27.36	27.56
Our proposed approach	20	1560	82.84	17.16	19.46

IV. CONCLUSION AND FUTURE PERSPECTIVES

A. Conclusion

In handwritten signatures, there is a great interest toward the development of effective and robust signature methods for online signatures. The last few decades have witnessed extensive research in the development of OHSV systems and a wide range of verification techniques and framework proposed with promising results. However, there exist a number of challenges which make OHSV a hot research area. Due to the inter-class and intra-class variability, detection of signatures with respect to their corresponding classes with maximum precision is still a challenging task. Similarly, the robustness of the verification system against the imposter's imitation of generating the genuine signature patterns by minimizing false positive and false negative ratio is another challenge at hand.

To meet the challenges of OHSV systems, this research work is aimed at developing an OHSV system for the identification of signatures and the detection of skilled forgeries with accuracy greater than the available state-of-the-art approaches. To cope with the challenges of OHSV systems, we have presented a compact feature vector representation by incorporating speed, pen positions and mean distance as features for our system as given in Section III. From this compact representation, we have learnt the signature model and classify the signatures according to the identified system parameters. During the training phase, we have obtained 84.23% detection accuracy with 15.77% false acceptance rate and 17.54% false rejection rate. In the evaluation phase, we obtained 82.84% detection accuracy with 17.16% false acceptance rate and 19.46% false rejection rate.

B. Future Perspectives

Development of an OHSV system for targeting skilled forgeries is a very challenging task. We have attempted to address some of these issues with our proposed OHSV approach. The proposed system realizes very promising signature verification rates. However, the verification rates are slightly lower as the size of dataset in the evaluation phase increases. The most obvious extension of the proposed system is to speed up the processing by varying features as well as exploiting some other feature vector representation technique to achieve more compact and compressed representation. Another possible extension is to integrate our proposed approach with offline signature verification and then evaluate system performance in the hybrid environment in the near future.

REFERENCES

[1] A. Sharma, and S. Sundaram, "On the exploration of information from the DTW cost matrix for online signature verification," IEEE transactions on cybernetics, 2018, vol. 48(2), pp. 611-624.

[2] Z, Chen, X. Xia, and F. Luan, "Automatic online signature verification based on dynamic function features," in 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016, IEEE.

[3] K. Cpałka, and M. Zalasiński, "On-line signature verification using vertical signature partitioning," Expert Systems with Applications, 2014, vol. 41(9), pp. 4170-4180.

[4] R. Doroz, P. Porwik, and T. Orczyk, "Dynamic signature verification method based on association of features with similarity measures," Neurocomputing, 2016, vol. 171, pp. 921-931.

[5] V. Iranmanesh, S. M. S. Ahmad, W. A. W. Adnan, S. Yussof, O. A. Arigbabu, and F. L . Malallah, "Online handwritten signature verification using neural network classifier based on principal component analysis," The Scientific World Journal, 2014.

[6] A. Sharma, and S. Sundaram, "An enhanced contextual DTW based system for online signature verification using Vector Quantization," Pattern Recognition Letters, 2016, vol. 84, pp. 22-28.

[7] A. Beresneva, A. Epishkina, and D. Shingalova, "Handwritten signature attributes for its verification," in 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018, IEEE.

[8] Y. Liu, Z. Yang, and L. Yang, "Online signature verification based on DCT and sparse representation," IEEE transactions on cybernetics, 2015, vol. 45(11), pp. 2498-2511.

[9] K. S. Manjunatha, S. Manjunath, D. S. Guru, and M. T. Somashekara, "Online signature verification based on writer dependent features and classifiers," Pattern Recognition Letters, 2016, vol. 80, pp. 129-136.

[10] X. Song, X. Xia, and F. Luan, "Online signature verification based on stable features extracted dynamically," IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, vol. 47(10), pp. 2663-2676.

[11] M. Nilchiyan, R.B. Yusof, and S. Alavi, "Statistical on-line signature verification using rotation-invariant dynamic descriptors," in 2015 10th Asian Control Conference (ASCC), 2015, IEEE.

[12] M. Diaz, A. Fischer, M. A. Ferrer, and R. Plamondon, "Dynamic signature verification system based on one real signature," IEEE transactions on cybernetics, 2016, vol. 48(1), pp. 228-239.

[13] K. Cpałka, M. Zalasiński, and L. Rutkowski, "A new algorithm for identity verification based on the analysis of a handwritten dynamic signature," Applied soft computing, 2016, vol. 43, pp. 47-56.

[14] A. N. Azmi, D. Nasien, and F.S. Omar, "Biometric signature verification system based on freeman chain code and k-nearest neighbor," Multimedia Tools and Applications, 2017, vol. 76(14), pp. 15341-15355.

[15] R. Plamondon, G. Pirlo, and D. Impedovo, "Online signature verification," Handbook of Document Image Processing and Recognition, 2014, pp. 917-947.

[16] P. Kanu, and K.G. Singh, "PCA based Feature Selection of Online Signature Verification system using DFrCT," 2016, Doctoral dissertation.

[17] M. I. Malik, M. Liwicki, l. Alewijnse, W. Ohyama, M. Blumenstein, and B. Found, "ICDAR 2013 competitions on signature verification and writer identification for on-and offline skilled forgeries (SigWiComp 2013)," In 2013 12th international conference on document analysis and recognition, pp. 1477-1483, IEEE.

[18] D. Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll, "SVC2004: First international signature verification competition," In International conference on biometric authentication, July 2004, pp. 16-22, Springer, Berlin, Heidelberg.

[19] P. Porwik, R. Doroz, and T. Orczyk, "Signatures verification based on PNN classifier optimised by PSO algorithm," Pattern Recognition, 2016. vol. 60, pp. 998-1014.

[20] M. Bashir, and J. Kempf, "Area bound dynamic time warping based fast and accurate person authentication using a biometric pen," Digital Signal Processing, 2013, vol. 23(1), pp. 259-267.

[21] M. I. Malik, M. Liwicki, and A. Dengel, "Local Features for Forensic Signature Verification," in Proceedings of the ICIAP 2013 International Workshops on New Trends in Image Analysis and Processing—ICIAP 2013-Volume 8158. 2013. Springer-Verlag.

[22] M. Zalasiński, K. Cpałka, and E. Rakus-Andersson, "An idea of the dynamic signature verification based on a hybrid approach," in International Conference on Artificial Intelligence and Soft Computing. 2016, Springer.

[23] M. Zalasiński, K. Cpałka, and Y. Hayashi, "New method for dynamic signature verification based on global features," in International Conference on Artificial Intelligence and Soft Computing, 2014, Springer.

[24] C. Gruber, T. Gruber, S. Krinninger, and B. Sick, "Online signature verification with support vector machines based on LCSS kernel functions," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2009, vol. 40(4), pp.1088-1100.

[25] S. Rohilla, A. Sharma, and R. Singla, "Online signature verification at sub-trajectory level," in Advanced Computing, Networking and Informatics, vol. 2, 2014, Springer, pp. 369-374.

[26] K. Cpałka, M. Zalasiński, and L. Rutkowski, "New method for the on-line signature verification based on horizontal partitioning," Pattern Recognition, 2014, vol. 47(8), pp. 2652-2661.

[27] R. A. Fisher, "The use of multiple measurements in taxonomic problems," Annals of eugenics, 1936, vol. 7(2), pp. 179-188.

[28] P. Porwik, R. Doroz, and T. Orczyk, "The k-NN classifier and self-adaptive Hotelling data reduction technique in handwritten signatures recognition," Pattern Analysis and Applications, 2015, vol. 18(4), pp. 983-1001.

[29] M. Parodi, and J.C. Gómez, "Legendre polynomials based feature extraction for online signature verification. Consistency analysis of feature combinations," Pattern Recognition, 2014, vol. 47(1), pp. 128-140.

[30] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," IEEE Signal Processing Magazine, 2013, vol. 30(1), pp. 82-105.

[31] S. Rashidi, A. Fallah, and F. Towhidkhah, "Similarity evaluation of online signatures based on modified dynamic time warping," Applied Artificial Intelligence, 2013, vol. 27(7), pp. 599-617.

[32] M. U. Akram, S. Khalid, A. Tariq, and M. Y. Javed, "Detection of neovascularization in retinal images using multivariate m-Mediods based classifier," Computerized Medical Imaging and Graphics, 2013, vol. 37(5-6), 346-357.

Image Denoising using Wavelet Cycle Spinning and Non-local Means Filter

Giat Karyono[1], Asmala Ahmad[2], Siti Azirah Asmai[3]
Faculty of Computer Science, Universitas Amikom Purwokerto, Purwokerto, Indonesia[1]
Faculty of Information and Communications Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia[2, 3]

***Abstract*—Removing as much noise as possible in an image while preserving its fine details is a complex and challenging task. We propose a wavelet-based and non-local means (NLM) denoising method to overcome the problem. Two well-known wavelets: dual-tree complex wavelet transform (DT-CWT) and discrete wavelet transform (DWT), have been used to change the noise image into several wavelet coefficients sequentially. NLM filtering and universal hard thresholding with cycle spinning have been used for thresholding on its approximation and detail coefficients, respectively. The inverse two-dimensional DWT was applied to the modified wavelet coefficients to obtain the denoised image. We conducted experiments with twelve test images on the set12 data set, adding the additive Gaussian white noise with variances of 10 to 90 in increments of 10. Three evaluation metrics, such as peak signal noise to rate (PSNR), structural similarity index metric (SSIM), and mean square error (MSE), have been used to evaluate the effectiveness of the proposed denoising method. From these measurement results, the proposed denoising method outperforms DT-CWT, DWT, and NLM almost in all noise levels except for the noise level of 10. At that noise level, the proposed denoising method is lower than NLM but better than DT-CWT and DWT.**

Keywords—Image denoising; discrete wavelet transform (DWT); dual-tree complex wavelet transform (DT-CWT); non-local means (NLM); cycle spinning

I. Introduction

The emergence of noise in digital images is possible during image acquisition, transmission, and processing steps [1]. The additive Gaussian noise is the type of noise most often found [2]. Hence, suppressing this noise type from digital images is necessary before further processing like texture analysis, feature extraction, and segmentation [3]. Maintaining the essential features of the images, such as edges and textures, is one of the main issues faced during the denoising process [4]. However, since noise, texture, and edge are high-frequency components, it is arduous to distinguish them in the denoising process, and the denoised images could ineluctably lose some details [5].

Numerous denoising methods have been developed in the literature. Among such methods, wavelet transforms and non-local means (NLM) filters are one of the suggested denoising methods [6]. In the wavelet transforms method, the noisy image is decomposed into the low- and high-frequency sub-bands, followed by wavelet thresholding on these frequency sub-bands. The wavelet thresholding is quite effective applied to the high-frequency sub-bands but fails when applied to the low-frequency sub-band [7]. In most of them, discrete wavelet transform (DWT) was widely used, but it has three other main issues. These issues are lack of poor directionality, shift-invariant, and aliasing [8]. Conversely, the non-local means filter is highly effective in retaining the proper morphology of the signal at low-frequency. At the same time, the NLM filter fails to properly denoise the high-frequency [7]. Another drawback is a very time-consuming process.

The ill effects of noise can be reduced by addressing the shortcomings of both above-denoising methods. Motivated by this, a denoising method is proposed by utilizing the efficacy of both wavelet- and NLM-based methods. The NLM is more efficient in denoising the low-frequency content, and applying it in the wavelet domain can significantly decrease the processing time [9]. On the other hand, although DWT offers the advantages of smoothness and adaptation, as Coifman and Donoho [10] suggest, DWT exhibits visual artefacts known as Gibbs phenomena in the vicinity of discontinuities. To address this issue, the translation-invariant denoising method called cycle spinning is applied to remove such artefacts. Meanwhile, DT-CWT is a well-known method introduced to solve the main issues of DWT. Its implementation is to be combined with NLM can be found in [11] [12] [13]. Two well-known wavelets, i.e., DT-CWT and DWT with one-level decomposition, are applied sequentially. Only the low-frequency sub-band is denoised utilizing the NLM filtering. Since the high-frequency sub-bands contain noises, those are denoised using hard thresholding with cycle spinning-based. The inverse DWT on the modified sub-bands are used to reconstruct the image.

The main contributions of the proposed denoising method are summarized below:

1) The proposed denoising method utilizes the advantages of the wavelet- and NLM-based methods for eliminating high-frequency and low-frequency noises present in the noisy image, respectively.

2) The main disadvantage of DWT, i.e., it does not effectively eliminate the low-frequency noise, lack of poor directionality, shift-invariant, and aliasing, is mainly resolved by NLM, cycle spinning, and DT-CWT.

The fundamentals of the DWT, DT-CWT, wavelet thresholding, cycle spinning, and non-local means filter are summarized in the Section ‘‘Theoretical Background’’, and the proposed denoising algorithm to suppress noise is explained in the Section ‘‘The Proposed Method’’. In the Section ‘‘Experimental Results’’, the performance of the proposed denoising method is evaluated using test images simulated with additive white Gaussian noise. Section ‘‘Conclusion’’ describes our conclusions.

II. Theoretical Background

Discrete Wavelet Transform

In general, the discrete wavelet transform (DWT) is an image decomposition at the sub-band frequency of the image. The wavelet transforms sub-band component is generated by decreasing the decomposition level. DWT implementation can be done by passing the signal through a low and high pass filter. Filterization itself is a function used in signal processing [9]. The decomposition of averages and differences plays a vital role in understanding the wavelet transform. Averaging is done by calculating the average value of 2 pairs of data using Eq. (1).

$$p = \frac{x+y}{2} \tag{1}$$

where, p is pixel in the digital image, x is the first number in decimal is obtained, and y is the second number in decimal is obtained.

While the reduction is carried out with the following Eq. (2).

$$p = \frac{x-y}{2} \tag{2}$$

The decomposition process is carried out on the results of the flattening process. The result of the decomposition process is a combination of the flattening process results with all the image pixel reduction processes. The decomposition process is carried out in two stages. The first stage is performed on all rows, and then is carried out in the column direction on the resulting image of the first stage.

The signal is passed through a high-pass filter and a low-pass filter, and then half of each output is taken as a sample through a down-sampling operation or referred to as a one-level decomposition process. The output of the low-pass filter is used as input for the next level of the decomposition process. This process is repeated until the desired level of the decomposition process. The decomposition process produces the wavelet coefficient, a combination of the output of the last high-pass and low-pass filters. The wavelet coefficient contains compressed transformed signal information.

The one-level decomposition is written using the mathematical expressions in Eq. (3) and (4):

$$y_{high}[k] = \sum_n x[n]h[2k= n] \tag{3}$$

$$y_{low}[k] = \sum_n x[n]k[2k= n] \tag{4}$$

where $y_{high}[k]$ is the result of high pass filter (which is a detail of signal information), $y_{low}[k]$ is the result of low pass filter (which is a rough approximation of the scaling function), $x[n]$ is source signal, $h[n]$ is high pass filter, and $g[n]$ is low pass filter.

Using this DWT coefficient, the Inverse Discrete Wavelet Transform (IDWT) process can be carried out to reconstruct it into the original signal, as shown in Eq. (5).

$$y_{high}[k] = \sum_n x[n]h[2k= n] \tag{5}$$

In the discrete wavelet transform, an image is decomposed into sub-images (sub-bands) at different frequencies and orientations, namely low-low (LL), low-high (LH), high-low (HL), and high-high (HH). An illustration of the discrete wavelet transform is shown in Fig. 1.

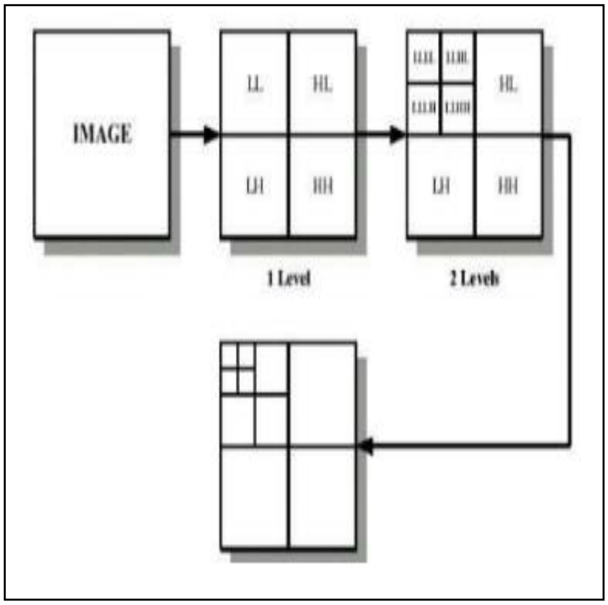


Fig. 1. Two dimensional discrete wavelet transform.

Several parameters, such as the selected mother wavelet function and decomposition level, should be chosen carefully when DWT-based processing methods are used [14]. Due to improper selection of the mother wavelet function and the number of decomposition levels may cause distortion or under denoising of the signal [15]. To ensure an effective denoising procedure of the image denoising, we selected the mother wavelet families of Coiflets with the order of 4 (Coif4) and one-level decomposition.

Dual-Tree Complex Wavelet Transform

Dual-Tree Complex Wavelet Transform (DT-CWT) was the combination of the advantages of DWT and CWT (complex wavelet transform). It is shift invariance, perfect reconstruction, either in directional selectivity, has a little redundancy, and the minimalist computation algorithm. DT-CWT transformation was a variation of DWT implementation, but the main difference is DT-CWT uses two tree filters, as shown in Fig. 2.

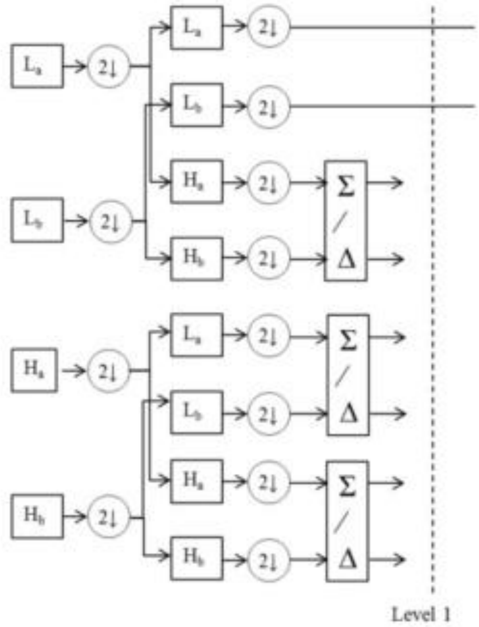


Fig. 2. Two dimensional dual-tree complex wavelet transform.

Unlike the DWT, the DT-CWT is built through a complex-valued wavelet function $\varphi_c = (t)$ and a complex-valued scaling function. The complex-valued wavelet function is built as follows: The complex-valued scaling function is expressed similarly [16].

$$\varphi_c(t) = \varphi_r(t) + j\varphi_i(t) \tag{6}$$

where $\varphi_r(t)$ is even and real, $j\varphi_i(t)$ is odd and imaginary, but $\varphi_i(t)$ is real. In addition, $\varphi_r(t)$ and $\varphi_i(t)$ form a Hilbert transform pair.

Consider 2D DT-CWT associated with the row-column implementation of the 1D DT-CWT.

$$\varphi(x, y) = \varphi(x)\varphi(y) \tag{7}$$

where $\varphi(x)$ and $\varphi(y)$ are given by formula (6). To represent an integrated real 2D signal completely, the row or column of the complex conjugate filter is required. Three sub-bands are produced in both the first and second quadrants, corresponding to six directions in space: $\pm 15^0$, $\pm 45^0$, and $\pm 75^0$.

We used one-level decomposition when applying DT-CWT in this study.

Wavelet Thresholding

Wavelet thresholding is a method that maintains wavelet coefficients whose value is greater than a particular threshold value and ignores small wavelet coefficients. This value is called the threshold value, and the estimator can be written as:

$$\hat{\mathbf{f}}_\lambda(\mathbf{u}) = \sum_{j=0}^{j-1} \sum_{k=0}^{2^{J-1}} \mathrm{I}\left\{\left|W_{(j,k)}^{(n)}\right| > \lambda\right\}^{W_{(j,k)}^{(n)}\psi_{j,k}(u),} \tag{8}$$

where λ is threshold value, I_A represents the indicator function of set A. The estimator in Equation (8) can be considered a non-linear operator in the coefficient vector, which produces vector $\hat{\theta}$ of coefficient estimation. The thresholding estimator is defined as.

$$\hat{\theta}_{j,k} = \frac{\sigma}{\sqrt{n}} \delta\lambda \left(\frac{\sqrt{n w_{j,k}^{(n)}}}{\sigma} \right) \tag{9}$$

with $\delta\lambda$ is thresholding function and λ is threshold parameter.

The thresholding steps can be sequenced as follows: select thresholding function, value estimation σ, and selection of threshold parameter.

1) Thresholding function: According to Coifman and Donoho [10], there are two thresholding functions: hard and soft thresholding. Both can be written with their respective equations as follows:

$$\delta_\lambda^H(t) = \begin{cases} t, & \text{if} |t| > \lambda \\ 0, & \text{other} \end{cases} \quad \text{for hard thresholding} \tag{10}$$

$$\delta_\lambda^S(t) = \begin{cases} t - \lambda, & \textbf{if } \mathbf{t} > \lambda \\ 0, & \text{if } |t| \leq \lambda \\ t + \lambda, & \textbf{if } \mathbf{t} < -\lambda \end{cases} \text{, for soft thresholding function} \tag{11}$$

The hard thresholding function is better known because there is a discontinuity in the function, so the t values above the threshold λ are not touched. On the other hand, the soft thresholding function is continuous since the t value is above the threshold λ. In this study, we used hard thresholding rules.

2) Value estimation σ: Wavelet thresholding is enforced rules where at least to estimate the value σ because its value is usually unknown. The value σ is deviation standard value from the observation $T_1, T_2, T_3, \ldots, U_n$. The authors in [14] proposed an estimation σ is based on empirical wavelet coefficient at the high level resolution. This consideration is because, at the highest coefficient level, there is usually a lot of noise. Given in [17], the Median of Absolute Deviation (MAD) estimation to estimate value is expressed as

$$\hat{\sigma} = \frac{median(|W_{J-1,k}^{(n)} - median\,(W_{J-1,k}^{(n)})|)}{0\ 6745} \tag{12}$$

with $\mathbf{J = \log_2(n)}$. Because coefficient $W_{J-1,k}$, $\mathbf{k = 0}$, $\mathbf{2^{J-1} - 1}$ close to zero, then it can be replaced median value ($W_{J-1,k}^{(n)}$) above with zero.

3) Selection of threshold parameter: There are two selection categories to select the optimal threshold value: selecting one threshold value for all resolution levels (global selection) and selecting a threshold that depends on the resolution level (level-dependent thresholding). Ogden [17] provides two threshold choices for global threshold selection that only depend on the number of n observation data. Both are tabulated by Donoho and Johnstone [17], known as the universal threshold $(\lambda_j = \sqrt{2 \log n})$ and the minimax threshold. Minimax threshold values are always smaller than universal threshold values for the same sample size. Choosing a threshold that depends on the level of resolution means choosing λ_j depending on the resolution j. Thus there is a possibility of differences in the threshold value λ_j selected from for each level of wavelet j. This study used the universal threshold, which applied to the detail coefficients.

Cycle Spinning

References [14] introduced a translation-invariant denoising method called cycle spinning. This method calculates different estimates of noisy image by shifting images to different phases and then linearly averaging these estimates. The cycle spinning will result in different estimates of the original image with statistically other noises reduced by the averaging.

If we show the two-dimensional circular shift by S_{ij}, the denoising operator by T, and the thresholding operator by η, then cycle spinning can be expressed as:

$$\hat{C} = \frac{1}{k_1 k_2} \sum_{i=1,j=1}^{k_1 k_2} C_{-1-j} \left(T^{-1}\left(\eta\left(T\left(C_{ij}(x)\right)\right)\right)\right), \tag{13}$$

where k_1 and k_2 are the maximum number of shifts. In this study, we used two shifts.

Non-local Means

The non-local means (NLM) was proposed by Buades et al. [17]. Given the 2D noise image $v = \{v(k) | k \in K\}$, where k is the pixel index and K is the number of pixels, then the NLM is calculated as a weight average of all the pixels in an image,

$$\text{NLM}[v](k) = \sum_{j \in v_i} w(a,b)\, v(j) \tag{14}$$

where v_i is a square neighborhood of the pixel k referred to as the search window, $w(a,b)$ is the weight depend on the similarity between the pixels a and b, and satisfying the conditions $0 \leq w(a,b) \leq 1$ and $\sum_j w(a,b) = 1$. The weight is built as follows.

$$w(a,b) = \frac{1}{Z(a)} \exp\left(-\frac{\|x(\eta_a) - x(\eta_b)\|_2^2}{h^2}\right) \tag{15}$$

where η_a is a square neighborhood of fixed size and centered at a pixel a referred to as the similarity window, $x(\eta_a)$ denotes the vector of pixel values within the similarity windows, and $Z(a)$ is the normalizing factor,

$$Z(b) = \sum_b \exp\left(-\frac{\|x(\eta_b) - x(\eta_b)\|_2^2}{h^2}\right) \tag{16}$$

where the parameter filter h controls the degree of filtering. We set the parameter of NLM as follows: the patch size is 7x7 pixels, the search window is 5x5 pixels, and the filtering parameter h is obtained from the deviation standard value.

III. Proposed Method

The wavelet-based and NLM denoising techniques are efficient and have complementary benefits and drawbacks. Therefore, combining both can produce a powerful denoising method. However, the direct cascading of these methods will result in a denoising system that is ineffective and costly in computation. Henceforth, this study combines wavelet-based and NLM techniques to get the desired results.

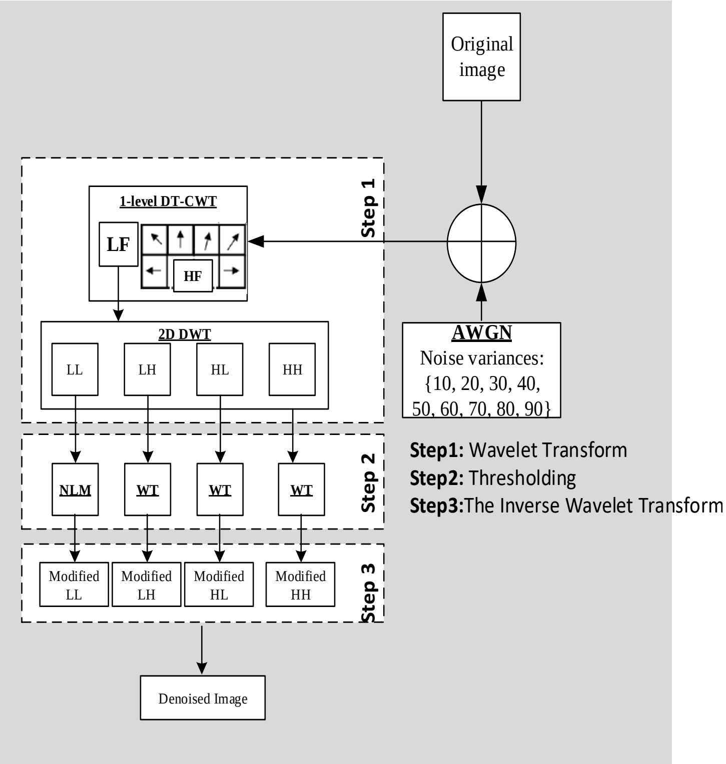


Fig. 3. Block diagram of the proposed denoising method.

The block diagram of the proposed denoising method is shown in Fig. 3. It consists of three steps: decomposition of the noisy image using DT-CWT and DWT, denoising of the low- and high-frequency sub-bands, and reconstruction. A detailed description of the proposed denoising method and its advantages over DWT, DT-CWT, and NLM is discussed in the following.

Step 1: Decomposition: Decompose the noise image into one real and imaginary approximation coefficient and the six real and imaginary detail coefficients using 2D DT-CWT with one-level decomposition. Process the real approximation coefficient through DWT to obtain the high- and low-frequency sub-bands. The decomposition process, including the selection of the decomposition level and the mother wavelet function, has already been briefed in Section II.

Step 2: Denoising of the low- and high-frequency sub-bands: Denoise the low-frequency sub-band obtained in one level of decomposition using the non-local means filter. Next, carry out the hard-thresholding-based cycle spinning to the high-frequency sub-bands. The selection of NLM parameters and wavelet threshold has already been briefed in Section II.

Step 3: Reconstruction: Reconstruct the denoised image by inverse 2D-DWT using denoised low-frequency and high-frequency sub-bands. The reconstruction process has already been briefed in Section II.

In the proposed denoising method, the thresholding of high-frequency sub-bands will effectively eliminate the high-frequency noise components present in the noisy image. Besides, the NLM filter will denoise the low-frequency sub-band and retain the morphological structure of the image. The proposed denoising process consists of only one-level DWT decomposition in contrast to the earlier DWT method requiring more significant decomposition levels to effectively denoise the low-frequency components. In addition, hard thresholding-based cycle spinning of the high-frequency and NLM filter of the low-frequency sub-band can be performed simultaneously. As a result, the overall computing cost of the process is significantly decreased.

IV. Experimental Results

As was the case with the experiment carried out by Balasubramanian et al. [8], the effectiveness of the proposed denoising method is evaluated on four standard test images: Lena, the boat, the house, and the cameraman corrupted by AWGN with zero mean and standard deviation σ=10 to 90 with increment 10. The proposed denoising method has been implemented by writing Python code. Python function random is used to add additive white Gaussian noise to the test images.

During experiments, to evaluate the effectiveness of the denoising performance, apart from the peak signal-to-noise ratio (PSNR) and mean square error (MSE), which is the general measure of denoising performance, we also measure the structural similarity index (SSIM).

As we stated earlier, our proposed denoising method is compared with the conventional DT-CWT, DWT, and NLM using PSNR, MSE, and SSIM. The parameters of each method have been set according to our experiment to have uniformity in comparison. The denoising experiment has been performed, and the results of PSNR, MSE, and SSIM are tabulated in Table I.

A detailed examination of the tabulated results led to the following findings:

- Cameraman image: The proposed denoising method outperforms DT-CWT, DWT, and NLM when the noise level of 40 and above for all measurement results. In contrast, the obtained PSNR and MSE are lower than NLM when the noise level of 30 and below. However, still better than DT-CWT. Lower than DWT when the noise level is 10 when measured using PSNR. Including a noise level of 20 when measured using MSE. For the SSIM result, the proposed denoising method outperforms all except the noise level of 10. At that noise level, NLM is better than the proposed denoising method.

- House and Boat images: When observing both images using PSNR, MSE, and SSIM results, the proposed denoising method is superior to DT-CWT, DWT, and NLM in almost all the noise levels except for 10. At that noise level, NLM is better than the proposed denoising method. However, the proposed denoising method is still better than DT-CWT and DWT.

- Lena image: Out of these three other test images; Lena's image is the only best gain of our proposed denoising method because it the superior to all techniques used for comparison in all noise levels with different measurement techniques results.

To compare the visual quality of the denoising methods, the free noise, noisy, and denoised images when AWGN added with variances of 70 are shown in Fig. 4.

- Cameraman image: Based on denoising results on this image show the proposed denoising method has better appearance of the denoising result. The noise in the texture of clothes and background can be effectively removed, including the noise present at edges.

- House image: The proposed denoising method results on house image can be seen in Fig. 4. It can be seen that the proposed denoising method is better in removing amount noise on the texture of house wall and sky as background. Comparatively, noise present in the edges also can be removed very well.

- Lena image: Like the cameraman and house images, the result of the proposed method in removing the noise in the hat, skin, mirror, hair, and wall is better than other denoising method.

- Boat image: Visually, the proposed method result is more effective in suppressing the noise in the boat image.

From the analysis above, it can be stressed that the suggested work effectively addresses the individual limitations of NLM and DWT approaches.

TABLE I. THE PSNR, MSE, AND SSIM RESULTS OF THE DENOISING METHOD

	Cameraman Image											
	DT-CWT			**DWT**			**NLM**			**The Proposed Method**		
	PSNR	**MSE**	**SSIM**	**PSNR**	**MSE**	**SSIM**	**PSNR**	**MSE**	**SSIM**	**PSNR**	**MSE**	**SSIM**
10	28.4022	0.001445	0.9154	29.9913	0.001002	0.9129	**32.7869**	**0.000526**	**0.9583**	28.9348	0.001278	0.9305
20	26.2598	0.002366	0.8391	26.6179	0.001357	0.8982	**28.6740**	**0.001357**	0.8982	27.5517	0.001757	**0.9001**
30	24.6384	0.003437	0.7501	24.5911	0.003474	0.7579	**26.5071**	**0.002235**	0.8363	26.4572	0.002261	**0.8742**
40	23.1465	0.004846	0.6617	23.0228	0.004986	0.6795	24.9824	0.003175	0.7726	**25.5801**	**0.002767**	**0.8497**
50	21.8069	0.006596	0.5811	21.8012	0.006605	0.6069	23.6854	0.004280	0.7072	**24.8014**	**0.003310**	**0.8242**
60	20.6161	0.008677	0.5116	20.7957	0.008326	0.5414	22.5294	0.005585	0.6421	**24. 0689**	**0.003918**	**0.7972**
70	19.5441	0.011107	0.4527	19.8959	0.010243	0.4824	21.5184	0.007049	0.5807	**23.3722**	**0.004600**	**0.7686**
80	18.5698	0.013900	0.4031	19.0999	0.012303	0.4302	20.6467	0.008617	0.5254	**22.7298**	**0.005334**	**0.7399**
90	17.6842	0.017044	0.3612	18.3364	0.014667	0.3829	19.8715	0.010300	0.4757	**22.1486**	**0.006097**	**0.7108**
House Image												
10	32.6951	0.000538	0.9154	32.6901	0.000538	0.9129	**34.0484**	**0.000394**	**0.9583**	34.0112	0.000397	0.9305
20	29.3708	0.001156	0.8391	29.0037	0.000896	0.8982	30.4759	0.000896	0.8982	**31.9477**	**0.000639**	**0.9001**
30	26.7387	0.002119	0.7501	26.6371	0.002169	0.7579	28.1672	0.001525	0.8363	**30.3818**	**0.000916**	**0.8742**
40	24.6448	0.003432	0.6617	24.8644	0.003263	0.6795	26.3344	0.002326	0.7726	**29.0941**	**0.001232**	**0.8497**
50	22.9206	0.005104	0.5811	23.4535	0.004515	0.6069	24.8298	0.003289	0.7072	**27.8881**	**0.001626**	**0.8242**
60	21.4704	0.007128	0.5116	22.2593	0.005944	0.5414	23.5629	0.004403	0.6421	**26.8096**	**0.002085**	**0.7972**
70	20.2175	0.009511	0.4527	21.1955	0.007594	0.4824	22.5096	0.005611	0.5807	**25.8506**	**0.002600**	**0.7686**
80	19.1169	0.012255	0.4031	20.2345	0.009474	0.4302	21.5705	0.006966	0.5254	**25.0130**	**0.003153**	**0.7399**
90	18.1388	0.015350	0.3612	19.3887	0.011512	0.3829	20.7281	0.008457	0.4757	**24.2695**	**0.003742**	**0.7108**
Lena Image												
10	33.4483	0.000452	0.9446	32.4713	0.000566	0.9315	34.0168	0.000397	0.9541	**34.5822**	**0.000348**	**0.9591**
20	29.8266	0.001041	0.8726	29.0362	0.000897	0.9012	30.4733	0.000897	0.9012	**32.0975**	**0.000617**	**0.9360**
30	27.0414	0.001976	0.7813	26.7845	0.002097	0.7903	28.1826	0.001520	0.8374	**30.3599**	**0.000920**	**0.9113**
40	24.8681	0.003260	0.6884	25.1226	0.003074	0.7151	26.4086	0.002286	0.7686	**29.0039**	**0.001258**	**0.8852**
50	23.1044	0.004893	0.6026	23.7793	0.004189	0.6442	24.9782	0.003178	0.7011	**27.8807**	**0.001629**	**0.8585**
60	21.6281	0.006874	0.5273	22.6338	0.005453	0.5785	23.7495	0.004217	0.6358	**26.9452**	**0.002021**	**0.8320**
70	20.3576	0.009210	0.4624	21.6339	0.006865	0.5196	22.7007	0.005369	0.5766	**26.1415**	**0.002431**	**0.8057**
80	19.2434	0.011903	0.4069	20.7259	0.008461	0.4669	21.7584	0.006670	0.5221	**25.4331**	**0.002862**	**0.7796**
90	18.2538	0.014949	0.3597	19.9004	0.010232	0.4207	20.9137	0.008103	0.4733	**24.8001**	**0.003311**	**0.7541**
Boat Image												
10	30.7026	0.000851	0.9208	29.9916	0.001002	0.8925	**32.4293**	**0.000572**	0.9402	31.4759	0.000712	0.9320
20	28.1594	0.001528	0.8556	27.0997	0.001317	0.8762	28.8032	0.001317	0.8762	**29.3409**	**0.001164**	**0.8935**
30	26.0124	0.002505	0.7750	25.2269	0.003001	0.7441	26.7378	0.002119	0.8097	27.8329	0.001647	**0.8578**
40	24.1782	0.003821	0.6919	23.8144	0.004155	0.6724	25.1766	0.003036	0.7422	26.6872	0.002144	**0.8253**
50	22.6114	0.005481	0.6138	22.6575	0.005423	0.6050	23.9009	0.004073	0.6769	25.7502	0.002661	**0.7948**
60	21.2535	0.007493	0.5434	21.6454	0.006846	0.5428	22.8124	0.005233	0.6154	24.9365	0.003209	**0.7653**
70	20.0659	0.009849	0.4821	20.7611	0.008392	0.4871	21.8690	0.006503	0.5591	24.2396	0.003767	**0.7373**
80	19.0109	0.012558	0.4290	19.9742	0.010060	0.4389	21.0205	0.007906	0.5075	23.6396	0.004326	**0.7108**
90	18.0658	0.015611	0.3833	19.2500	0.011885	0.3964	20.2594	0.009420	0.4615	23.1187	0.004877	**0.6856**

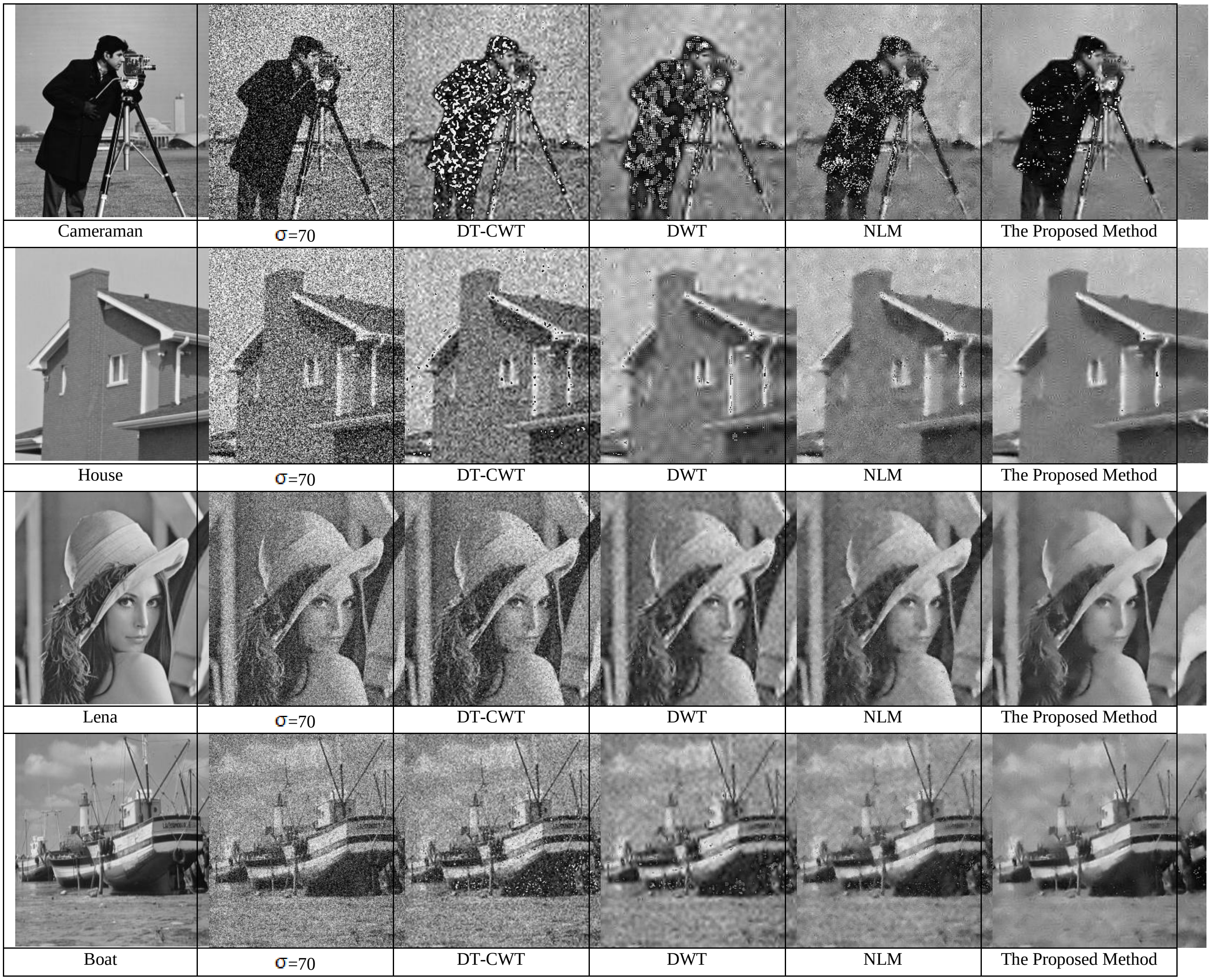


Fig. 4. The comparison denoising results.

V. Conclusion

Noise shows the image quality that has begun to lose detail, where large numbers of dots will appear in the image. Noise appearance dramatically affects an image's sharpness, clarity, and quality. Therefore, noise within an image must be appropriately addressed to minimize noise while maintaining fine image details, such as edges and textures. This paper presents the denoising method to handle such issues. The proposed denoising method utilizes the efficacy of wavelet- and NLM-based denoising. In the proposed denoising method, the noisy image is first decomposed using the DT-CWT, followed by DWT, to obtain the low- and high-frequency sub-bands. The high-frequency sub-bands are then threshold in the output of DWT to eliminate high-frequency noise using hard thresholding with cycle spinning. Meanwhile, the NLM removes the low-frequency noise. When the proposed method is applied to Lena, the boat, the house, and the cameraman images with AWGN variance noise of 10 to 90 in increment 10, the effectiveness outperforms the conventional DT-CWT, discrete wavelet transforms, and NLM. This superiority is numerically analyzed using three evaluation metrics: SSIM, PSNR, and MSE, also including when analyzed based on its visual quality results.

Future improvement is needed to increase the capability of the proposed denoising method, such as by selecting the mother wavelet and decomposition level, NLM parameter setting, and experimenting with another image to know the capability of the proposed denoising method in a different case.

References

[1] M. M. P. nik and S. V. H. se, “A Review Paper: Study of Various Types of Noises in Digital Images,” Int. J. Eng. Trends Technol., vol. 57, no. 1, pp. 40–43, 2018, doi: 10.14445/22315381/ijett-v57p208.

[2] Y. Li, C. Liu, X. You, and J. Liu, “A Single-Image Noise Estimation Algorithm Based on Pixel-Level Low-Rank Low-Texture Patch and

Principal Component Analysis," Sensors, 2022, doi: https://doi.org/10.3390/s22228899.

[3] S. Rani, Y. Chabrra, and K. Malik, "An Improved Denoising Algorithm for Removing Noise in Color Images," Eng. Technol. Appl. Sci. Res., vol. 12, no. 3, pp. 8738–8744, 2022, doi: 10.48084/etasr.4952.

[4] R. Al-Shamasneh and R. W. Ibrahim, "Image Denoising Based on Quantum Calculus of Local Fractional Entropy," Symmetry (Basel)., vol. 15, no. 2, p. 396, Feb. 2023, doi: 10.3390/sym15020396.

[5] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," Vis. Comput. Ind. Biomed. Art, vol. 2, no. 1, p. 7, Dec. 2019, doi: 10.1186/s42492-019-0016-7.

[6] H. Hu, B. Li, and Q. Liu, "Removing Mixture of Gaussian and Impulse Noise by Patch-Based Weighted Means," J. Sci. Comput., vol. 67, no. 1, pp. 103–129, 2016, doi: 10.1007/s10915-015-0073-9.

[7] P. Singh, G. Pradhan, and S. Shahnawazuddin, "Denoising of ECG signal by non-local estimation of approximation coefficients in DWT," Biocybern. Biomed. Eng., vol. 37, no. 3, pp. 599–610, 2017, doi: 10.1016/j.bbe.2017.06.001.

[8] R. Mohan et al., "Improved Procedure for Multi-Focus Images Using Image Fusion with qshiftN DTCWT and MPCA in Laplacian Pyramid Domain," Appl. Sci., vol. 12, no. 19, p. 9495, 2022, doi: 10.3390/app12199495.

[9] J. Zhang, "Research on Image Nonlocal Denoising Algorithm based on Wavelet Decomposition," Int. J. Signal Process. Image Process. Pattern Recognit., vol. 8, no. 9, pp. 353–362, 2015.

[10] R. R. Coifman and D. L. Donoho, "Translation-Invariant De-Noising," Lect. Notes Stat. Wavelets Stat., 1995.

[11] Z. H. Shamsi and D. Kim, "Multiscale Hybrid Non-local Means Filtering Using Modified Similarity Measure," pp. 1–7, 2011.

[12] H. Rabbouch and F. Saâdaoui, "A wavelet-assisted subband denoising for tomographic image reconstruction," J. Vis. Commun. Image Represent., vol. 55, pp. 115–130, 2018, doi: 10.1016/j.jvcir.2018.05.004.

[13] N. P. Raj and T. Venkateswarlu, "Denoising of MR images using adaptive multiresolution subband mixing," 2013 IEEE Int. Conf. Comput. Intell. Comput. Res. IEEE ICCIC 2013, 2013, doi: 10.1109/ICCIC.2013.6724247.

[14] N. K. Al-Qazzaz, S. H. Bin Mohd Ali, S. A. Ahmad, M. S. Islam, and J. Escudero, "Selection of mother wavelet functions for multi-channel EEG signal analysis during a working memory task," Sensors (Switzerland), vol. 15, no. 11, pp. 29015–29035, 2015, doi: 10.3390/s151129015.

[15] M. Srivastava, C. L. Anderson, and J. H. Freed, "A New Wavelet Denoising Method for Selecting Decomposition Levels and Noise Thresholds," IEEE Access, vol. 4, no. 1, pp. 3862–3877, 2016, doi: 10.1109/ACCESS.2016.2587581.

[16] W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury, "The dual-tree complex wavelet transform," IEEE Signal Process. Mag., vol. 22, no. 6, pp. 123–151, 2005, doi: 10.1109/MSP.2005.1550194.

[17] R. T. Ogden, Essential Wavelets for Statistical Applications and Data Analysis. 1997.

Advanced Detections of Norway Lobster (Nephrops Norvegicus) Burrows using Deep Learning Techniques

Atif Naseer
Science and Technology Unit, Umm al Qura University, Makkah, Saudi Arabia
ETSI Telecomunicación, Universidad de Málaga, Málaga, 29071, Spain

***Abstract*—Marine experts are facing lot of challenges in habitat monitoring of marine species. One of the biggest challenges is the underwater environment and species movement. The other challenge is the data collection of marine species. People used the camera sensors and satellite data in the past for data collection but in this era the scientists are using underwater Autonomous Underwater Vehicles (AUVs), the Remotely Operated Vehicles (ROVs), and certain sledges with high-definition still and video cameras to record the underwater footages. The ocean is composed of thousands of species which make the environment more challenging to monitor any specific specie. This work will focus on specie named Norway lobster (*Nephrops* norvegicus). The *Nephrops* norvegicus is one of the commercial specie in the Europe and generates millions of dollars yearly. This specie lives under the seabed and leaves behind the burrow structure on the sea ground. The *Nephrops* spend most of their time under the seabed. The scientists are currently monitoring the habitat of *Nephrops* norvegicus by underwater television (UWTV) surveys that is collected yearly on many European grounds. The collected data is reviewed manually by the experts who count the burrows on the sheet. This work focuses on the automatic detection of *Nephrops* burrows from underwater videos using the deep learning techniques. This work trained the Faster R-CNN models Inceptionv2, MobileNetv2, ResNet50, and ResNet101. Instead of training the models from scratch we used the transfer learning technique to fine tune these networks. The data is obtained from the Gulf of Cadiz (FU30) station. Twenty-eight different set of experiments are performed. The models are evaluated quantitatively using the mean Average Precision (mAP), precision and recall curves. Also, the models are qualitatively analyzed by visually presenting the output. The results prove that deep learning techniques are very helpful for marine scientists to assess the *Nephrops* norvegicus abundance.**

***Keywords*—Nephrops norvegicus; deep learning; stock assessment; faster RCNN**

I. Introduction

Marine ecosystems include the open, deep oceans and marine species. The environment has high level of dissolved salts. Marine ecosystem is one of the main sources of our daily food. The marine species have different physical and biological characteristics. Coral reef is a good example of marine ecosystem that is associated with other marine life like fishes and turtles. The oceans cover 70% of our planet, so the marine ecosystem covers most of our earth. As compared to the terrestrial ecosystem, the marine ecosystem is very challenging to study. Most of the challenges come due to the complex medium of sea. The environment of marine ecosystem has certain challenges like color variations, species movement, and turbidity [1]. Marine scientists are monitoring the environment from decades by collecting underwater species images using satellite, shipborne and camera. With the advancement of technologies, several new techniques like ROVs and AUVs are used by the scientists to record the images and videos of marine ecosystem. The scientists are still facing many challenges in the sea due to the illumination, views, variation in the lighting conditions and free natural environment [2].

Marine ecosystems have thousands of underwater species. Out of these species one of the important specie in Europe is Nephrops norvegicus (a Norway Lobster). This specie is considered as a commercial specie in Europe. This specie supports Europe with almost 60,000 t [3] and an income of 300 million € per year approximately [4].

The International Council for the Exploration of the Sea (ICES) is a marine science organization that leads the scientific forums on all domains of marine sciences. Their major goal is to advance the marine ecosystem. They provide state-of-the-art goals and facilities that help the scientists to do research in the marine eco system. There are many working groups under the umbrella of ICES that are conducting annual survey and monitoring the habitat of marine species. One of the major groups for Nephrops habitat monitoring is Working Group on Nephrops Surveys (WGNEPS), formerly known as the Study Group on Nephrops Surveys (SGNEPS). The aim of this group is to provide international coordination for Nephrops UWTV and trawl surveys in the North Atlantic. Each year the WGNEPS conducted a UWTV and trawl survey to assess the population of Nephrops. Nephrops populations are assessed and managed by Functional Units (FU) where there is a specific survey for each FU. Fig. 1 shows an individual Nephrops. Nephrops norvegicus lives in the sandy-muddy sediments and create burrows in the seabed [5]. An individual Nephrops specimen ranges in length of 2 – 5.5 cm with a maximum length of 24.0 cm. The most common length is about 19.0 cm [6].

A special equipment is used in the survey for data collection. Every year the UnderWater TeleVision (UWTV) and Trawl surveys are conducted all over the Europe by WGNEPS to estimate the abundance of Nephrops norvegicus specie. The surveys are used to provide population estimates for Nephrops based on Functional Units (FU). The survey data is stored in disks in the form of high-definition images and videos. The data is analyzed manually using the TV survey to classify and count the Nephrops burrows. Currently, the Nephrops data are collected through the yearly UWTV surveys and are reviewed by the marine experts manually. This manual process is very time consuming and leads to many errors due to environmental complexity and data variation. In this work, we are using the data obtained from the Gulf of Cadiz (FU30) station.

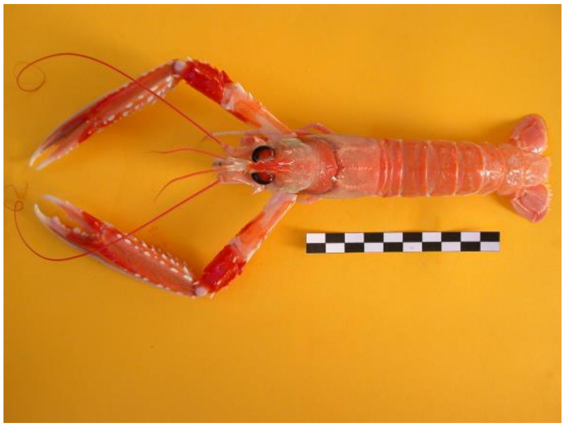


Fig. 1. Nephrops norvegicus.

Artificial Intelligence (AI) is an emerging field that solves many object detection problems including underwater species classification and detections. However, to detect and classify the Nephrops burrows for habitat monitoring, literature is unable to provide many solutions. One of the main reasons is the unavailability of Nephrops survey data. The complexity of data is also one of the reasons. This thesis is an effort to automate the existing method of Nephrops counting.

In our previous work [7] we trained and tested the MobileNet and Inception model and compare their results. In this work, we used denser Faster R-CNN models for training. We used the transfer learning technique and fine tune the MobileNetv2, Inceptionv2, ResNet50 and ResNet101 with FU30 dataset. The results obtained the good level of accuracy. ResNet101 obtained the highest level of accuracy among the other models.

The rest of the paper is organized as follows. Problem statement and definition is defined in Section II, the methodology is presented in Section III followed by the experiments and results in Section IV. The paper is concluded in Section V.

II. Problem Statement and Definition

We describe the problem of detections of Nephrops burrows in videos as currently the burrows are counting manually using the TV surveys. In this work, we demonstrate that the deep learning techniques will help to automatically detect and count the Nephrops burrows. Before going into the details, it is important to define the Nephrops burrows and their pattern.

Definition: A Nephrops burrow is an opening with following signature features: The burrow opening is like a half-moon shaped. The opening has proof of expelled sediments that creates scratches and tracks on the burrow opening. The pattern of burrows makes them unique as compared to other species burrows.

III. Proposed Methodology

This section discusses the approach for automatically identifying the Nephrops burrows from the video sequences using the deep learning techniques. The proposed methodology is illustrated in Fig. 2. The first part of the work is to collect the data from different stations of UWTV survey. In this study we are using the data collected from the Gulf of Cadiz (FU30) station. Specialized equipment is used for data collection at FU30. The second part of the methodology is the preprocessing of data that includes data cleaning, image annotations and data preparation. The third part is the detection of Nephrops burrows by applying the deep learning techniques. The deep learning models are trained and tested on the different datasets. We used transfer learning and fine tune the Faster R-CNN algorithms for model training.

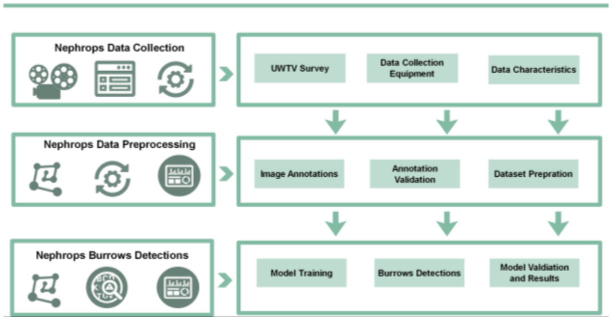


Fig. 2. Proposed methodology for Nephrops burrows detections.

A. *Data Collection*

In this research we used the data from the Gulf of Cadiz (FU30) station. To observe the habitat of *Nephrops* at FU30 station, a survey is designed yearly to collect the data. The survey used specially designed equipment for data collection.

1) Data collection equipment: A special sledge is designed in the survey of FU30 station. The sledge is equipped with Sony FDRAX33 camera that is used to capture the videos in high quality. The camera is mounted at an angle of 45 degree. Two laser lights are used in the sledge that define the field of view. The field of view is set to 75 cm. Fig. 3. shows the inner view of the sledge used in FU30 station survey.

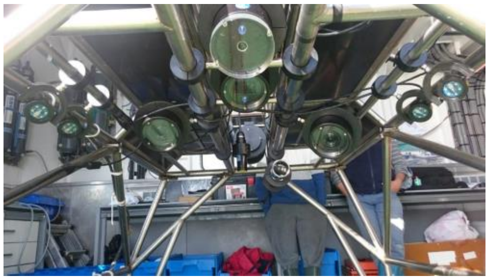


Fig. 3. Sledge inner view used in FU30 survey.

2) Data collection procedure: In 2018, the survey at FU30 is conducted at 70 different stations. The station is defined as a geostatistical location in the sea where the *Nephrops* density is assumed to be present and estimated in the past. The sledges are placed on a big ship and at each station it is dropped down in the sea and deployed to the sea ground. To maintain a constant speed of sledge, it is towed between 0.6–0.7 knots. This condition of sledges is the best possible condition for *Nephrops* burrows counting. The sledge is mounted with high-definition video camera that records the video footage of 10–12 min at 25 frames per seconds. The area covered by the sledge during this video footage is around 200m approximately. The sledge position is recorded after every 1 to 2 seconds for calibration. Two laser lights are placed that confirmed the field of view of the video footage. The field of view is set to be 75cm and the sledge distance over ground (DOG) is estimated from the position of the sledge. Fig. 4. shows the sample image collected during the survey.

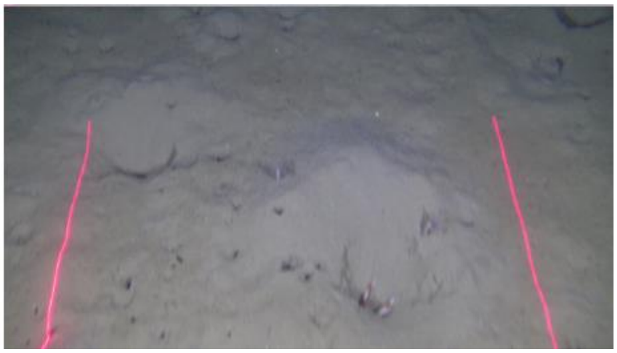


Fig. 4. FU30 Sample image.

3) Data characteristics: At FU 30, Video footages of 10-12 minutes has been recorded at 25 frames per second in good lighting condition. The video is recorded with a resolution of 3840 x 2160. We got the data of FU30 from the UWTV survey of 2018. A total of 70 stations are surveyed. After evaluating all the stations carefully, we only choose seven stations for annotations. The stations are selected based on their higher contrast, good video quality, lightning conditions, and high burrows density rates. The recorded videos are saved to the disks for manual *Nephrops* counting.

B. Data Preprocessing

The data preprocessing is one of the important phases of our methodology. Without performing the preprocessing step, the data comes with lot of noise and error that can affect the accuracy of the model training. The preprocessing is defined as cleaning of data, annotating the images and validation and preparation of dataset.

1) Data cleaning: The collected data is converted into frames and each frame is observed carefully. The frames with poor lighting and visibility are discarded initially. Also, the frames with zero burrows density are discarded before the annotations.

2) Image annotations: Image annotation is the most critical part of this study. Before the annotation, a comprehensive training is required to understand the burrow's characteristics. Certain protocol is followed in manual counting that should be observed during the manual image annotations. For image annotation, we used the Microsoft VoTT [8] tool. The *Nephrops* burrow is annotated by drawing the bounding box around it. The annotations are saved in the Pascal VOC format.

3) Annotation validations: Each annotation is validated by the Marine experts from the Gulf of Cadiz. The final annotations for model training are obtained after several iterations of validations. The validated annotations are saved in the XML format.

4) Dataset preparation: The annotated image dataset is divided into training and testing data separately. Each station consists of around 15,000 - 18,000 frames. From 2018 survey, almost 105,000 frames were recorded from seven different stations. The training and testing dataset is divided to 80-20 ratio respectively. Table I shows the training and testing dataset distribution used in this study.

TABLE I. DATASET DISTRIBUTION

***Nephrops* Dataset Distribution**			
Functional Unit	**Training Images**	**Testing Images**	**Total Images**
FU30	200	48	248

C. Nephrops Burrows Detection

AI plays an important role nowadays in automating the analysis. In marine sciences many scientists apply AI techniques to monitor the habitats of marine species. Computer Vision and Deep learning shows a significant improvement in the object detection [9,10], classification [11,12], and segmentation [13].

1) Model training: To train the models, the transfer learning [14] technique is utilized to fine-tune the Faster R-CNN Inceptionv2 [15], MobileNetv2 [16], ResNet50[17], and ResNet101[18] models in TensorFlow [19].

Inception networks are considered as one of the big milestones in CNN. For computational complexity, the Inception v2 network used smart factorization method with 5x5 convolution and two 3x3 convolution. We fine tune the network parameters with a learning rate of 0.01 and a batch size of 1. The value of Maxpool stride and Maxpool kernel size are set to 2. The gradient clipping [20] value if set too low or too high will result in the model instability so we set its threshold value to 2.0. For activation function we used the Softmax and Mask RCNN is used as a box predictor.

MobileNetv2 architecture is used with the relatively small dataset. MobileNetv2 architecture used a depth-wise separable convolution instead of conventional convolution. The architecture of this network is composed of 32 convolutional layers and 17 residual bottleneck layers. We fine tune the certain parameters of the network to get the optimize results. The learning rate is set to 0.01. The batch size is set to 24 with truncated normal initializer. For activation function we used Rectified Linear Unit (ReLU) and Convolutional box predictor is used as a box predictor.

ResNet50 [17] is a variant of the ResNet model. The ResNet50 is 50-layers deep convolutional network. Out of these 50 layers, 48 are convolutional layers, one max pool, and one average pool layer. In the first convolution only one layer is used with a kernel size of 7 × 7,64 kernels with stride 2 and a max pool of size 3 × 3. In the second convolution, nine layers are used with a kernel size of 1 × 1,64; 3 × 3,128. In the third convolution, 12 more layers are used with 1 × 1,128; 3 × 3,128, and 1 × 1,512 kernel. The fourth convolution uses 18 more layers with kernel sizes of 1 × 1,256 ;3 × 3,256 and 1 × 1,1024. Nine layers are used in the fifth convolution with kernel sizes of 1 × 1,512; 3 × 3,512 and 1 × 1,2048. Finally, the last convolution layer is used for avg pool and a Softmax function. We set the learning rate of 0.003 with batch size 1 and L2 regularization. The Softmax is used as an activation function and Mask RCNN is used as a box predictor.

The ResNet101 [18] is a considered to be 101 layers dense convolutional neural network. The first convolution layer has a kernel size of 7 × 7,64 with stride 2 and a max pool of size 3 × 3. The second convolutional layer used nine layers with kernel sizes of 1 × 1,64 and 3 × 3,28. The third convolutional layer used 1 × 1,128; 3 × 3,128, 1 × 1,512 kernels. Sixty-nine layers are used in the fourth convolutional layer with following kernels 1 × 1,256; 3 × 3,256 and 1 × 1,1024. The fifth convolution uses 9 layers with 1 × 1,512; 3 × 3,512 and 1 × 1,2048. Finally, the last convolutional layer is used for avg pool and a Softmax function. The learning rate is set to 0.0003 with 24 batch size and L2 regularization. The Maxpool kernel size and Maxpool stride is set to 2 and Mask RCNN is used as a box predictor.

Table II shows the parameter list and their values used in the Inceptionv2, MobileNetv2, ResNet50 and ResNet101 model.

2) Model validation: The models used in this study for training used approximately 80% of the annotated dataset. The remaining 20% is used for the testing. All the models are trained over 70k iterations and the turning checkpoints during the training is recorded after every 10k iterations. For validation of models the models are evaluated using mAP, precision and recall curve and through visual inspection of the output images with detections.

TABLE II. FASTER R-CNN MODELS TRAINING PARAMETERS

Parameters	**Inceptionv2**	**MobileNetv2**	**ResNet50**	**ResNet101**
Number of Classes	01	01	01	01
Optimizer	Momentum	RMSProp	Momentum	Momentum
Momentum Rate	0.9	0.9	0.9	0.9
Learning Rate	0.01	0.01	0.0003	0.0003
Batch Size	1	24	1	24
Initializer	truncated_normal_ initializer	truncated_normal_initializer	truncated_normal_initializer	truncated_normal_initializer
gradient_clipping_by_norm	10	-	10	10
Regularization	L2	L2	L2	L2
Activation Function	Softmax	RELU	Softmax	Softmax
Maxpool kernel size	2	-	2	2
Maxpool stride	2	-	2	2
Box Predictor	Mask RCNN box predictor	Convolutional box predictor	Mask RCNN box predictor	Mask RCNN box predictor

IV. EXPERIMENTS AND RESULTS

1) Experiments: In this section we will summarizes all the experiments and results performed to automatically detect the *Nephrops* burrows. The models are trained and tested with FU30 dataset. Four different combinations of set of experiments are performed with the current dataset. Each set of experiment is iterated 7 times hence a total of 28 experiments are performed.

2) Results

a) Quantitative analysis: The mAP of all the models trained and tested by FU 30 stations are calculated during the quantitative analysis. In this study we trained the models with 70k iterations. The model performance is recorded in terms of precision and recall values after every 10k iterations. The precision is the prediction accuracy measurement and recall are the measurement of positive predictions. The mAP is calculated for each model which is a very common metric to evaluate the performance of object detection algorithms. The mAP is defined in the in Eq. (1).

$$mAP = \int_0^1 P(R)dR \qquad (1)$$

Precision can be seen as how robustly the model identifies *Nephrops* burrows' presence, and Recall is the rate of True Positive (TP) over the total number positives detected by the model [21]. The precision and recall curves are used to measure the model behavior.

In our study, the ground truth annotations and model predictions are rectangular boxes that usually don't fit perfectly. In this paper, the detections are considered as TP if the detection and ground truth overlap more than 50%. This is calculated by the Jaccard index J, as defined in Eq. (2).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \qquad (2)$$

Here, A and B are the set of pixels in the ground truth annotation and model predictions respectively, and | . | means the number of pixels in the set. If J ≥ 0.5, a TP is detected, but if J < 0.5, detection fails with a False Negative (FN). This methodology is used to calculate the precision and recall values.

Table III shows the maximum mAP obtained by MobileNetv2, Inceptionv2, ResNet50, and ResNet101 models, respectively. The maximum mAP obtained using the MobileNet model is 65.69. The maximum mAP obtained by the Inception model is 77.18. The ResNet50 and ResNet101 models achieve better precision values as compared to MobileNet and Inception. The maximum mAP in ResNet50 is 80.16 while in ResNet101 it is 81.59.

The results are also presented in the form of precision and recall curves. Fig. 5 shows the results obtained with the models trained and tested by FU 30 dataset. The best mAP is 81.59 with ResNet101 model.

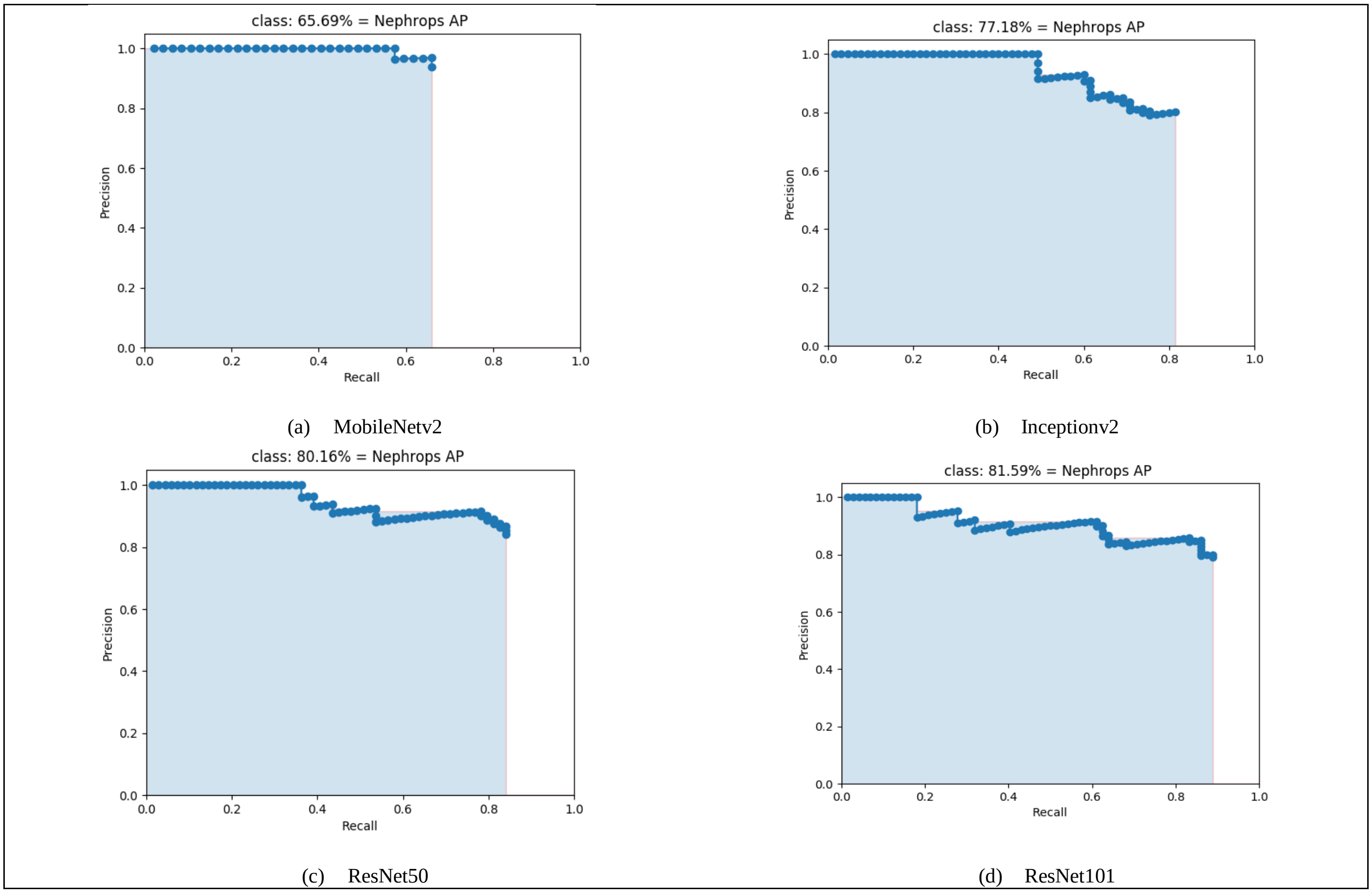
b) Qualitative analysis: In this section, the performance of different models on the dataset is analyzed qualitatively. The visualization results are from the MobileNetv2, Inceptionv2, ResNet50, and ResNet101 models that are trained and tested by the FU30 dataset.

Fig. 6 shows the Nephrops burrows detections visually using the MobileNetv2, Inceptionv2. ResNet50, and ResNet101 models with FU 30 dataset. The green rectangular boxes on the images shown are the TP detections by the model. The blue bounding boxes are the actual ground truth annotations while, the red bounding boxes are the False Positive (FP) detections that are detected by the trained models. In this example, MobileNetv2 model detects one TP burrow while the inception and all other models correctly detects the two TP Nephrops burrows.

The overall study shows that the ResNet101 model performs better in terms of mAP and provides an accuracy of more than 80%.

TABLE III. mAP OBTAINED USING MULTIPLE TRAINING MODELS.

mAP obtained with multiple Training models	
Trained Model	mAP
MobileNetv2	65.69
Inceptionv2	77.18
ResNet50	80.16
ResNet101	81.59



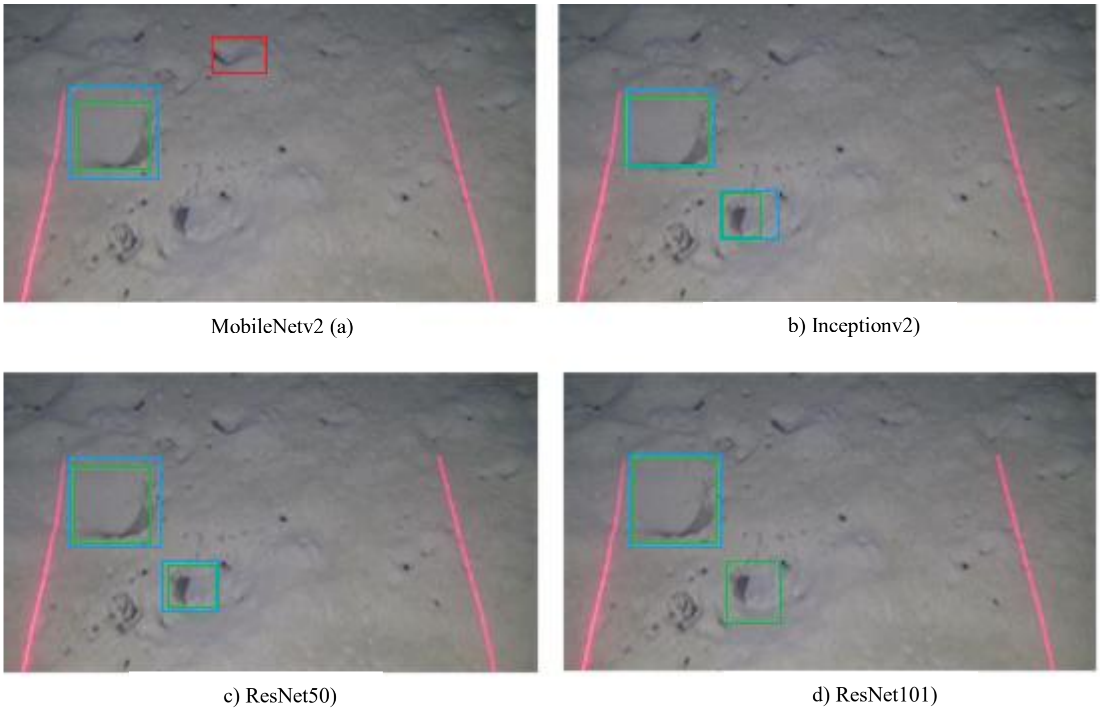
(a) MobileNetv2

(b) Inceptionv2

(c) ResNet50

(d) ResNet101

Fig. 5. Precision-recall curve obtained using FU 30 dataset (a) PR-curve of MobileNet, (b) PR-curve of Inception, (c) PR-curve of ResNet50, (d) PR-curve of ResNet101.



MobileNetv2 (a)

b) Inceptionv2)

c) ResNet50)

d) ResNet101)

Fig. 6. Nephrops burrows detection with FU30 dataset (a) Detection with MobileNet model, (b) Detection with Inception model, (c) Detection with ResNet50 model, (d) Detection with ResNet101 model.

V. Conclusion and Future Work

The aim of this study is to automatically detect and classify the Nephrops norvegicus burrows from underwater videos. We used the dataset from the Gulf of Cadiz (FU30) survey in 2018. We trained four different Faster R-CNN models to study the detection of Nephrops burrows using deep learning. The results show that the deep learning algorithms are very effective in detecting the burrows automatically. The ResNet101 model performs better and achieves the mAP of 81.59. This practice helps the marine scientists to correctly estimate the abundance of Nephrops from the underwater videos. The automatic detection algorithms could replace the manual counting process of marine experts and provide an accurate count in very less time.

In future work, we will plan to use a bigger curated dataset from different stations working under the ICES. The more data will improve the performance of accuracy of the deep learning models. Also, the newer model based on YOLO architecture will be trained in the future work. Finally, we will plan to integrate the spatial and temporal information of the Nephrops burrows to estimate the burrow sizes and their complexes.

Acknowledgment

We thank the Spanish Oceanographic Institute, Cádiz, Spain for providing the dataset for research.

References

[1] T. Rimavicius and A. Gelzinis, “A comparison of the deep learning methods for solving seafloor image classification task,” Communications in Computer and Information Science, vol. 756, pp. 442–453, 2017.

[2] H. Qin, X. Li, Z. Yang and M. Shang, “When underwater imagery analysis meets deep learning: A solution at the age of big visual data,” in Proc OCEANS’15 MTS/IEEE, Washington, DC, USA, pp. 1–5, 2015.

[3] FAO (2021a). “FAO yearbook. fishery and aquaculture statistics 2019/FAO annuaire,” in Statistiques des pê ches et de l'aquaculture 2019/FAO anuario. estadısticas de pesca y acuicultura 2019´ (Rome/Roma: FAO).

[4] Issifu, I., Alava, J. J., Lam, V. W., and Sumaila, U. R. (2022). Impact of ocean warming, overfishing and mercury on European fisheries: A risk assessment and policy solution framework. Front. Mar. Sci. 8. doi: 10.3389/fmars.2021.770805.

[5] M. Jiménez, I. Sobrino and F. Ramos, “Objective methods for defining mixed-species trawl fisheries in spanish waters of the gulf of cádiz,” Fisheries Research, vol. 67, no. 2, pp. 195–206, 2004.

[6] Fischer, W., G. Bianchi and W.B. Scott 1981 Lobsters. 5: pag.var. In FAO Species identification sheets for fishery purposes. Eastern Central Atlantic (fishing areas 34, 47; in part). Canada Funds-in-Trust. Ottawa, Department of Fisheries and Oceans Canada, by arrangement with the Food and Agriculture Organization of the United Nations. 1-7.

[7] Naseer, A., Baro, E. N., Khan, S. D., Vila, Y., Doyle, J. (2022). Automatic Detection of *Nephrops* Norvegicus Burrows from Underwater Imagery Using Deep Learning. CMC-Computers, Materials & Continua, 70(3), 5321–5344.

[8] Microsoft CSE group. (2020, June 3), “Visual object tagging tool (VOTT), an electron app for building end to end object detection models from images and videos, v2.2.0. [Online]. Available: https://github.com/microsoft/VoTT.

[9] R. Girsshick, J. Donahue, T. Darrell and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142–158, 2016.

[10] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-cNN: Towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[11] R. Shima, H. Yunan, O. Fukuda, H. Okumura, K. Arai et al. “Object classification with deep convolutional neural network using spatial

information,” in Proc. Int. Conf. on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, pp. 135–139, 2017.

[12] S. Soltan, A. Oleinikov, M. Demirci and A. Shintemirov, “Deep learning-based object classification and position estimation pipeline for potential use in robotized pick-and-place operations,” Robotics, vol. 9, no. 3, 2020.

[13] S. Masubuchi, E. Watanabe, Y. Seo, S. Okazaki, K. Watanabe et al. “Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials,” npj 2D Mater Appl, vol. 4, no. 3, pp. 1–9, 2020.

[14] https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 2818–2826, 2016.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, “Mobilenetv2: inverted residuals and linear bottlenecks,” in Proc. Conf. on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 4510–4520, 2018.

[17] Understanding and Coding a ResNet in Keras. Available online: https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

[18] TensorFlow Core v2.8.0. Available online: https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet/ResNet101 (accessed on 20 March 2022).

[19] https://www.wgu.edu/blog/neural-networks-deep-learning-explained2003.html#close.

[20] R. Pascanu, T. Mikolov and Y. Bengio. “On the difficulty of training recurrent neural networks,” ArXiv Preprint, vol. 1211, 5063, pp. 1–12, 2012. [Online]. Available: https://arxiv.org/pdf/1211.5063.pdf.

[21] M. Everingham, L. Van Gool, Cl. Williams, J. Winn and A. Zisserman, "The pascal Visual Object Classes (VOC) challenge," International Journal of Computer Vision, vol 88, pp. 303-338, 2010.

Implementation of a Smarter Herbal Medication Delivery System Employing an AI-Powered Chatbot

Maria Concepcion S. Vera[1], Thelma D. Palaoag[2]

College of Information and Communications Technology, Catanduanes State University, Catanduanes, Philippines[1]
College of Information Technology and Computer Science, University of the Cordilleras, Baguio City, Philippines[2]

***Abstract*—Medicinal plants are a practical and cost-effective alternative for treating common ailments, especially in areas with limited access to public healthcare systems. This paper introduces a prototype of an intelligent interactive system that merges chatbot technology with artificial intelligence (AI) to address inquiries related to treatment alternatives and the application of different medicinal plants for prevalent health conditions which promote and advance alternative healing practices in the locality. The platform is a hybrid online chat service that prioritizes consumer health and encourages the responsible use of medicinal plants. This study used a survey questionnaire to gather information from traditional healers and users and concerned government agencies about how well the system prototype performed. The system's performance was assessed in terms of effectiveness, efficiency, and customer satisfaction, with respondents providing an aggregate rating of "Strongly Agree". Significantly, this study lays the groundwork for education on the use of local medicinal plants to cure illnesses and highlights the importance of providing users with accurate and reliable information on the safe use of medicinal plants. This approach empowers users to make informed decisions about the plants they use, reducing the likelihood of harmful effects and optimizing the potential benefits of medicinal plants. By supporting this effort, this study contributes to the achievement of the third Sustainable Development Goal of the UN, which aims to promote health and well-being by offering the local populace a low-cost option as a first line of defense for improving their health and wellness.**

Keywords—Chatbot; artificial intelligence; intelligent interactive system; applied computing; consumer health; medicinal plants; traditional medicine

I. INTRODUCTION

Individuals who have limited access to the nation’s healthcare system frequently rely on herbal cures and medicines that treat a wide variety of illnesses and diseases. The World Health Organization (WHO) estimates that 21,000 pharmacological flowering plants have medicinal potential, which represents more than 30% of all plant species [1].

The Philippine TAMA Law of 1997 or the Traditional and Alternative Medicine Act represents a legitimate justification for highlighting the significance of traditional and complementary medicine for people's insurance. It was established to enhance the quality and health care services provided to the Filipino populace. The Philippine Institute of Traditional and Health Care (PITAHC) has been tasked with implementing the law whose primary purpose is to promote alternative and traditional medical items, methods, and techniques that are effective, safe, efficient, and readily available [2].

The Traditional Knowledge Digital Library (TKDL) website in the Philippines was created to ease the execution of the law. The use of traditional medicine in customs and activities associated with healing can be researched and examined by readers. The TKDL website lists 864 medicinal plants with 3,737 different medical uses [3]. Moreover, a group of researchers in the province of Catanduanes made available recorded data on specific plant species that are effective for treating a wide range of common ailments; the procedures used by healing practices while employing medicinal plants; and how the indigenous people's wisdom and experience shaped the plant medicines’ formulation and its usage, and how they observed their effects on their patients [4][5]. Whilst there is significance in using herbal treatments in medicine, there isn't a publicly available internet database that might give plant medicinal information in the form of a virtual conversation, making it challenging to spread and advance such information. Residents of Catanduanes' distant areas, who depend mostly on in-demand herbal medications from clinics, may have problems because of the absence of such a repository. Utilizing information and communications technology (ICT) may facilitate quick and accurate data analysis, effective decision-making, and an improved understanding of the requirements [6]. Knowing that folks prefer to communicate online, specifically on social media platforms like Facebook, developing an application addressing this issue would allow people in rural locations to be educated on plant healing techniques from the comfort of their own homes. ICT facilitates quick and easy access to knowledge, concepts, and experiences from numerous cultures, civilizations, and beliefs [7].

The use of Chatbot technology combined with AI can provide real-time access to information. Developing a chatbot with integrated AI can help provide humans with decision-making capabilities and web application experiences [8]. Using machine learning and natural language processing (NLP), chatbots facilitate human-to-human communication. In a range of fields, including medicine, agriculture, education, business, and e-commerce, NLP-based chatbots analyze and interpret real human language. Intelligent agents are capable of a variety of tasks, from straightforward physical work to intricate operations. One of the simplest and most popular types of intelligent human-computer interaction is a chatbot which is a basic illustration of an AI system. The use of chatbots in a diversity of businesses has been extensively researched in the past, but research on medicinal plants has not yet been conducted.

The aim of the present study is to bridge the gap and offer design considerations for developing a prototype system that can retrieve up-to-date information on medicinal plants and their applications, which can then be disseminated to people in underserved areas or in places without access to proper medical care.

This paper presents the development of MedPlantBot, a prototype system that integrates chatbot technology with AI. Its purpose is to provide a comprehensive database of various medicinal plants, including taxonomic classification, medicinal benefits, and categorization. The MedPlantBot framework model was carefully designed and built on established chatbot design techniques and AI frameworks, leveraging Google's Dialogflow to provide users with insightful guidance on appropriate medicinal plants for their specific health concerns. This user-friendly system is available round-the-clock and can be accessed from anywhere with internet connectivity, making it a convenient source of prompt and accurate information for both simple and complex queries. The evaluation of the Chatbot system prototype in this study utilized the Chatbot Usability Questionnaire (CUQ), a survey instrument that measured the chatbot’s performance. MedPlantBot has the potential to assist individuals in making informed decisions regarding their health and well-being.

This study aimed to develop an AI Chatbot model that could recommend medicinal plants for treating various ailments, illnesses, and diseases. The study involved identifying plant species with medicinal properties, analyzing the design techniques of existing Chatbots and AI Chatbot frameworks, and developing a framework model for the MedPlantBot Chatbot system. Additionally, the study evaluated the performance of the framework's prototype in terms of efficiency, effectiveness, and satisfaction. The development of MedPlantBot which shares information about using medicinal plants to treat illnesses is a game-changer in healthcare computing. This innovative technique harnesses the power of AI and NLP to offer users customized treatment programs that integrate traditional knowledge of medicinal plants. By merging the use of medicinal plants with modern healthcare practices, this chatbot presents an exciting opportunity to enhance healthcare outcomes and encourage a more comprehensive and sustainable approach to medicine. In essence, this study has the potential to provide residents with a cost-effective alternative to their initial healing practice, hence fostering better health and wellness. This coincides with the United Nations' third Sustainable Development Goal, which is Good Health and Well-Being.

The subsequent sections are organized as follows: Section II examines related works. Section III details the methodology employed by our proposed prototype system. Section IV contains the results and discussion. Section V summarizes our research findings while Section VI discusses the study’s ramifications and future undertakings.

II. Related Works

This section presents the analysis of the design methodologies of existing chatbots. This study discovered that creating a chatbot requires a range of strategies and approaches.

A. Chatbot Design Process

Building a chatbot begins with determining the scope and needs [9]. It emphasizes that two strategies such as conversation flow and knowledge modeling are important considerations while creating the chatbot. An adequate definition of the questions and answers must come first. The design of the chatbot, which is built using a decision tree algorithm that creates branches based on user input intents, is noteworthy. It simplifies the search procedure. Also, [10] constructed the MedBot using random forest, an ML decision tree algorithm, to make decisions. Significantly, this highly accurate approach minimizes uneven data.

Fig. 1 shows the general design of a chatbot system [11]. It is recommended to build a chatbot utilizing the aforesaid design, which comprises natural language and the user interface. backend, answer generation, natural language understanding (NLU), and dialogue management [12][10]. Ahmad et al. describes how a chatbot system works [13]. The user submits text using the chatbot's graphical interface. The text is then divided to find terms that may be used as patterns in pattern recognition. Finally, the system produces dynamic replies that are sent to the user. Many chatbots use the term "pattern matching" method that uses pattern recognition. Matching patterns are used to deliver valid replies to user inquiries founded on matching groups, such as plain statements, natural language, and semantic analysis and interpretation of the queries.

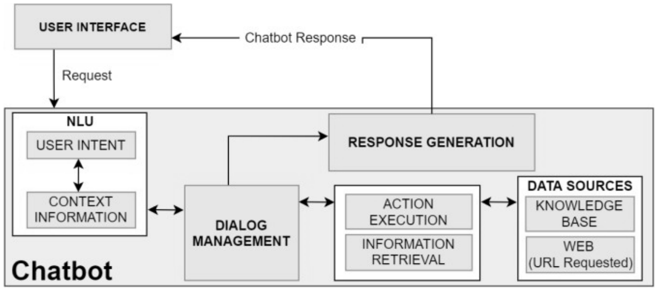


Fig. 1. Chatbot architecture.

Based on the Google Dialogflow (GDF) architecture and NLP, [12] created an AI Chatbot application that gives users access to information on COVID-19. The Chatbot agent combined a rule-based and a learning-based algorithm using the GDF framework design architecture to create a hybrid agent. The user interfaces for end users, natural language comprehension, and natural language creation make up its three (3) components. Similar work was done by Ranavare et al. who used GDF to create an NLP-based chatbot system. The purpose of the lesson was to inform the students about placement activities [14]. Furthermore, Muhammad et al. provides a thorough explanation for creating AI-enabled and speech-recognition chatbots that are utilized for English discussions where GDF acts as the AI’s engine [15]. Software developers leverage NLP-based GDF in constructing conversational user interfaces. GDF's APIs permit the incorporation of cognitive and AI services from other cloud providers. Facebook Messenger and other related services are already incorporated. Chatbots for popular messaging applications may be easily made by developers. Considerations for the design of intentions, entities, questions, and solutions must all be incorporated into a GDF-based intelligent chatbot. GDF provides knowledge base applications and sentiment classification [16]. According to the high-level design of Dialogflow, any channel technique that GDF provides can be used to connect a user to the platform. This platform converts real English into data that computers can comprehend by utilizing comparable sentences to build a machine-learning model. The GDF backend interacts with other APIs, databases, or services to complete the user's query [17]. The client interacts with the input device as shown in Fig. 2. The GDF engine accepts the query of the user. GDF then tries to determine the user's intent. Based on the intent, the fulfillment is conducted, and the data is sent from its source. The response is then processed into data that is usable. Eventually, the output device provided the user with the necessary information. The essential components of a chatbot are intents. To parse user data, intents' logic and elements are employed. This synchronizes the user's inputs with the result. In addition to contexts, Dialogflow supports entities, utterances, actions, training texts, and responses [9][18][19][15]. The intent describes each intended Chatbot action and the possible queries that could be used to complete the task. User requests are expressions or utterances that must be matched to the intent's questions. Entities are used to detect and extract meaningful information from natural language feeds. The inclusion of annotations to training words enables GDF to recognize generated data. Annotated group of words permits the specification of which variables to match and against which entities to match them. Once established in actions and parameters, tagged parameters are automatically applied.

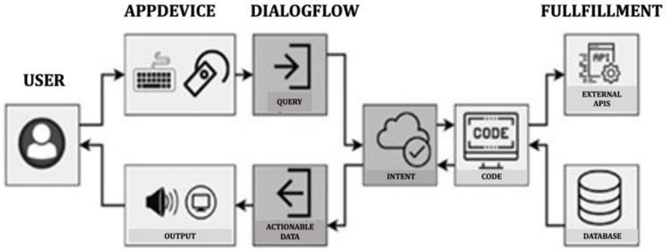


Fig. 2. Google dialog flow architecture.

B. *Evaluating AI Chatbot System Performance*

The system prototype is evaluated to determine whether it achieved the goal for which it was built. Furthermore, it enables design iteration before takeoff, which aids in the delivery of a successful and efficient product.

The AI Chatbot System's performance is evaluated based on three (3) metrics, namely, effectiveness, efficiency, and satisfaction [20]. Table I lists the quality characteristics arranged in accordance with ISO25000 based on an analysis of the evaluation instrument used in the existing chatbot system.

TABLE I. LIST OF QUALITY CHARACTERISTICS ARRANGED IN ACCORDANCE WITH ISO25000

Criteria	***Category***	***Quality Attribute***	***References***
Efficiency	Performance	• Degradation with grace • Adaptability to manipulation	• Adamopoulou & Moussiades, (2020) • Ahmad et al. (2018)
Effectiveness	Functionality	• Computers may interpret and modify natural expressions • Effective speec synthesis • Correctly translate instructions	• Adamopoulou & Moussiades, (2020) • Ahmad et al. (2018) • Daniel et al. (2021)
	Humanity	• Succeeds on the Turing test • Natural, convincing, and fulfilling interaction.	• Huang et al. (2021) • Karanja (2018)
Satisfaction	Affect	• Salutations and personality expression • Recognize the participant's mood and act accordingly • Make things mor enjoyable an captivating.	• Muhammad et al. (2020) • Neapolitan et al. (2018) • Ranavare et al. (2020)
	Ethics & Behavior	• The morals of users and their cultural awareness • Respect and safeguard privacy. • Nondeception	• Huang et al. (2021) • Sabharwal & Agrawal (2020) • Sánchez-Díaz et al. (2018)
	Accessibility	• Reacts to socia signals or lac thereof • Able to discer purpose c meaning	• Talan et al. (2022) • Vergadia (2022)

III. METHODOLOGY

This study established a process for building the system prototype from data gathering and collection of medicinal plants that exist in Catanduanes to identifying the technology and designing the system's framework as the basis for the development of the AI-based chatbot system grounded on the mixed-approaches strategy combining documentary analysis with qualitative and quantitative research methods. The methodology used in this study to construct the Chatbot framework was based on a review of [21], as shown in Fig. 3.

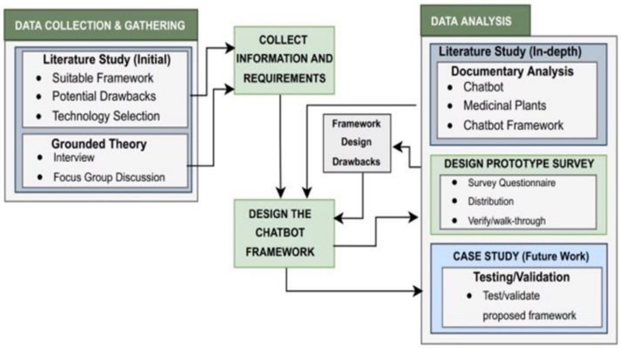


Fig. 3. Research methods for the development of the MedPlantBot Chatbot.

A. Data Collection and Gathering

An initial review of the existing literature, in-depth interviews, and focus group discussions was conducted to identify the needs and select the appropriate technology and framework design for the study. Participants of the study were chosen voluntarily, and it was ensured that they had research experience and a deep understanding of the subject under investigation. Consequently, between January 2022 and March 2022, participants were interviewed in the province of Catanduanes, including domain experts, users, and practitioners of medical plants, as well as representatives from relevant government agencies and researchers who studied medicinal plants.

B. Data Analysis

A comprehensive literature analysis was conducted to organize and assess the documentary evidence of different discoveries from studies on frameworks, chatbots, and herbal medicines. The researcher looked up and analyzed previous research on chatbots in several internet databases.

1) Medicinal plants data: Focus groups, interviews, and documentary analyses and assessments of past studies on the Catanduanes' medicinal plant study were undertaken by the researcher. There were 56 medicinal plants that have been identified [4]. The same study was carried out by [22]. The author provided a list of 115 medicinal plants utilized by Catanduanes' indigenous healers. From the list, ten (10) medicinal plants, such as Acapulco, Ampalaya, Bawang, Bayabas, Niyug-niyogan, Sambong, Tsaang-gubat, Ulasimang bato (pansit-pansitan) and Yerba Buena have been approved for usage by the Philippine Department of Health (DOH) and PITAHC [23]. Based on the information gathered, Catanduanes is home to all the medicinal plants endorsed by DOH and PITAHC.

Twenty-six (26) medicinal plants in Catanduanes were identified by the researchers. Limited research undertaken between 1998 and 2014 yielded the data, which included information on taxonomic classification, symptoms or diseases healed, and the preparation and application of the plants. Fig. 4 depicts the names of these plants: Ageratum Conyzoides, Areca Catechu, Capsicum Frustescens, Centella Asiatica, Cordia Dichotoma, Crinum Latifolium, Cyprus Halpan, Dicranopteris Linearis, Diplazium Esculentum, Drynaria Quercifolia, Duknay, Elephantopus Scaber, Emilia Sonchifolia, Euphorbia Hirta, Gliricidia Sepium, Hyptis Capitata, Imperata Cylindreica, Kolowratia Elegans, Ludwigia Micrantha, Lygodium Circinnatum, Lygodium Flexuosum, Musa Paradisiaca, Myristica Simiarum, Pariya, Solanum Torvum, and Stachytarpheta Jamaicensis.

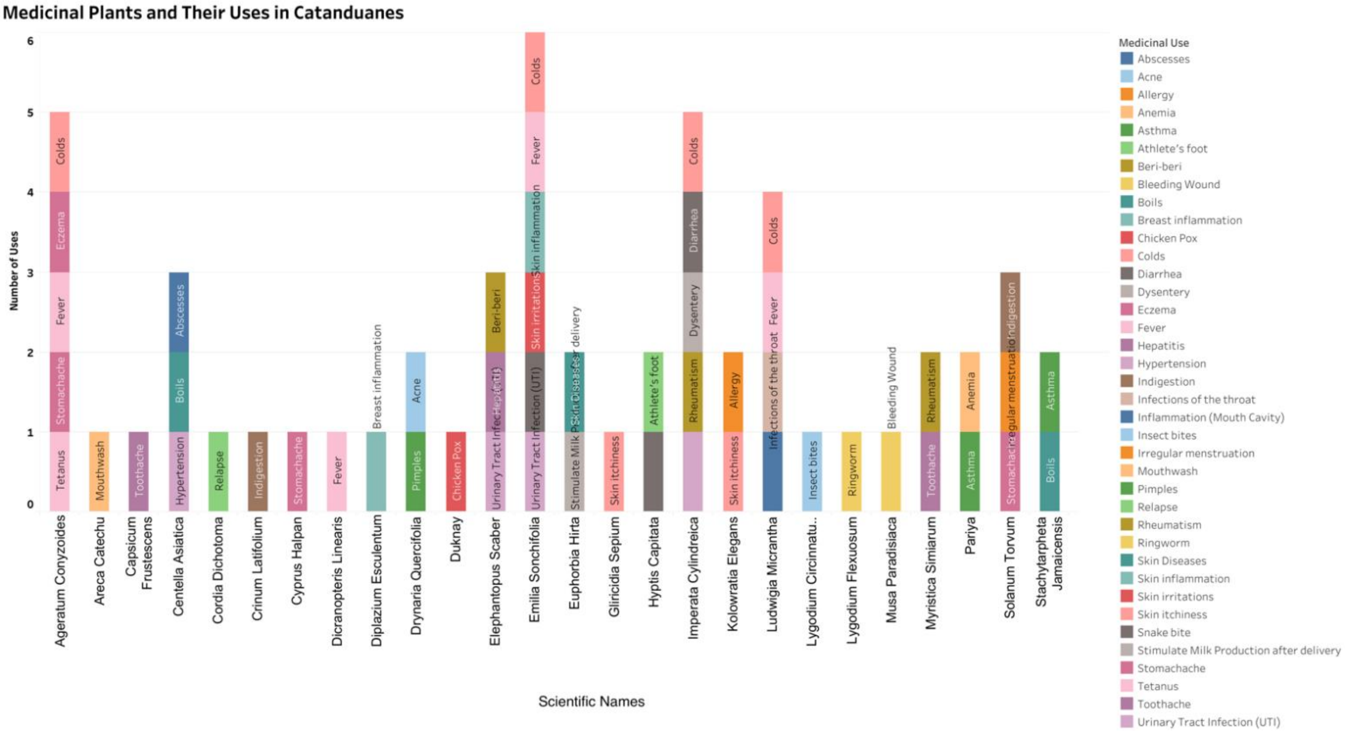


Fig. 4. Catanduanes medicinal plants and their application.

2) Proposed medplantbot system design architectural framework: After collecting the foundational data on medicinal plants, this study created training datasets, a knowledge base, and a conversational flow that revolves around the subjects and elements of a Chatbot based on the medicinal plant data from Catanduanes. The knowledge base offers a list of keywords, synonyms, and other phrases when conducting a keyword search. A group of knowledge bases consisting of MedPlantBot's frequently asked questions about medicinal plants is shown in Fig. 5. Each knowledge base contains medicinal plants profile, including their description, taxonomic classification, therapeutic uses, ailments cured, method of preparation, and application. The conversation flow for learning about medicinal plants from the MedPlantBot FAQ and knowledge base is shown in Fig. 6. It explains how to advance through the various tiers of the dialogue. The orange and green boxes reflect the user's input and MedPlantBot response, respectively. The flow of conversation directs communication while creating a chatbot. The connection between chatbots and humans remains flexible and personalized.

As indicated in Fig. 7, the overall serverless system design architecture for the MedPlanBot prototype system was then established. The framework comprises the system's architecture, which is designed on a serverless infrastructure for scalability and cost-effectiveness. The system employs multiple services, including Django, GDF, and Google Cloud Functions, to deliver a seamless and effective user experience. The design also includes natural NLP and machine learning (ML) algorithms to improve the chatbot's capacity to effectively comprehend and react to user queries. The created framework seeks to provide a user-friendly and reliable platform for accessing information about medicinal plants in Catanduanes. The MedPlantBot framework consists of multiple components that aim to offer a user experience that is smooth and efficient. These include (a) User Interface (UI). The UI is built using Django, a high-level python web application framework that enables the quick creation of websites. Users interact with the chatbot via text or voice through the UI; (b) GDF. The GDF serves as the main natural language understanding (NLU) and conversation engine. It receives user queries, identifies their intent, and retrieves relevant data from the data source. The response is processed and made usable before being sent to the output device to provide the user with the appropriate response. (c) Google Cloud Functions (GCF) is a serverless cloud-based core fulfillment solution that does not require server administration, operating system updates, or software configuration [24]. Its fully customizable chat widget provides consumers with live chat support. To connect a GDF-generated MedPlantBot to a website, obtain the account key from the GDF portal and then browse to the bot integration part of the chatbot platform, where the application id is provided.

The MedPlantBot framework employs NLU, AI, and serverless architecture to provide a customized chatbot experience. The seamless integration of these technologies facilitates user access to medicinal plant information. The framework strives to improve the user experience and give users a dependable and accurate platform.

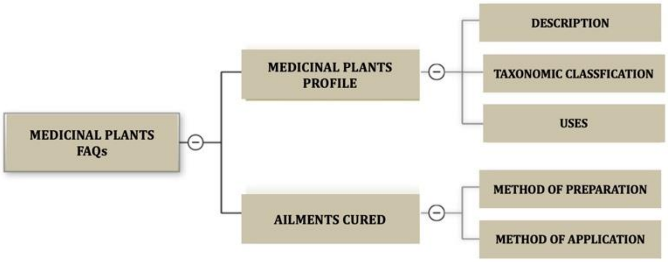


Fig. 5. Medicinal plants knowledge bases.

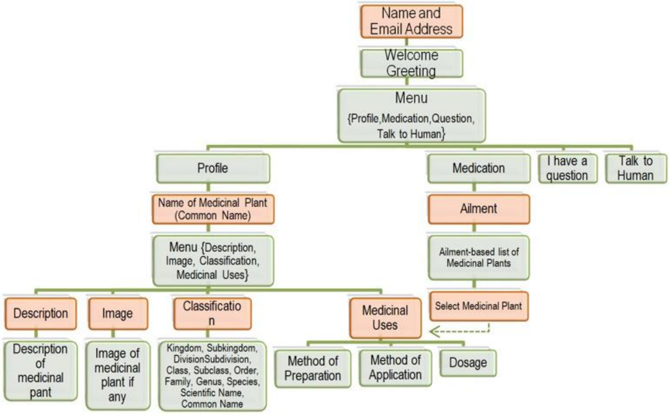

Name and Email Address
Welcome Greeting
Menu {Profile,Medication,Question, Talk to Human}
Profile
Medication
I have a question
Talk to Human
Name of Medicinal Plant (Common Name)
Ailment
Menu {Description, Image, Classification, Medicinal Uses}
Ailment-based list of Medicinal Plants
Select Medicinal Plant
Description
Image
Classificatio n
Medicinal Uses
Description of medicinal pant
Image of medicinal plant if any
Kingdom, Subkingdom, DivisionSubdivision, Class, Subclass, Order, Family, Genus, Species, Scientific Name, Common Name
Method of Preparation
Method of Application
Dosage

Fig. 6. The conversational flow of the MedPlantBot system.

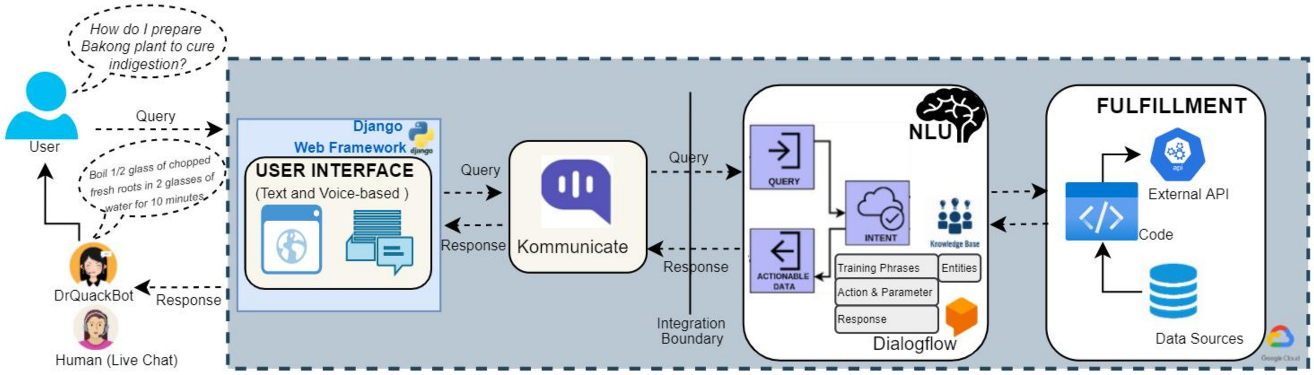


Fig. 7. MedPlantBot system design architecture.

3) Design prototype survey: The objective of this study is to develop a prototype system providing customers with accurate and useful information about medicinal plants. To evaluate its effectiveness, the researchers used a survey questionnaire that included three metrics: effectiveness, efficiency, and customer satisfaction [20]. The effectiveness metric measures how well the chatbot can provide accurate and relevant information. The efficiency metric measures how quickly the chatbot can respond to customer queries and provide solutions. The customer satisfaction metric measures how satisfied customers are with the overall performance of the chatbot.

The questionnaire was distributed to a selected group of one hundred fifty (150) participants, comprising traditional healers and herb users from Catanduanes with a keen interest in the use of medicinal plants for health and wellness, and IT professionals with expertise in chatbot technology and system design. The participants were chosen using the chain-referral and snowball methodology. During the survey, participants were required to engage in an online chat with MedPlantBot using any device such as a laptop, desktop computer, tablet, or smartphone. The survey questionnaire consisted of a series of questions that measured the participants' agreement with the chatbot's performance using a 5-point Likert scale. The scale included the following responses: Strongly Agree (SA), Agree (A), Neutral (N), Disagree (D), and Strongly Disagree (SD). Weighted mean statistics and frequency counts were used to evaluate the responses and determine the chatbot's overall performance. Along with the survey questionnaire, a non-disclosure agreement was provided to safeguard the participants' responses. The participants voluntarily participated in this study, and they had the right to quit at any time for any reason.

4) Designed framework's drawback: The survey results revealed that respondents advocated for the inclusion of more comprehensive explanations of medicinal plant data, such as videos, images, and web pages. This suggestion was considered and incorporated into the knowledge bases of the proposed framework. Yet, the survey results demonstrated that only a small proportion of respondents could predict the user experience. Thus, future studies will require a larger data set covering all medicinal plants information found in the province of Catanduanes and a bigger sample size to obtain more precise and reliable results.

5) Case Study (Future Work): Prior to the future use of the recommended platform, the system must be thoroughly tested to ensure that it meets the needs and expectations of the clients and that it functions properly. User acceptability testing, system testing, integration testing, and unit testing are a few instances of validation activities.

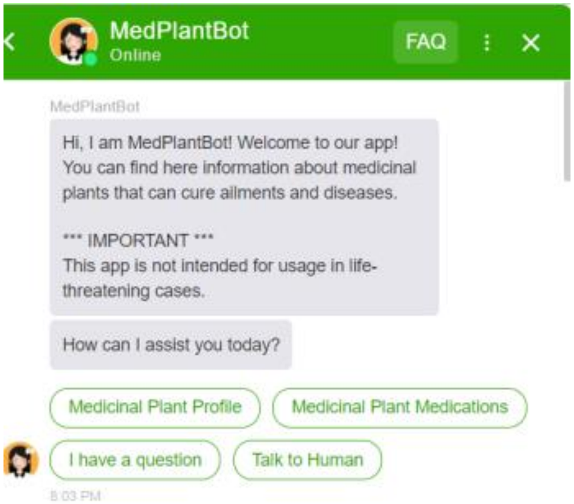
IV. RESULTS AND DISCUSSION

The focus of this research is to create, develop, and evaluate the effectiveness of an AI-based chatbot system that recommends beneficial plants and explains how to utilize them to treat diseases and illnesses. This section presents the developed MedPlantBot system prototype and its evaluation result.

A. The Developed MedPlantBot System Prototype

The MedPlantBot system prototype was developed using the dynamic approach when retrieving data. The prototype's system has functional and non-functional requirements. The functional requirements for the MedPlantBot are as follows: both regular users and admins must be given access to the system, and admins must be given the ability to administer inquiries, answers, and knowledge bases; the chatbot must be able to interact with users via text and speech, require the user to supply their name and email address before starting a conversation, answer user questions instantly, and be simple to integrate with well-known platforms like Messaging Apps, Google Assistant, Facebook pages, Skype, Telegram, and others; and the system shall permit real-time human interaction to answer questions that cannot be answered by the chatbot, and b) Non-Functional Requirements include, the system employs concise, clear, and easy-to-understand language; it shall maintain the knowledge bases and FAQs that are required for retrieval up to date, and it shall have an interactive and user-friendly interface. In less than three seconds, the system must immediately assign a live chat representative, react to inquiries in no more than three seconds, maintain discussion transcripts, be changeable and flexible, and must ensure the availability and accessibility of the system from any location; be able to seamlessly integrate with external APIs, and not need more than 24-hour maintenance outage per day.

Fig. 8 to 12 demonstrate the dialogue between the user and MedPlantBot. The first window that displays after login into the system is depicted in Fig. 8. The bot explained the application's objective. The user can choose from a menu of topics such as medicinal plant profile and medicinal plant drugs, enter the desired question, or talk with the available human agent to address their concerns.



Hi, I am MedPlantBot! Welcome to our app! You can find here information about medicinal plants that can cure ailments and diseases.

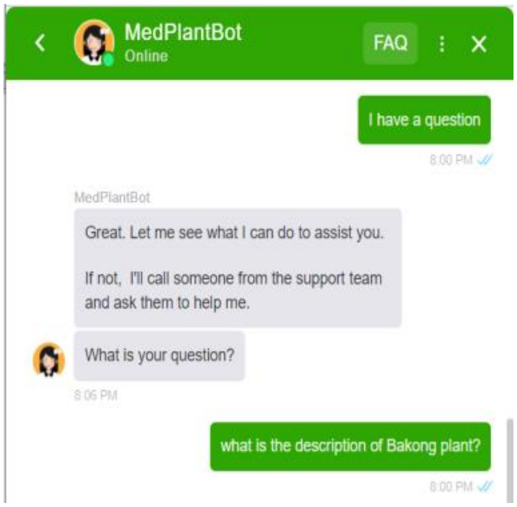
*** IMPORTANT ***
This app is not intended for usage in life-threatening cases.

How can I assist you today?

Fig. 8. MedPlantBot's welcome screen.

Fig. 9 shows an example of a user selecting "I have a question" from the menu. "What is the description of the Bakong plant?" asks the user. The customer is looking for information on Bakong, a medicinal plant. Fig. 10 depicts

MedPlantBot's response to the query. To help the user become more familiar with the Bakong plant, the bot supplied both a description and an image of the plant. The description and the image are presented where it was retrieved from the MedPlantBot system's knowledge base.



I have a question

8:00 PM

MedPlantBot

Great. Let me see what I can do to assist you.

If not, I'll call someone from the support team and ask them to help me.

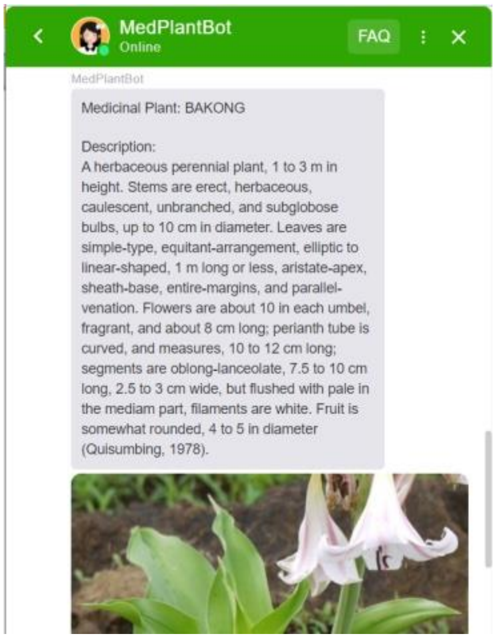
What is your question?

8:06 PM

what is the description of Bakong plant?

8:00 PM

Fig. 9. An inquiry about a certain medicinal plant.



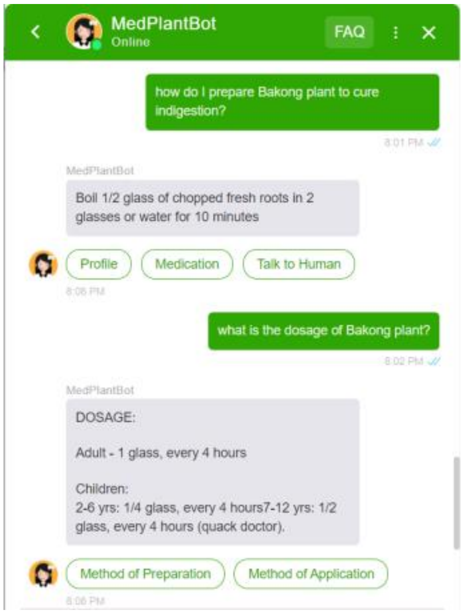
MedPlantBot

Medicinal Plant: BAKONG

Description:
A herbaceous perennial plant, 1 to 3 m in height. Stems are erect, herbaceous, caulescent, unbranched, and subglobose bulbs, up to 10 cm in diameter. Leaves are simple-type, equitant-arrangement, elliptic to linear-shaped, 1 m long or less, aristate-apex, sheath-base, entire-margins, and parallel-venation. Flowers are about 10 in each umbel, fragrant, and about 8 cm long; perianth tube is curved, and measures, 10 to 12 cm long; segments are oblong-lanceolate, 7.5 to 10 cm long, 2.5 to 3 cm wide, but flushed with pale in the mediam part, filaments are white. Fruit is somewhat rounded, 4 to 5 in diameter (Quisumbing, 1978).

Fig. 10. The response to the query of what a Bakong medicinal plant is.

Fig. 11 shows the user's request for instructions on how to prepare Bakong plant to treat indigestion. A response to the query is provided once it is found on MedPlantBot's knowledge base. As reflected, the MedPlantBot gave the customer the instruction: "Boil ½ glass of chopped fresh roots in 2 glasses of water for 10 minutes." And if additional questions arise, the user can choose an option from a menu such as a plant profile, medication, or even talking with the available human agent. The user can just put the desired question in the box if he prefers not to choose from the available alternatives.



how do I prepare Bakong plant to cure indigestion?

8:01 PM

MedPlantBot

Boil 1/2 glass of chopped fresh roots in 2 glasses or water for 10 minutes

8:06 PM

what is the dosage of Bakong plant?

8:02 PM

MedPlantBot

DOSAGE:

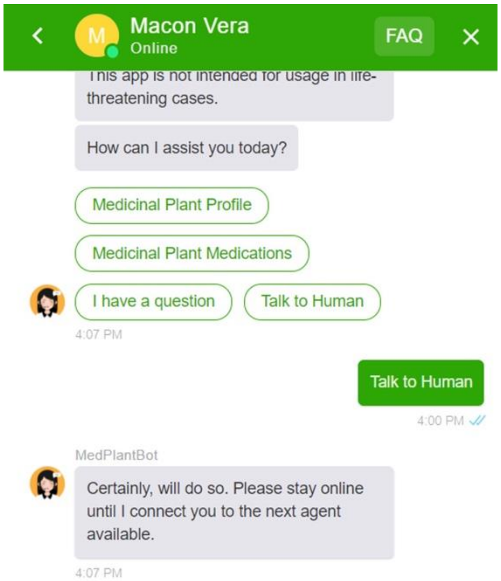
Adult - 1 glass, every 4 hours

Children:
2-6 yrs: 1/4 glass, every 4 hours7-12 yrs: 1/2 glass, every 4 hours (quack doctor).

8:06 PM

Fig. 11. Query on how to prepare and use a medicinal plant.

When the user wants to have a real-time chat with the available human agent, he can simply select from the menu "Talk to Human". Once selected, MedPlantBot automatically displays the name of the available human agent in the window. They can now converse with each other. This is depicted in Fig. 12.



This app is not intended for usage in life-threatening cases.

How can I assist you today?

4:07 PM

Talk to Human

4:00 PM

MedPlantBot

Certainly, will do so. Please stay online until I connect you to the next agent available.

4:07 PM

Fig. 12. The inquiry is forwarded by MedPlantBot to the available human agent.

B. *MedPlantBot's Performance Evaluation Result*

The proposed MedPlantBot prototype design framework underwent evaluation using a CUQ instrument, which assessed its effectiveness, efficiency, and satisfaction. A total of 150 people were contacted to provide comments and suggestions for improving the application's design. As shown in Fig. 13, the MedPlantBot system received a high-performance rating of

4.76 out of 5, indicating a strong level of agreement among users. This positive response is attributed to the user-friendly approach and architecture of the system.

While the feedback was generally positive, some respondents did provide improvement suggestions. Specifically, users requested clearer notifications that the application should not be used for life-threatening situations, as well as more detailed explanations of medicinal plant information, such as videos, images, and web pages, and they even suggested that the system should cover all medicinal plants with therapeutic properties found in Catanduanes These suggestions could further improve the user experience and usability of the MedPlantBot system if implemented.

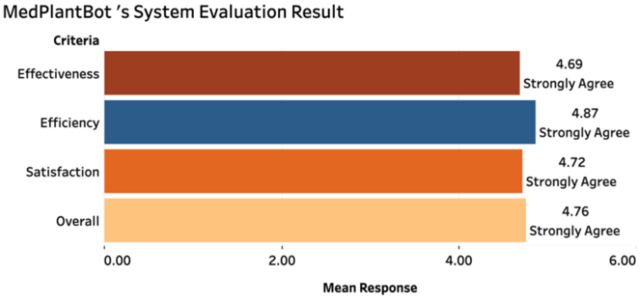


Fig. 13. MedPlantBot's system evaluation result.

V. Conclusion

This paper has introduced a new design paradigm for MedPlantBot, an AI chatbot that leverages medicinal plants to provide natural remedies for a variety of ailments. The study has identified local medicinal plant species and provided detailed descriptions of their taxonomy, applications, and uses. The development of chatbots requires a combination of various strategies and approaches, as demonstrated by several studies.

The findings of this study indicate that chatbots are extremely user-friendly and can provide appropriate responses and treatment recommendations for a wide variety of ailments, making them readily available to anyone seeking natural remedies. Using a survey questionnaire, the effectiveness of the MedPlantBot framework in delivering herbal medication was evaluated. Respondents strongly agreed that the chatbot provides an intuitive interface that is highly efficient, effective, and satisfying. Users lauded the chatbot's ability to prescribe relevant herbal treatments and provide customized treatment plans. The MedPlantBot system prototype has the potential to contribute significantly to the healthcare industry, while simultaneously encouraging sustainability and ethical practices. This discovery could serve as a springboard for the development of comparable technologies that can aid in the responsible use of natural resources.

VI. Implications and Future Work

The MedPlantBot technology has the potential to revolutionize healthcare through natural therapies. Its encouragement of medicinal plant use increases awareness of the importance of natural resources and contributes to the protection of biodiversity. In addition, the MedPlantBot serves as an educational tool for the community to learn more about the therapeutic usage of local medicinal plant resources. This framework offers a low-cost method for enhancing public health and welfare, aiding policymakers, and healthcare professionals in the development of herbal medication recommendations and regulations. Likewise, this research helps to the creation of a more sustainable and ethical approach to the use of medicinal plants, leveraging technology to ensure environmentally responsible resource management.

Future research can be carried out to enhance the chatbot's performance and broaden its capabilities, which can ultimately benefit more individuals in search of natural therapies. The MedPlantBot can be assessed and analyzed in terms of chatbot design characteristics based on ChatbotTest [18] such as a) *Personality*. The onboarding process is crucial for chatbots. Onboarding is uncommon, however. We observed that many users see Pedagogical Conversational Tutor (PET) more like a search engine than a conversational partner and that this perception is shared by many users: b) *Onboarding*. The Microsoft Azure Bot Framework-based personality-adaptive chatbot prototype is constructed on the web and can be inserted into a variety of channels since it is browser-based. For the task of employment suggestion, which is separated into two sub-dialogs, the appropriate discussion tree is triggered depending on the input personality: c) *Understanding*. The adoption of a chatbot for tourist planning in a unique way. By understanding the adoption variables, tourism businesses may use chatbot technology to improve the customer experience and increase consumer engagement; d) *Answering*. It must have access to an external body of wisdom and knowledge (for example, through data corpora) to perform the role of competence and respond to user inquiries, should keep data unique to the situation: e) *Navigation*. Information that links customers with programs services – which may contain a contact record's regular features as well as a textual description of the resource; f) *Error management*. The handling of errors in iHelper must be improved right away since typos are not caught. Using spell-checking APIs might help with this; and g) Intelligence. The conversation flow of intelligence chatbots may be tailored to clients and their use cases. The design component of chatbot software is essential. Furthermore, exploring the possibility of integrating a geographic information system (GIS) to map and expand the dataset of medicinal plants may be a promising avenue for future endeavors.

Acknowledgment

The authors would like to thank the Catanduanes State University (CatSU)-Natural Science Department Chairperson, Prof. Estrella T. Tribiana, for graciously sharing her knowledge, experiences, and expertise; Dr. Belen M. Tapado and other faculty members of the College of ICT, CatSU, for reviewing and providing feedback on MedPlantBot; and the concerned personnel of the Department of Environment and Natural Resources, Virac, Catanduanes, for their assistance.

References

[1] Yadav, P., Pandiaraj, T., Yadav, V., Yadav, V., Yadav, A., & Singh, V. (2020). Traditional values of medicinal plants, herbs an d their curable benefits. Journal of Pharmacognosy and Phytochemistry, 9(1), 2104-2106.

[2] Proclamation No. 698, S. 2004 | GOVPH. (2004, August 19). Official Gazette of the Republic of the Philippines. https://www.officialgazette.gov.ph/2004/08/19/proclamation-no-698-s-

2004/.

[3] Otto, M., Thornton, J., & Contributors, B. (2020). Traditional Knowledge Digital Library. Philippine Traditional Knowledge Digital Library on Health. https://www.tkdl.ph.

[4] Talan, L. (1998). Medicinal plants of Virac, Catanduanes [Philippines] a morpho-systematic study. Philippine Journal of Crop Science (Philippines).

[5] Tulay, F. O. (2003). Evaluation and Documentation of Herbal Remedies Among Folk Doctors in Libjo, Bato, Catanduanes. Catanduanes State Colleges.

[6] Karanja, M. (2018). Role of ICT in dissemination of information in secondary schools in Kenya: A literature based review. Journal of Information and Technology, 2(1).

[7] Vera, M. C. S., & Comendador, B. E. V. (2016). A web-based student support services system integrating short message service application programming interface. International Journal of Future Computer and Communication, 5(2), 77.

[8] Neapolitan, R. E., & Jiang, X. (2018). Artificial intelligence: With an introduction to machine learning: CRC Press.

[9] Sánchez-Díaz, X., Ayala-Bastidas, G., Fonseca-Ortiz, P., & Garrido, L. (2018). A knowledge-based methodology for building a conversational chatbot as an intelligent tutor. Paper presented at the Mexican International Conference on Artificial Intelligence.

[10] KV, S. G., and N. Malarvizhi. . (2021). Medbot-Medical Diagnosis System using Artificial Intelligence. EAI Endorsed Transactions on Smart Cities, e12.

[11] Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. Paper presented at the IFIP International Conference on Artificial Intelligence Applications and Innovations.

[12] Daniel, E. E., et al. . (2021). COVID-19 I-SABI Chat-bot Application Using the Natural Language Processing with Dialogflow. innovative Journal of Science (3.3), 55-72.

[13] Ahmad, N. A., Che, M. H., Zainal, A., Abd Rauf, M. F., & Adnan, Z. (2018). Review of chatbots design techniques. International Journal of Computer Applications, 181(8), 7-10.

[14] Ranavare, S. S., & Kamath, R. (2020). Artificial intelligence based chatbot for placement activity at college using dialogflow. Our Heritage, 68(30), 4806-4814.

[15] Muhammad, A. F., Susanto, D., Alimudin, A., Adila, F., Assidiqi, M. H., & Nabhan, S. (2020). Developing English conversation chatbot using dialogflow. Paper presented at the 2020 International Electronics Symposium (IES).

[16] Sabharwal, N., & Agrawal, A. (2020). Cognitive virtual assistants using google dialogflow: develop complex cognitive bots using the Google dialogflow platform: Apress.

[17] Vergadia, P. (Accessed 28 Jan. 2022). Deconstructing Chatbots: Getting Started with Dialogflow. medium.com/google- cloud/deconstructing-chatbots-getting-started-with-dialogflow- 4f91deb32135, Google Cloud, 11 Feb. 2019.

[18] Chatbottest. “The Free Guide for You to Understand What Is Your Chatbot Doing Wrong.” Chatbottest https://chatbottest.com.

[19] Huang, X., and ASCS CIS. (2021). Chatbot: Design, Architecture, and Applications.

[20] Radziwill, Nicole M., and Morgan C. Benton. "Evaluating quality of chatbots and intelligent conversational agents." arXiv preprint arXiv:1704.04579 (2017).

[21] Butcher, C., & Schutte, C. (2015). Technology selection framework for port development projects. Paper presented at the Proceedings of the International Association for Management of Technology IAMOT 2015 Conference Proceedings, Cape Town, South Africa.

[22] Avila, T. J. (2004). Medicinal Plants Utilization and Practices of Traditional Healers in Catanduanes. Virac, Catanduanes, Philippines. Graduate School Catanduanes State Colleges.

[23] Department of Environment and Natural Resources-Ecosystems Research and Development Bureau(2015), Research Information Series on Ecosystems Vol 14, No.2, May-August 2002. URL: https://erdb.denr.gov.ph/wp-content/uploads/2015/05/r_v14n2.pdf.

[24] McGrath, G., & Brenner, P. R. (2017). Serverless computing: Design, implementation, and performance. Paper presented at the 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW).

Suppressing Chest Radiograph Ribs for Improving Lung Nodule Visibility by using Circular Window Adaptive Median Outlier (CWAMO)

Dnyaneshwar Kanade[1], Jagdish Helonde[2]
Research Scholar, School of Engineering and Technology, Sandip University[1]
Professor, School of Engineering and Technology, Sandip University[2]

***Abstract*—Chest radiograph ribs obstruct lung nodules. To see the nodule under the chest radiograph ribs, remove or suppress them. The paper describes a circular median filter approach for finding outliers in chest radiographs. The method uses 147 Japanese Society of Radiological Technology x-ray pictures (JSRT). Pixels with intensities two standard deviations above the median are median outliers. Contrast-Limited Adaptive Histogram Equalization enhances nodule visibility (CLAHE). The method is tested on modest chest radiographs and compared to the Budapest University Bone Shadow Eliminated X-Ray Dataset methodology. The initial test uses 50 modest chest radiographs (Test 1). The proposed approach is applied after active shape modelling (ASM) lung segmentation. True positive nodules are seen on 89% of chest radiographs of various subtleties. Test-2 and Test-3 used 20 subtlety-level photos. In Test-2, the peak signal-to-noise ratio (PSNR), mean-to-standard deviation ratio (MSR), and universal image quality index (IQI) are evaluated for the full image and compared to the existing algorithm. For all three parameters, the suggested technique outperforms the algorithm. Test-3 computes nodule MSR and compares it to Budapest University's Bone Shadow Eliminated Dataset and original chest radiographs. The new algorithm improved nodule area contrast by 3.83% and 23.94% compared to the original chest radiograph. This approach improves chest radiograph nodule visualization.**

***Keywords*—Lung cancer; chest radiograph; contrast limited adaptive histogram equalization; median outlier**

I. Introduction

Imaging techniques such as radiography, magnetic resonance imaging (MRI), and computed tomography (CT) are all regularly utilized in the field of medical diagnosis for the purpose of locating lung cancer. Despite the fact that MRI and CT are more sensitive and accurate techniques, chest radiography is still the most recommended type of process for the initial diagnosis and detection of lung cancer due to the fact that it is a noninvasive form of operation, has a low radiation dosage, and is affordable. It is necessary to eliminate the shadow of the chest ribs from the chest radiograph before beginning the process of identifying candidate nodules from the chest radiograph. The chest rib shadows have been removed from the chest radiograph using a variety of methods that have been proposed in a number of studies. These methods also detect lung nodule-like components in the chest radiograph. This article discusses the elimination of chest rib shadows from chest radiographs using the median outlier technique and the detection of nodule-like parts using the thresholding technique. Both of these techniques are discussed in detail in Section III. Before the nodule was found, the images were improved using a technique called CLAHE. This allowed for a clearer view of the area. The lungs are segmented with ASM [1], [2], [3], [4]. The image database maintained by the JSRT [5] is utilized by the proposed approach (CWAMO). The proposed method is unsupervised and does not require either large dataset or training of it. The adaptive nature of the median outlier filter archives consistent results for differently illuminated chest radiographs. The proposed method is evaluated for its usefulness by calculating different measure such as PSNR, universal IQI and MSR for the entire image and nodule area and then compared them with the existing algorithm.

The paper can be structured in the following manner: In the Second II, several similar efforts on the removal of bone shadows from chest x-ray pictures are discussed. In the Section III of this research project, both the problem statement and the technique are discussed. Experimentation is performed in Section IV. Section V presents a comprehensive study of the currently used method as well as the proposed improvements, and Section VI brings the entire research to a close.

II. Related Work

The current state of the art incorporates a number of techniques for the suppression of ribs and the segmentation of lung nodules, all of which are based on standard chest radiography practices. The outcomes of the offered strategy are evaluated in light of the dataset called the "Bone Shadow Eliminated Image Dataset” [6] developed by Budapest University (BSE-BU). By appointing a slope to every point in the image, a one-of-a-kind process creates a model that ignores the information about its positioning. These slopes can be graphed using a slope field and are represented by a function that requires two inputs to operate. Three characteristics are sufficient to identify the ribcage, as indicated by research that was carried out on a range of tasks. The function was analyzed through the use of principal component analysis, which resulted in the generation of the function's final iteration, which was a two-variable, third-order rational function. Due to the fact that it spans chest bones, the nodule is difficult to identify with conventional chest radiography. Independent Component Analysis [7] is used in the process, which suppresses the rear ribs and clavicles (ICA). By utilizing this method, the visibility of the nodule will be improved, and the strain on the automatic nodule recognition module will be decreased. Active appearance models and active shape models are integrated with a k-nearest neighbor classifier and a filter bank composed of different scales of Gaussian derivatives for the purpose of evaluating and contrasting various supervised segmentation algorithms. The techniques were tested using a database that was available to the general public and had 247 chest radiographs. In this database, all of the elements had been manually segmented by two human observers. The process of segmenting the clavicle is challenging regardless of the approach used [8]; dynamic shape models produce the greatest results, but the performance of humans is noticeably superior. Iterated contextual pixel classification (ICPC) [9] is an iterative, pixel-based, supervised, statistical classification procedure. The initial segmentation of the ribs acquired by pixel classification is updated by the ICPC, which does so by reclassifying each pixel by making use of both its original attributes and the knowledge regarding the class labels of neighboring pixels. The methodology is analyzed by using thirty radiographs taken from a database maintained by the JSRT. ICPC achieves a classification accuracy of 0.86 spl plus mn or 0.06 in an experiment using sixfold cross-validation, whereas the second human observer achieves an accuracy of 0.94 spl plus mn or 0.02 in the same experiment. In [10], x-rays is described. This system combines three different methods of doing so. The interior, border, and head of the clavicle are each classed in a separate category and divided into two stages, respectively. This is the data that is sent into the active shape model so that it can be segmented. In order to arrive at the desired end result, information regarding the shape obtained from the active shape model is combined with appearance data obtained from the inside of the clavicle, the border, the head, and dynamic programming. The method is evaluated against a wide variety of techniques that have been published in the past as well as against impartial human observers using a huge database. The approach that is advised produces a mean contour distance of 1.1 millimeters and an intersection over union distance of 0.86 millimeters when applied to 249 test photographs. a one-of-a-kind method that segments x-rays through the use of a dynamic programming strategy, so releasing them from the shadows that are cast by the rib cage and the clavicle [11]. When moving to a new place, the separated shadows will no longer be present. After being cleaned, the images are put through a hybrid lesion detector that uses data on gradient convergence, contrast, and intensity. The use of a support vector machine helps get rid of inaccurate findings. The method can eliminate 80% of bone shadows and 84% of them in the posterior region, despite having an average segmentation error of 1 millimeter. The frequency of false positives was reduced from 2,94 to 1 thanks to shadow reduction, which achieved a sensitivity of 63% for malignant tumors. The following is a rundown of the five primary stages involved in the suggested methods: After the lung has been segmented, the clavicle and ribs are located, their edges are determined, their profiles are computed, and suppression is carried out by making use of the projected profiles. The clavicle and the rib components are hidden from view in the image provided by the bone-suppression algorithm [12]. On a total of 491 photographs, the effectiveness of rib suppression was analyzed. It was found that 83.06% (or 6.59%) of the rib structures on a typical chest image were suppressed, which is approximately comparable to an average of one visible rib on a rib-suppressed image. This was discovered through the use of image analysis software. This conclusion was reached after comparing the rib areas recognized by the computer with the rib areas sketched by hand. Data mining and image processing are used to create a system for automatic rib recognition [13]. To begin, a number of template matching techniques and approaches based on graphs are employed to determine the position of the rib center line. After that, a support vector machine is utilized in order to model the way in which the ribs are positioned with respect to one another as well as to locate mistakes. In conclusion, decision trees are utilized in order to enhance the result of center line recognition. The JSRT dataset is used to conduct an analysis of the approach. The findings of rib recognition are greater than ninety-two percent for sensitivity and ninety-eight percent for specificity. A novel method expresses [14] that there is a different method for dividing up the ribs in chest radiography. Several applications of the Gaussian filter were required in order to get rid of the dataset's chaotic background. Detail images could be generated using the multi-scale wavelet decomposition, and then features extracted from the detail images using the Gaussian derivative. In order to properly categorize ribs, the support vector machine (SVM) rib model was developed. The convolutional neural filter (CNF) [15] for bone suppression is based on a convolutional neural network, which possesses excellent image processing capabilities and is frequently used in the field of medicine. [CNF] stands for "convolutional neural filter." The system that utilized CNF and consisted of six convolutional layers was successful in suppressing 89.2 percent of the bone. An intensive multitasking U-Net is offered as a solution to the problem of segmenting bones from a chest radiograph [16]. This solution is based on a sufficient dataset. A four-fold cross-validation is used to assess the performance of the proposed network on 88 chest radiograph pictures. The suggested technique for segmenting all bones results in average DSC values of 88.38%. These values are listed from highest to lowest. The proposed research produced lung digital x-rays, bone digital x-rays, and bone-free digital x-rays from 59 high-resolution CT images of the lung [17]. These DRs were utilized for the purpose of the suggested cascade convolutional neural network's internal evaluation as well as its training (CCNN). The authors propose an improved lung-simulated DR with varied bone intensity weightings by utilizing a 3-step image processing architecture. In addition to DR modeling and feature expansion, CT segmentation was also performed. Networks for bone identification and suppression are available through the CCNN. During the process of external validation, the trained CCNN was examined on a total of 30 chest radiographs. The clavicles and ribs have the ability to disguise even relatively minor abnormalities, which can lead to incorrect diagnoses. To reduce the number of radiological interpretation errors, particularly DL processes that are associated with tuberculosis identification, the purpose of this study [18] is to develop a deep learning-based bone suppression model. This model will identify and remove these obstructing bony structures from the frontal CXRs (TB). Multiple bone suppression models, each with their own unique deep architectures, are trained, optimized, and finally evaluated at a number of different institutions using the combined loss function that was mentioned before. The theory was put to the test with the help of DeBoNet, which is a collection of models for convolutional neural networks [19] that inhibit frontal CXR bones. A specific collection of computed tomography images and their bone-suppressed analogues were used in the training and evaluation of U-Nets, Feature Pyramid Networks, and several other custom models. In terms of PSNR (36.797771.6210), MS-SSIM (0.98480.00), and other measures, DeBoNet surpassed the top three models that were used in its development. A novel approach generated a virtual dual energy (VDE) image with MTANN that suppressed ribs and clavicles to expose hidden nodules [20]. On chest radiographs, the ability of radiologists to detect pulmonary nodules was compared to the performance of bone suppression imaging using deep learning and bone subtraction imaging using a dual energy technique (CXRs) [21]. A completely automated system performs an analysis of digital posteroanterior (PA) chest radiographs [22]. This analysis begins with the segmentation of the lung field. Even the lungs that are normally neglected in conventional approaches are included in the segmented lung region because it includes the lungs that are concealed behind the heart, spine, and diaphragm. A projected nodule selection strategy is put into action once the partitioned region has been processed using a straightforward multiscale method in an effort to enhance the visibility of the nodules. To remove false positives, cost-sensitive support vector machines (SVMs) detect actual nodules. On two different feature-selected data sets, a number of Gaussian and polynomial SVM learning tests with variable parameter settings were carried out. The best SVM models have a false-positive rate of 1.5 per image when the sensitivity is set at 0.71. When the sensitivity is increased to 0.78 and 0.85, the rate increases to 2.5 and 4, respectively. The suggested approach generates seven or eight frames per second while maintaining the highest possible sensitivity (0.92–1.0). Using a Laplacian of Gaussian (LoG) filter as the basis for an approach to chest X-ray dot-like augmentation is discussed [23]. The directional texturing of lung nodules is improved when the X and Y axes are given the ability to have varied scale values. Image capture, image pre-processing, candidate nodule recognition, and feature extraction are the individual components that make up a computerized technique for recognizing nodules on chest radiographs [24]. The program is responsible for processing both SCLC and NSCLC pictures. In order to analyze the geometrical and textural properties of fifty photographs, twenty-five from each category were selected. Textural features were determined using GLCMs. Experiments involving the selection of image features provided novel results [25]. In order to get the best combination of traits, 210 different features were evaluated. Employing a procedure known as "forward stepwise selection," For each feature set, the area under the ROC curve served as the criterion for evaluation. It is suggested in [26] that images be pre-processed using a median filter. The thresholding method developed by Otsu is utilized in the segmentation process. The GLCM algorithm and various measurements of the physical dimensions are utilized throughout the process of feature extraction. ANNs are used to categorize the phases of disease. In this study, twenty-three wavelets from four different wavelet families—Daubechies, Haar, Biorthogonal, and Reverse Biorthogonal Spline—are compared in order to determine if lung nodules are benign or malignant [27]. In order for researchers to differentiate very small items from the background, they use three different sizes of regions of interest. For testing the suggested system, actual database pictures accompanied by tags are employed. Detecting lung nodules in chest radiographs is made possible by a computer system that uses multi-scale image processing [28]. This strategy accounts for the wide range of sizes that lung nodules can take, such as Active Shape Model lung field segmentation; quadratic classification with Lindeberg's multi-scale blob detector for nodule candidate detection; multi-scale edge focusing for blob segmentation; k-nearest neighbor classification; k-nearest neighbor classification. When looking at the comprehensive JSRT database, we find that allowing two false positives per image results in the detection of 50.6% of nodules. The percentage rises to 69.5% after the addition of 10 false positives. The classification of image thresholding techniques, the standardization of their formulas, and the comparison of their performances are all carried out [29]. The shape of the histogram, the grouping of the measurement space, the entropy of the data, the qualities of the objects, the spatial correlation, and the local gray-level surface data are all factors that differentiate the thresholding procedures. A comparison of forty different thresholding methods for nondestructive testing and document photos is shown here. Considering integrated performance indicators side by side We have developed thresholding methods that are more effective than nondestructive testing and document photography. Background may be removed with the help of the local mean and standard deviation, thanks to an innovative locally adaptive thresholding technique [30]. Integral sum image preparation calculates the local mean. A variance-based threshold selection criterion is developed by projecting the two-dimensional histograms of the original image and the local average image onto a one-dimensional space [31]. This results in the establishment of a variance-based threshold selection criterion. The suggested image thresholding strategy outperforms the Otsu, two-dimensional Otsu, and least class variance thresholding techniques, according to results obtained on bi-level and multi-level thresholding for artificial and real-world images. These results indicate that the proposed strategy is superior. The detailed study that provides a description of the finding of lung tumors on a chest graph [32] is presented. The approaches of watershed segmentation, thresholding, active contour, and differential operator, as well as expanding region [33], are compared. Two radiologists [34] used a retrospective investigation of a total of 100 patients, of whom 50% had lung nodules and the remaining 50% had no nodules, to confirm the chest CT findings. Using a computer-aided design (CAD) program, researchers compared four different methods for diagnosing lung nodules and masses. One hundred incidents were investigated in total. The mass algorithm outperformed the other three. A new hybrid deep learning framework (CNN) was created by combining convolutional neural networks, visual geometry group-based neural networks (VGG), data augmentation, and spatial transformer networks (STN) [35]. The validation accuracy for the suggested technique is 73%, whereas the accuracy ratings for vanilla gray, vanilla RGB, hybrid CNN and VGG, and modified capsule networks are 67.8%, 69%, and 63.3%, respectively, for the complete dataset. An improved region growth (IRG) algorithm was used to segment the lung tumor in a way that was both more accurate and quicker than the traditional methods [36]. When applied to CT scans of four different patients, the overall strategy resulted in a 98% increase in segmentation accuracy when compared to the straightforward approach. The tumor was successfully contained by the multipoint growth-start combination. The technique improved tumor identification by a margin of less than 13% while also increasing compliance accuracy to an adequate level. A novel computer vision technology called understandable decision trees is used to predict the aggressiveness of lung cancer by employing decision trees [37]. This technique combines decision trees and deep learning. The deep learning component of this strategy was created with the use of a large set of pathological markers that are readily available to the public and are related to lung cancer. These techniques estimate chest X-ray biomarker scores utilizing two data sets tagged with malignancy. We found that fitting shallow decision trees to datasets that had been stratified based on the chance of cancer produced the best results. The most effective decision tree model has a specificity of 80%, a sensitivity of 86.7%, and a positive predictive value of 92.9%. Most of the proposed techniques are supervised in nature and need large dataset. These techniques also require training the model which takes lot of time. On the other hand, the non-supervised techniques proposed by other researchers assume equal illumination of all the images which is not the case in reality.

III. Proposed Method

A. Assumptions

Prior to using the proposed method for suppressing the chest radiograph ribs for improving nodule visibility and detecting nodule-like areas, the following assumptions are made:

1) The JSRT digital x-ray images [6] with bone shadow are the ones that were used for the dataset for the suggested approach. The dataset is made up of 154 traditional chest radiographs that show a lung nodule and 93 radiographs that do not show a lung lesion.

2) In the JSRT dataset, each chest radiograph with a lung nodule has only one nodule.

3) The study makes use of images ranging from a subtlety level of 5 (the nodule is extremely evident) to a subtlety level of 2 (the nodule is very subtle). The proposed method doesn't use images with a subtlety level of 1, which is sometimes called "very subtle," because it's very hard to pull nodules out of those images because they have low contrast, are small, or overlap with normal structure.

4) The circular median filter used has a radius of 50 pixels, and the threshold used is two standard deviations.

5) The block size of CLAHE [1] is 127, and the number of histogram bins is 256.

6) The maximum slop of CLAHE is 10.

7) For comparing the proposed method results, the JSRT bone shadow-eliminated images dataset by Budapest University (BSE-BU) [7] is used.

B. Algorithm

The proposed method is divided into five steps, namely: (1) Preprocessing of the image (2) Chest radiograph suppression (3) Contrast enhancement using CLAHE (4) Lung segmentation using ASM (5) Candidate nodule detection using Bernsen local contrast thresholding with a circular window (CWLCT) (Fig. 1).

1) Preprocessing of chest radiograph: In the preprocessing phase, the original chest radiograph pictures are boosted for contrast and then down sampled to a resolution of 1024 by 1024 pixels, which is a reduction from their original size of 2048 by 2048 pixels, before being used to suppress rib shades.

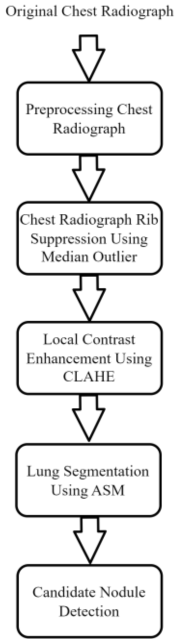

Original Chest Radiograph
Preprocessing Chest Radiograph
Chest Radiograph Rib Suppression Using Median Outlier
Local Contrast Enhancement Using CLAHE
Lung Segmentation Using ASM
Candidate Nodule Detection

Fig. 1. System flow chart.

2) Chest radiograph rib suppression using median outlier: A circular median filter is applied to the chest radiograph in order to reduce the visibility of the ribs' shadows. The size of the image and the maximum height of a rib on the chest radiograph are used to establish the value for the radius of the circular median filter. The circular window was centered at the pixel *f(x,y)* to calculate the median of the pixel intensities of the image region under the circular window. By using the standard deviation as the threshold value, the pixel intensity at *f(x,y)* will be replaced with the median value ***m*** if it is brighter than the median by the threshold value t; otherwise, it will be kept at its original value.

$$I_1(x,y) = \begin{cases} m \; if \; I(x,y) > m+t \\ I(x,y) \; otherwise \end{cases} \quad (1)$$

Where *I(x,y)* is the original gray level intensity at pixel *f(x,y)* and *I_1(x,y)* is modified gray level intensity at pixel *f(x,y).*

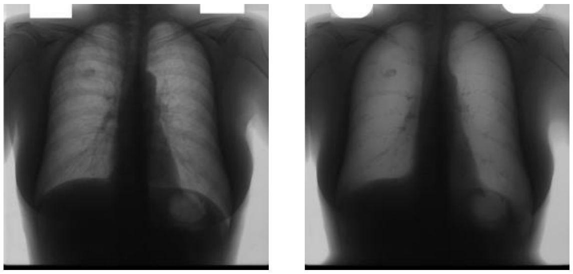

(a) (b)

Fig. 2. Rib suppression. (a) Original chest radiograph. (b) Rib suppressed chest radiograph.

Fig. 2(a) shows original chest radiograph while Fig. 2(b) shows ribs suppressed image using proposed AMO method.

3) Local contrast enhancement by CLAHE: Due to the concentration of the histogram in regions with nearly constant contrast, standard AHE has a tendency to overamplify the contrast in these areas. AHE may thus lead to noise amplification in regions with near-constant noise. In order to lessen the issue of noise amplification, contrast amplification is restricted in adaptive histogram equalization (CLAHE).

The magnitude of the contrast amplification in CLAHE close to a particular pixel value is determined by the slope of the transformation function. This is proportional to the value of the histogram at that pixel value as well as the slope of the cumulative distribution function (CDF) for the surrounding area. Before computing the CDF, CLAHE performs a clipping operation on the histogram at a value that has been previously determined in order to reduce amplification. Because of this, the slope of the transformation function is restricted due to the constraints placed on it. The so-called clip limit, also known as the value at which the histogram is clipped, is established based on the normalization of the histogram and, as a direct consequence of this, the size of the neighborhood region. Fig. 3(c) shows an enhanced image using CLAHE.

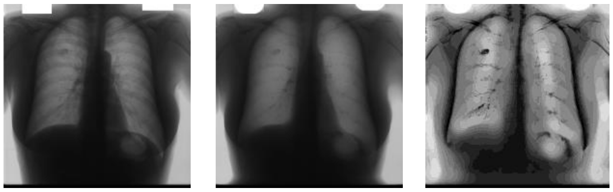

(a) (b) (c)

Fig. 3. Contrast enhancement using CLAHE (a) Original chest radiograph. (b) Rib suppressed chest radiograph (c) Contrast enhanced image.

4) Lung segmentation: An active shape model is used for extracting lung areas from the chest radiograph. One of the most reliable methods for picture segmentation is ASM. The active shape model is a parameterized contour model. The parameters are obtained using principal component analysis (PCA) and the statistics of numerous sets of points collected from various contours of comparable images. In ASM, the object's boundary is established by n points. These points produce the descriptor vector, which is generated as

$$x = (x_1, y_1, x_2, y_2, \ldots\ldots.. x_n, y_n)^T \quad (2)$$

where x_i and y_i are the i-th point on the contour's x and y coordinates respectively. The principal component analysis-calculated s training vectors have a mean shape of

$$\bar{x} = \frac{1}{s}\sum_{i=1}^{s} x_i \quad (3)$$

There will be a covariance matrix, defined as

$$S = \frac{1}{1-s}\sum_{i=1}^{s}(x_i - \bar{x})(x_i - \bar{x})^T \quad (4)$$

The covariance matrix's first t biggest Eigen values are picked. Due to the fact that only the eigenvalues with the greatest significance are taken into account, the number of parameters is drastically reduced and ultimately falls below n. The appropriate eigenvectors are all present in the matrix. The model's inputs are calculated as follows:

$$b = \Phi(x - \bar{x}) \quad (5)$$

From this, the approximate form is determined as follows.

$$x \approx \bar{x} + \Phi b) \quad (6)$$

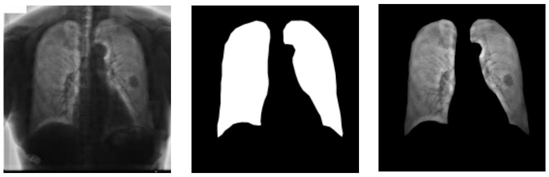

(a) (b) (c)

Fig. 4. Lung segmentation. (a) Original chest radiograph. (b) ASM lung mask. (c) Lungs part of chest radiograph.

The appropriate b vector is found for a given contour when all of its components fall inside the range ±m√λi, with the right constant m. Starting with the mean shape, the parametric description of an object's contour is looked up. Until convergence or a preset number of iterations are reached, two alternate steps are used. Each contour point perpendicular to the contour is moved in the first step of the procedure. After numerous iterations of position testing on both sides of the contour, the ideal position is identified. Picture resolution is used in the training model to determine where an intensity gradient profile should be placed at each contour node. Finally, the ideal replacement location for the contour point is determined using the Mahalanobis distance. Then, all of the contour points are modified in order to fit a model to the new point set. The ideal b parameter is searched using (6). As the image's spatial resolution increases, the entire procedure is repeated numerous times. The proposed method used 100 chest radiographs from the JSRT dataset to train the model. The original chest radiograph is shown in Fig. 4(a). Fig. 4(b) displays the segmented lung mask that was produced using ASM. By using this mask on the original X-ray image, as seen in Fig. 4(c), the lungs are segmented.

5) Candidate nodule detection: The proposed method uses Bernsen local contrast thresholding [38] with a circular window with diameter w for image binarization instead of a square window while calculating local contrast, as shown in Fig. 5. For pixel P centered at i and j, a square window of size w x w takes 22% more neighboring pixels into account than the circular window having diameter w. The circular window used in the proposed method for image binarization plays an important role while calculating the local contrast for the boundary pixels of the true positive nodule. In the post-processing stage, morphological operations are used to remove unwanted regions. A significant reduction in false positive nodules is obtained by using prior knowledge of the true positive nodule. Based on the circularity index of the candidate nodules final candidate nodule segmented image is obtained as shown in Fig. 6(b) the white nodule is true positive while gray is false positive.

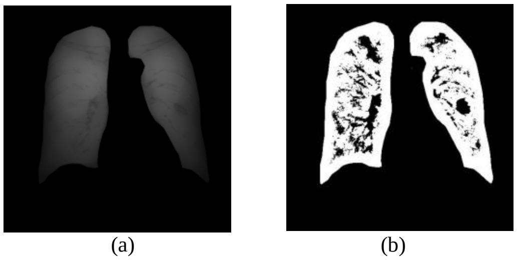


Fig. 5. Local contrast thresholding. (a) Rib suppressed chest radiograph. (b) Binarization using local contrast thresholding.

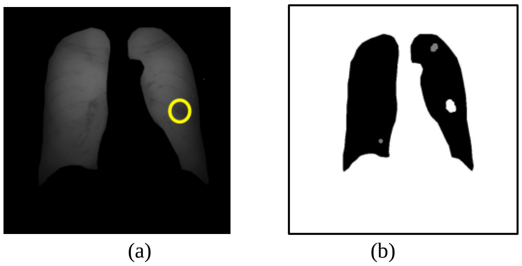


Fig. 6. Post processing. (a) Original chest radiograph with annotation. (b) Candidate nodules (White is True Positive).

IV. Experimentation and Results

1) Test-1: 50 images with subtlety level 5 to 2 are used for experimentation. After preprocessing and lung segmentation using ASM, circular window local contrast thresholding technique applied for binarization. It is observed that the 90 % of the true positive nodules extracted are part of candidate nodules. The average rate of false positive nodules per image is 3.5 fp/image. Fig. 7 shows the results of candidate nodules extraction using local contrast thresholding with circular window for the images with subtlety level 5 to 2.

2) Test-2: Test-2 deals with the image quality measures of the proposed rib suppression algorithm by comparing it's results with the (Budapest University Bone Eliminated JSRT dataset) existing algorithm [7]. The second test, known as Test-2, evaluates entire image quality. In Test-2, we make use of the sample original images; images from the Budapest University Bone Eliminated JSRT dataset, as well as images obtained utilizing the proposed approaches. The quality of the images was evaluated in this study with the use of a number of different criteria. The criteria that will be used for the evaluation are 1) Peak Signal to Noise Ratio (PSNR), 2) Mean to Standard Deviation Ratio (MSR), and 3) Universal Quality Index (IQI).

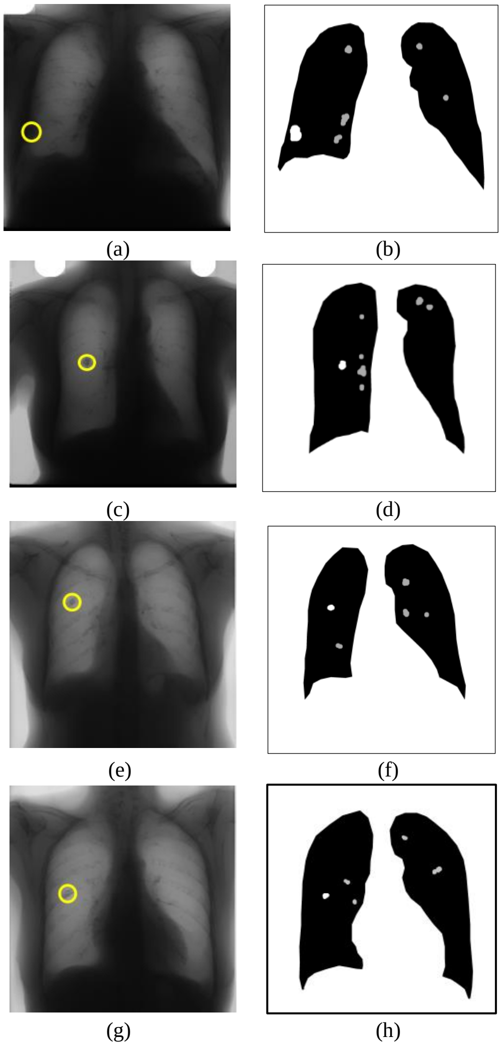


Fig. 7. Some of the results obtained -Rib suppressed using proposed method Images with subtlety levels 5,4,3 & 2 sequentially (Left Column-(a), (c), (e), (g)), results of candidate nodule segmentation (White-True positive nodules) using local contrast thresholding and circular window (Right column-(b), (d), (f), (h)).

For image level analysis all above three measures are used while for nodule area analysis only MSR is used.

a) Test 2.1 Peak signal-to-noise ratio: Peak signal-to-noise ratio (PSNR) is expressed in decibels (dB) (decibels). PSNR is defined as

$$PSNR = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \qquad (7)$$

where MAX_I is the highest value a pixel can have within an image, and MSE is determined by:

$$MSE = \frac{1}{ij}\sum_{m=0}^{i-1}\sum_{n=0}^{j-1} \parallel I(m,n) - K(m,n) \parallel^2 \quad (8)$$

Where I is original JSRT database image and K is either Budapest Bone Shadow Eliminated Image or Rib suppressed image using proposed method. Table I shows comparison of PSNR of Budapest University Images and result Images of the proposed method while Fig. 8 shows it graphically.

TABLE I. PEAK SIGNAL TO NOISE RATIO (PSNR)

Image	PSNR Existing Algorithm*	PSNR Proposed Method
JPCLN002.jpg	7.375601102	7.678650215
JPCLN003.jpg	12.11304948	12.47926008
JPCLN005.jpg	12.56962625	12.98963757
JPCLN006.jpg	9.227926597	9.618855472
JPCLN029.jpg	11.94079299	12.33675425
JPCLN034.jpg	12.1818185	13.05766792
JPCLN035.jpg	9.780592944	10.31366197
JPCLN041.jpg	11.88327603	12.30280025
JPCLN051.jpg	9.417197293	9.792465158
JPCLN052.jpg	9.856187759	10.17043184
JPCLN053.jpg	10.80607195	11.4255127
JPCLN058.jpg	11.31527493	11.97189975
JPCLN103.jpg	8.502726484	9.266778262
JPCLN104.jpg	11.25931517	11.85082129
JPCLN114.jpg	8.586153337	8.942296706
JPCLN116.jpg	9.779299456	10.15226717

*Existing Algorithm-Bone Shadow Eliminated X-Ray image dataset by Budapest University

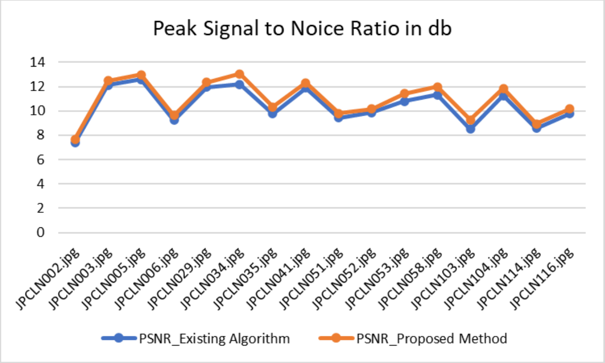


Fig. 8. PSNR using budapest bone shadow eliminated images dataset (existing algorithm) and proposed method.

b) Test 2.2 Mean to standard deviation ratio of Image: After calculating the mean and standard deviation of the entire image, it is simple to calculate the mean-to-standard deviation ratio (MSR). MSR stands for mean to standard deviation ratio. It would be better if MSR increased. Table II shows comparison of MSR of JSRT original images, Budapest University Images and result Images of the proposed method while Fig. 9 shows it graphically.

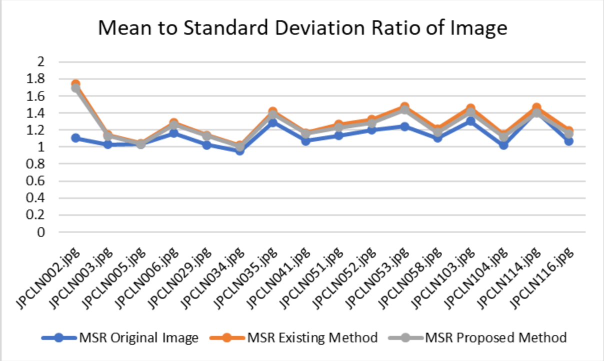


Fig. 9. Comparison between Image MSR of original image, MSR of budapest bone shadow eliminated images dataset and MSR of proposed method.

TABLE II. MEAN TO STANDARD DEVIATION RATIO (MSR)

Image	Original Image	Existing Method*	Proposed Method
JPCLN002.jpg	1.105865187	1.73929548	1.69239946
JPCLN003.jpg	1.029805377	1.143003543	1.130802017
JPCLN005.jpg	1.034735271	1.042355194	1.036175145
JPCLN006.jpg	1.163311881	1.285875138	1.260937779
JPCLN029.jpg	1.026111427	1.139242171	1.128066868
JPCLN034.jpg	0.953965214	1.021915224	1.00740993
JPCLN035.jpg	1.28773162	1.418071626	1.386284991
JPCLN041.jpg	1.072865683	1.166067981	1.158654616
JPCLN051.jpg	1.13379032	1.264658084	1.22893067
JPCLN052.jpg	1.19827801	1.32425825	1.278240793
JPCLN053.jpg	1.242163322	1.475137804	1.436426986
JPCLN058.jpg	1.106253489	1.215176321	1.17012551
JPCLN103.jpg	1.30222145	1.459562579	1.409503867
JPCLN104.jpg	1.02032688	1.146969601	1.117281164
JPCLN114.jpg	1.413209042	1.465449364	1.396894366
JPCLN116.jpg	1.066277074	1.195911959	1.158528356

*Existing Algorithm-Bone Shadow Eliminated X-Ray image dataset by Budapest University

c) Test 2.3 Universal image quality index: The image similarity across Rib suppression using existing algorithm and proposed method can be measured.

Let $I = \{i_n \mid n = 1, 2, ..., N\}$ and $J = \{j_n \mid n = 1, 2, ..., N\}$ be the original and Rib suppressed image signals, respectively. Universal image quality index (UQI) [39] is defined as

$$Q = \frac{4\sigma_{ij}\bar{\imath}\,\bar{\jmath}}{\left(\sigma_i^2+\sigma_j^2\right)[(\bar{\imath})^2+(\bar{\jmath})^2]} \qquad (9)$$

The average pixel intensity of the original image is denoted by $\bar{\imath}$ and $\bar{\jmath}$ indicates average pixel intensities of the Rib suppressed images. The minimum value of Q is 0 and maximum value of Q is 1, where the value 1 is considered to be the best value. It is only possible if and only if $i_n = j_n$ for all $n = 1, 2, ..., N$.

Table III shows comparison of MSR of JSRT original images, Budapest University Images and result Images of the proposed method while Fig. 10 shows it graphically.

TABLE III. UNIVERSAL QUALITY INDEX (IQI)

Image	Existing Algorithm	Proposed Method
JPCLN002.jpg	0.766393412	0.786052698
JPCLN003.jpg	0.559502642	0.6154612
JPCLN005.jpg	0.5484891	0.62938914
JPCLN006.jpg	0.888165856	0.918416024
JPCLN029.jpg	0.550828852	0.59797016
JPCLN034.jpg	0.951746077	0.982698381
JPCLN035.jpg	0.64269835	0.664188397
JPCLN041.jpg	0.518551926	0.552074836
JPCLN051.jpg	0.530794333	0.55762417
JPCLN052.jpg	0.888784908	0.921368415
JPCLN053.jpg	0.87828747	0.899798241
JPCLN058.jpg	0.875172812	0.918980863
JPCLN103.jpg	0.838295369	0.843335781
JPCLN104.jpg	0.707448061	0.734453776
JPCLN114.jpg	0.867494523	0.905721354
JPCLN116.jpg	0.640720912	0.691021895

*Existing Algorithm-Bone Shadow Eliminated X-Ray image dataset by Budapest University

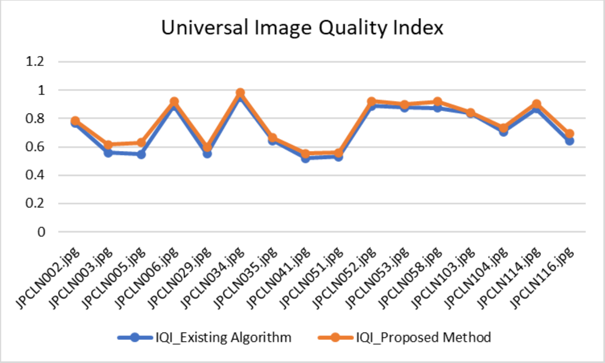


Fig. 10. Universal image quality using budapest bone shadow eliminated images dataset and proposed method.

3) *Test-3*

a) *Test 3.1 Mean to standard deviation ratio of nodule area:* In Test-3, the MSR is calculated for the nodule region, which takes into account its surroundings as well. The MSR of the nodule area calculated using the suggested method is compared to the MSR of the nodule area calculated using JSRT's original dataset and Budapest University's Bone Shadow Eliminated Image dataset. The Fig. 11(c) shows the sample result of the nodule area after suppressing ribs shadow from the chest radiograph using proposed method.

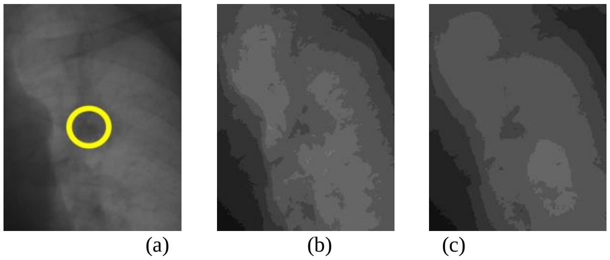


Fig. 11. Cropped nodule area (a) Nodule area in the original chest radiograph, (b) Nodule area in budapest bone shadow eliminated images dataset (c) Nodule area in the proposed method.

V. RESULT ANALYSIS

To evaluate the performance of the proposed method, a total of three tests are conducted. In Test-1, fifty images from the JSRT dataset with subtlety levels ranging from 5 to 2 are utilized. After preprocessing, the lungs are segmented using ASM. The ribs are removed using the proposed method (CWAMO), and finally, candidate nodules are extracted using circular window-based local contrast thresholding (CWLCT). From the results, it is observed that in 89% of the chest radiographs, true positive nodules are detected accurately. Test-2 was conducted to evaluate the performance of the proposed method at the image level. In this test, three measures were used for evaluation, namely PSNR, MSR, and IQI. Table I and Table III show the PSNR and IQI comparison between the existing algorithm and the proposed method. Fig. 8 and 10 depict it graphically. Table II shows the MSR comparison between the original image (with rib shadows), the bone shadow-eliminated image by Budapest University, and the proposed method (CWAMO). Fig. 9 shows the above comparison graphically. Four sample images ranging in subtlety from 5 to 2 were used for each of the measures.

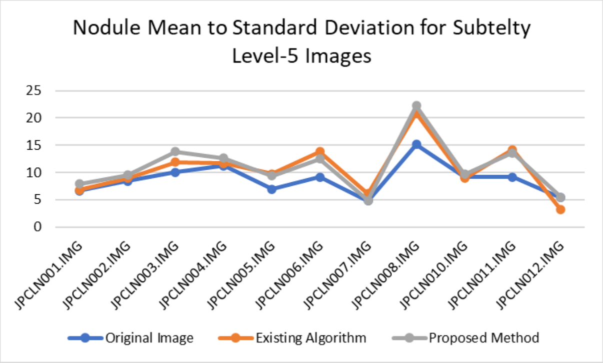


Fig. 12. Comparison between nodule MSR of original image, MSR of budapest bone shadow eliminated images dataset and MSR of proposed method for subtlety level-5.

As shown in Fig. 12, 13, 14, and 15, Test-3 employs ten sample images from each subtlety level (Level 5 to Level 2). It is observed that for all the above subtlety-level images, the proposed method performs equally with the existing algorithm.

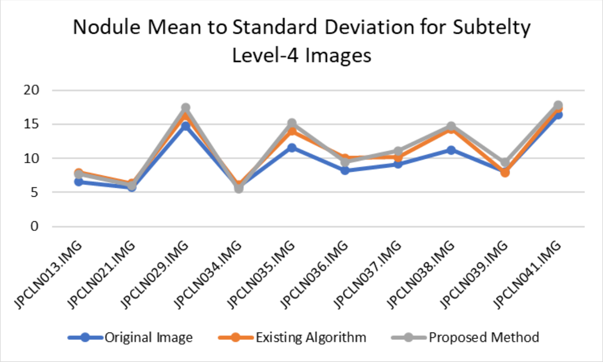


Fig. 13. Comparison between nodule MSR of original image, MSR of budapest bone shadow eliminated images dataset and MSR of proposed method for subtlety level-4.

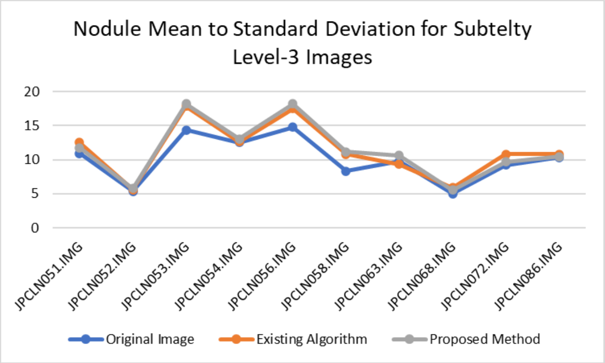


Fig. 14. Comparison between nodule MSR of original image, MSR of budapest bone shadow eliminated images dataset and MSR of proposed method for subtlety level-3.

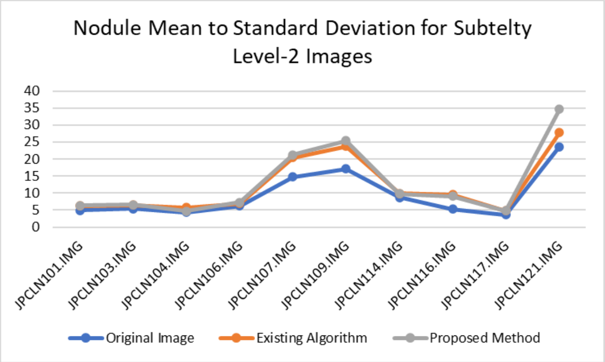


Fig. 15. Comparison between nodule MSR of original image, MSR of budapest bone shadow eliminated images dataset and MSR of proposed method for subtlety level-2.

For all three image quality measures, the proposed method performs over and above the existing algorithm. The ROI for evaluation in Test-3 was the nodule area, including its surroundings.

VI. Conclusion

The proposed work of bone suppression on chest radiographs using proposed method (CWAMO) achieves the goal of enhancing the accuracy of nodule detection in the chest radiographs. The first experimental setting, known as Test-1, uses chest radiograph images with modest levels 5 to 2 to detect actual positive nodules with an accuracy of 89%. This results in a significant reduction in the number of false positive nodules found in each image. During Test-2, the peak signal-to-noise ratio (PSNR), the mean-to-standard deviation ratio (MSR), and the universal image quality index (IQI) are analyzed for the entire image and compared with existing algorithm. The suggested technique performs better than the existing algorithm across all three parameters. Test-3 calculates the nodule MSR and then compares it to the Bone Shadow Eliminated Dataset created by Budapest University as well as the original chest radiographs. When compared to the existing bone shadow eliminated dataset, the new algorithm achieved an improvement in nodule area contrast of 3.83% and 23.94% shadow. This strategy results in an improvement in the visibility of nodules on chest radiographs. In conclusion, the utilization of CWAMO not only eliminates the shadow that is cast by the ribs on the chest radiograph, but it also assists in the localization of the nodules that are obscured by the shadows cast by the bones on the chest radiograph.

References

[1] T. Cootes, "An Introduction to Active Shape Models," in Image Processing and Analysis, Ed.R.Baldock and J.Graham, Eds. Oxford University Press, 2000, pp. 223–248.

[2] T. F. Cootes, A. Hill, C. J. Taylor, and J. Hastam, "Use of active shape models for locating structures in medical images," Image Vis Comput, vol. 12, no. 6, pp. 355–365, 1994.

[3] J.-Shyang. Pan, B.-Long. Guo, A. Abraham, Xi'an dian zi ke ji da xue., Guo li Gaoxiong ying yong ke ji da xue., and IEEE Computer Society., "Lung segmentation for chest radiograph by using adaptive active shape models," in 2009 Fifth International Conference on Information Assurance and Security, 2009, pp. 383–386. Accessed: Feb. 20, 2023. [Online]. Available: 10.1109/IAS.2009.353.

[4] B. Van Ginneken, A. F. Frangi, J. J. Staal, B. M. Ter Haar Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," IEEE Trans Med Imaging, vol. 21, no. 8, pp. 924–933, Aug. 2002, doi: 10.1109/TMI.2002.803121.

[5] S. Junji et al., "A Development of a Digital Image Database for Chest Radiographs With and Without a Lung Nodule: Receiver Operating Characteristic Analysis of Radiologists' Detection of Pulmonary Nodules," American journal of roentgenology, vol. 174, pp. 71–74, 2000, [Online]. Available: www.macnet.or.jp/jsrt2/.

[6] S. Juhász, Á. Horváth, L. Nikházy, and G. Horváth, "Segmentation of Anatomical Structures on Chest Radiographs," in Medicon 2010, IFMBE Proceedings 29, 2010, pp. 359–362. [Online]. Available: www.springerlink.com.

[7] A. Bilal, R. Tahir, A. ,U. , K. Mohammad, R. Abdur, and A. Saghir, "Rib Suppression in chest radiographs using ICA," Information Technology Journal, vol. 6, no. 7, pp. 1085–1089, 2007.

[8] B. van Ginneken, M. B. Stegmann, and M. Loog, "Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database," Med Image Anal, vol. 10, no. 1, pp. 19–40, Feb. 2006, doi: 10.1016/j.media.2005.02.002.

[9] M. Loog and B. Van Ginneken, "Segmentation of the posterior ribs in chest radiographs using iterated contextual pixel classification," IEEE Trans Med Imaging, vol. 25, no. 5, pp. 602–611, May 2006, doi: 10.1109/TMI.2006.872747.

[10] L. Hogeweg, C. I. Sánchez, P. A. de Jong, P. Maduskar, and B. van Ginneken, "Clavicle segmentation in chest radiographs," Med Image Anal, vol. 16, no. 8, pp. 1490–1502, 2012, doi: 10.1016/j.media.2012.06.009.

[11] Á. Horváth, G. Orbán, Á. Horváth, and G. Horváth, "An X-ray CAD system with ribcage suppression for improved detection of lung lesions," Periodica Polytechnica Electrical Engineering and Computer Science, vol. 57, no. 1, pp. 19–33, 2013, doi: 10.3311/PPee.2079.

[12] Z. Huo et al., "Bone suppression technique for chest radiographs," in Medical Imaging 2014: Image Perception, Observer Performance, and Technology Assessment, Mar. 2014, vol. 9037. doi: 10.1117/12.2043754.

[13] X. Li, S. Luo, and Q. Hu, "An Automatic Rib Segmentation Method on X-Ray Radiographs," in International Conference on Multimedia Modeling, 2015, vol. 8935, pp. 128–139.

[14] W. Haiping and Z. Guodong, "Rib Segmentation in Chest Radiographs by Support Vector Machine," in International Conference on Education, Management, Computer and Society (EMCS 2016), 2016.

[15] N. Matsubara, A. Teramoto, K. Saito, and H. Fujita, "Bone suppression for chest X-ray image using a convolutional neural filter," Australas Phys Eng Sci Med, vol. 43, 2019, doi: 10.1007/s13246-019-00822-w.

[16] W. Wang et al., "MDU-Net: A Convolutional Network for Clavicle and Rib Segmentation from a Chest Radiograph," J Healthc Eng, vol. 2020, 2020, doi: 10.1155/2020/2785464.

[17] G. Ren et al., "Deep learning-based bone suppression in chest radiographs using CT-derived features: A feasibility study," Quant Imaging Med Surg, vol. 11, no. 12, pp. 4807–4819, Dec. 2021, doi: 10.21037/qims-20-1230.

[18] S. Rajaraman, G. Zamzmi, L. Folio, P. Alderson, and S. Antani, "Chest X-Ray Bone Suppression for Improving Classification of Tuberculosis-Consistent Findings," Diagnostics (Basel), vol. 11, no. 5, 2021, doi: 10.3390/diagnostics11050840.

[19] S. Rajaraman, G. Cohen, L. Spear, L. Folio, and S. Antani, "DeBoNet: A deep bone suppression model ensemble to improve disease detection in chest radiographs," PLoS One, vol. 17, no. 3, Mar. 2022, doi: 10.1371/journal.pone.0265691.

[20] S. Chen and K. Suzuki, "Computerized detection of lung nodules by means of 'virtual dual-energy' radiography," IEEE Trans Biomed Eng, vol. 60, no. 2, pp. 369–378, 2013, doi: 10.1109/TBME.2012.2226583.

[21] K. Bae, D. Y. Oh, I. D. Yun, and K. N. Jeon, "Bone Suppression on Chest Radiographs for Pulmonary Nodule Detection: Comparison between a Generative Adversarial Network and Dual-Energy Subtraction," Korean J Radiol, vol. 23, no. 1, pp. 139–149, Feb. 2022, doi: 10.3348/kjr.2021.0146.

[22] P. Campadelli, E. Casiraghi, and D. Artioli, "A fully automated method for lung nodule detection from postero-anterior chest radiographs," IEEE Trans Med Imaging, vol. 25, no. 12, pp. 1588–1603, Dec. 2006, doi: 10.1109/TMI.2006.884198.

[23] **00**Z. Shi, J. Bai, K. Suzuki, L. He, Q. Yao, and T. Nakamura, "A Method for Enhancing Dot-like Regions in Chest X-rays Based on Directional Scale LoG Filter," Journal of Information & Computational Science, vol. 7, no. 8, pp. 1689–1696, 2010, [Online]. Available: http://www.joics.com.

[24] S. A. Patil and V. R. Udupi, "Chest X-ray features extraction for lung cancer classification," J Sci Ind Res (India), vol. 69, pp. 271–277, 2010.

[25] J. Wei, Y. Hagihara, A. Shimizu, and H. Kobatake, "Optimal image feature set for detecting lung nodules on chest X-ray images," Computer Assisted Radiology and Surgery, Springer, 2002, [Online]. Available: https://doi.org/10.1007/978-3-642-56168-9_118.

[26] K. Mya, M. Tun, and A. Soe Khaing, "Feature Extraction and Classification of Lung Cancer Nodule using Image Processing Techniques," International Journal of Engineering Research & Technology , vol. 3, no. 3, 2014, [Online]. Available: www.ijert.org.

[27] A. M. Al Gindi, E. A. Rashed, and M.-S. M. Mostafa, "Development and Evaluation of a Computer-Aided Diagnostic Algorithm for Lung Nodule Characterization and Classification in Chest Radiographs using Multiscale Wavelet Transform," Journal of American Science, vol. 10, no. 2, pp. 1545–1003, 2014, [Online]. Available: http://www.jofamericanscience.orghttp://www.jofamericanscience.org.1 4.

[28] A. M. R. Schilham, B. Van Ginneken, and M. Loog, "Multi-scale Nodule Detection in Chest Radiographs," in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2003, pp. 602–609. [Online]. Available: http://www.isi.uu.nl.

[29] S. Mehmet and S. Bulent, "Survey over image thresholding techniques and quantitative performance evaluation," J Electron Imaging, vol. 13, no. 1, pp. 146–165, Jan. 2004, doi: 10.1117/1.1631316.

[30] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh, "A New Local Adaptive Thresholding Technique in Binarization," International Journal of Computer Science Issues, vol. 8, no. 6, Nov. 2011, [Online]. Available: www.IJCSI.org.

[31] F. Nie, Y. Wang, M. Pan, G. Peng, and P. Zhang, "Two-dimensional extension of variance-based thresholding for image segmentation," Multidimens Syst Signal Process, vol. 24, no. 3, pp. 485–501, Sep. 2013, doi: 10.1007/s11045-012-0174-7.

[32] S. K. Chaya Devi and T. Satya Savithri, "Review: On segmentation of nodules from posterior and anterior chest radiographs," Int J Biomed Imaging, vol. 2018, pp. 1–11, Oct. 2018, doi: 10.1155/2018/9752638.

[33] L. Zheng and Y. Lei, "A Review of Image Segmentation Methods for Lung Nodule Detection Based on Computed Tomography Images," in MATEC Web of Conferences, Nov. 2018, vol. 232. doi: 10.1051/matecconf/201823202001.

[34] C. H. Liang, Y. C. Liu, M. T. Wu, F. Garcia-Castro, A. Alberich-Bayarri, and F. Z. Wu, "Identifying pulmonary nodules or masses on chest radiography using deep learning: external validation and strategies to improve clinical practice," Clin Radiol, vol. 75, no. 1, pp. 38–45, Jan. 2020, doi: 10.1016/j.crad.2019.08.005.

[35] S. Bharati, P. Podder, and M. R. H. Mondal, "Hybrid deep learning for detecting lung diseases from X-ray images," Inform Med Unlocked, vol. 20, p. 100391, Jan. 2020, doi: 10.1016/j.imu.2020.100391.

[36] J. Soltani-Nabipour, A. Khorshidi, and B. Noorian, "Lung tumor segmentation using improved region growing algorithm," Nuclear Engineering and Technology, vol. 52, no. 10, pp. 2313–2319, Oct. 2020, doi: 10.1016/j.net.2020.03.011.

[37] M. Horry, S. Chakraborty, B. Pradhan, M. Paul, D. Gomes, and A. Ul-Haq, "Deep Mining Generation of Lung Cancer Malignancy Models from Chest X-ray Images," Sensors (Basel), vol. 21, no. 19, p. 6655, 2021.

[38] J. Lee, S. R. Pant, and H.-S. Lee, "An Adaptive Histogram Equalization Based Local Technique for Contrast Preserving Image Enhancement," The International Journal of Fuzzy Logic and Intelligent Systems, vol. 15, no. 1, pp. 35–44, Mar. 2015, doi: 10.5391/ijfis.2015.15.1.35.

[39] J. Bernsen, "Dynamic thresholding of gray-level images," in Proceedings International Conference on Pattern Recognition, Jan. 1986.

AUTHORS' PROFILE

Dnyaneshwar Kanade Research Scholar, School of Engineering and Technology, Sandip University, Nasik, India. He received the B.E. degree in Electronics Engineering from the Savitribai Phule Pune University (Formerly University of Pune), in 1996 and the M.E degree from Babasaheb Ambedkar Marathwada University, Aurangabad, in 2007. He has 3 Granted Patents on his name. He has published many papers in national and international journals. His research interest includes biomedical signal and image processing, Embedded System Design and IOT. He is a Lifetime member of Indian Society for Technical Education (ISTE).

Dr. Jagdish Helonde Professor, Department of Electrical & Electronics Engineering, School of Engineering and Technology, Sandip University, Nasik, India. He received the B.E. degree in Electrical Engineering from the Govt. College of Engineering, in 1981 and the M.Tech degree from VNIT Nagpur in 1983. He obtained Ph.D. degree in Electrical Engineering from RTM Nagpur University, Nagpur, India in 2002. He has a teaching experience of 40 years. He has guided more than 10 Ph.D scholars. He has published three patents. He has presented 14 papers in National Conferences and 34 in International Conferences. He has published 52 papers in international journals. His research interest includes biomedical signal and image processing, energy management, statistical signal processing, non-conventional energy sources etc. He is a Lifetime member of Institution of Engineers (India) and Indian Society for Technical Education (ISTE).

Disease Identification in Crop Plants based on Convolutional Neural Networks

Orlando Iparraguirre-Villanueva[1], Victor Guevara-Ponce[2], Carmen Torres-Ceclén[3], John Ruiz-Alvarado[4], Gloria Castro-Leon[5], Ofelia Roque-Paredes[6], Joselyn Zapata-Paulini[7], Michael Cabanillas-Carbonell[8]

Facultad de Ingeniería y Negocios, Universidad Privada Norbert Wiener, Lima, Perú[1]
Escuela de Posgrado, Universidad Ricardo Palma, Lima, Perú[2, 6]
Facultad de Ingeniería, Universidad Católica los Ángeles de Chimbote, Ancash, Perú[3]
Facultad de Ingeniería, Universidad Tecnológica del Perú, Lima, Perú[4]
Facultad de Ingeniería y Gestión, Universidad Nacional Tecnológica de Lima Sur, Lima, Perú[5]
Escuela de Posgrado, Universidad Continental, Lima, Perú[7]
Facultad de Ingeniería, Universidad Privada del Norte, Lima, Perú[8]

***Abstract*—The identification, classification and treatment of crop plant diseases are essential for agricultural production. Some of the most common diseases include root rot, powdery mildew, mosaic, leaf spot and fruit rot. Machine learning (ML) technology and convolutional neural networks (CNN) have proven to be very useful in this field. This work aims to identify and classify diseases in crop plants, from the data set obtained from Plant Village, with images of diseased plant leaves and their corresponding Tags, using CNN with transfer learning. For processing, the dataset composing of more than 87 thousand images, divided into 38 classes and 26 disease types, was used. Three CNN models (DenseNet-201, ResNet-50 and Inception-v3) were used to identify and classify the images. The results showed that the DenseNet-201 and Inception-v3 models achieved an accuracy of 98% in plant disease identification and classification, slightly higher than the ResNet-50 model, which achieved an accuracy of 97%, thus demonstrating an effective and promising approach, being able to learn relevant features from the images and classify them accurately. Overall, ML in conjunction with CNNs proved to be an effective tool for identifying and classifying diseases in crop plants. The CNN models used in this work are a very good choice for this type of tasks, since they proved to have a very high performance in classification tasks. In terms of accuracy, all three models are very accurate in image classification, with an accuracy of over 96% with large data sets.**

***Keywords*—CNN; identification; models; pathogen; plant; classification; machine learning**

I. Introduction

The identification and classification of plant diseases is essential for agriculture, as the presence of pathogens can cause significant damage and reduce production [1]. Some of the most common diseases include: root rot, which is caused by fungi that attack the roots of the plant; powdery mildew, which is caused by fungi that affect the leaves and flowers of the plant; mosaic, which is caused by viruses that affect the growth and appearance of the leaves of the plant; leaf spot, which is caused by bacteria that affect the leaves of the plant; and fruit rot, which is caused by fungi or bacteria that affect the fruit of the plant [2], [3]. It is important to identify the disease as early as possible to take measures to combat it, either by fungicide treatments or traditional measures that help prevent the spread of the disease. Since ancient times, agriculture has been an important source of food [4], hence the importance of identifying diseases and being able to treat them, since diseases are the cause of a great loss of crop production, since about 36% of crops are lost because of not identifying diseases at early stages [5]. Early-stage identification of plant diseases can greatly reduce their considerable impact. This involves the use of computer technology in agriculture to help identify these diseases [6]. ML technology has proven to be useful in plant disease identification. One specific technique that has proven to be effective is transfer learning (TL) supported by CNN [7]. The idea behind TL allows models to leverage what they have previously learned to improve their performance on new tasks [8]. CNNs can be used to identify plant diseases by analyzing plant images. The CNN is trained on a labeled image dataset, where labels correspond to different types of diseases [3], [9]. During the training process, the CNN learns to recognize patterns in the images that are indicative of a particular disease. Once trained, the CNN can be used to classify new plant images as carrying or not carrying a particular disease [10]. This approach can be very effective in identifying plant diseases, as it allows the use of visual information, which can be very useful in identifying certain types of diseases that may not be evident from other types of data [11]. In this work we use the transfer learning models: DenseNet-201, ResNet-50 and Inception-v3. DenseNet-201 is a CNN with 201 layers deep. ResNet-50 is a CNN with 50 layers deep; Inception-v3 is a CNN with 48 layers deep. The models work with their own training architecture; therefore, they deliver different results. The models have been trained with Google's ImageNet dataset. Both models are characterized by training with large volumes of images and achieving a high level of accuracy [12]. This is due to their ability to learn specific image features through convolution and pooling layers [13], [14]. However, the specific performance of a model depends on several factors, such as the quality and quantity of training data, model architecture, and parameter optimization. This work aims to identify and classify crop plant diseases from PlantVillage dataset, diseased plant leaf images and their corresponding labels, using CNN with Transfer Learning.

II. Related Work

Lately, researchers and scientists related to the agricultural industry and computer science have conducted different studies related to plant diseases using ML techniques, specifically with CNN. For example, in [15] they developed a model to detect, quantify the severity and classify plant images in different species. To illustrate this, in [16] they developed a hybrid application with web technology and CNN models to detect plant diseases in real time. The model used in the web application achieved an accuracy performance rate of 0.9935 for which they used 9914 training arguments. Also, in [17], [18] implemented a CNN for automatic identification and classification of weeds using a mixed crop land, this model was trained in five different epochs; 10k, 20k, 100k, 200k and 242k images, with the purpose of automatic identification and classification of diseases. The results achieved for each epoch: in 10k the weed accuracy reached 0.67; in 20k the weed accuracy reached 0.962; in 100k the accuracy reached 0.983; in 200k the accuracy reached 0.95 and in 242k the accuracy improved significantly. Similarly, the authors in [18], [19] proposed a model for classifying disease affected leaves, for which they used images and obtained directly from cotton crop fields. They used the CNN DarkNet-19, which obtained a performance rate in accuracy of 0.91. Plant diseases directly affect food security, thus decreasing production. The identification of these diseases is the first important step for their treatment. That is why, in [20] they developed a mobile application with CNN to diagnose in real time 26 diseases of 14 crop species. It was validated and tested with 87k images, divided into 38 classes. The model achieved a classification accuracy of 0. 957. Similarly, in [21] they proposed a CNN with EfficientNet model to identify and classify images with diseases into categories. The model obtained an optimal performance rate of 0.9972 accuracy. Also, in [22] through the ResNet-50 model they classified diseases, applying CNN techniques for image prediction. In the same line, [23] used computer vision and ML technology to detect diseases in field crops. The proposed model achieved an accuracy of 0.978 in detecting four crop plant diseases. CNNs have contributed significantly to the detection of plant pathogens. As such, in [24] performed a work with CNNs using different optimization algorithms for detection. They used 5571 manually collected images, the model achieved an accuracy performance of 0.9259. Image recognition through ML is an active area by scientists and researchers. For example, in [25] they proposed a semi-automatic algorithm to detect diseases in crop plants, making use of CNNs. They used the KijaniNet model, who achieved an accuracy performance of 0.8448 and 0.6257.

III. Methodology

This section presents the theoretical basis of the DenseNet-201, ResNet-50 and Inception-v3 models and the development process to identify and classify crop plant diseases. For this purpose, initially the image dataset of infected and healthy plants, collected from PlantVillage, is loaded. This is followed by data cleaning, exploratory analysis and then training and validation of the model using the CNN architecture, which is particularly well suited for image recognition tasks.

A. CNN Architecture

The architecture of a CNN consists of several layers, including input layers, convolution layers, pooling layers, fully connected layers, and an output layer. For this case, the input layer is where the input image is provided to the CNN. Convolution layers perform filtering operations on the input data, using filters that are slid over the image to detect specific patterns. Pooling layers reduce the dimensionality of the data by grouping values from a subregion of the image and replacing them with a single value, such as the maximum or average value. The fully connected layers are used to classify the patterns detected in the previous layers. The output layer provides the final predictions of the CNN, as shown in Fig. 1. In general, CNNs are used for computer vision tasks.

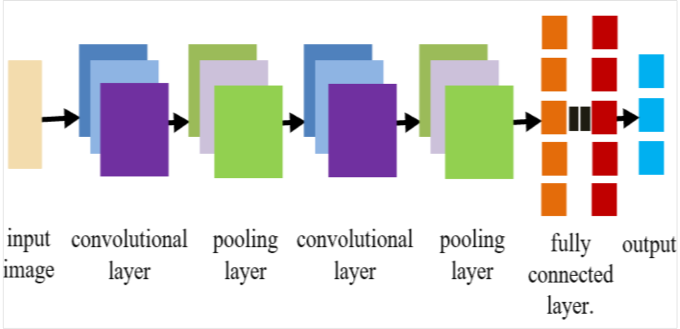


Fig. 1. CNN architecture.

B. Convolutional Layer

A convolution layer is a layer in a CNN that performs filtering operations on the input images. The layer is composed of a set of filters, each of which is slid over the input image and applied to a subregion of the input image [26]. Each filter looks for specific patterns in the image, such as edges or textures [27]. A filter is a small matrix of numbers that is multiplied with the image pixels in a specific window, and the products are summed. In general, convolution layers are used to detect patterns in the input data and to extract relevant features from these data.

C. Activation Functions

In CNNs, the activation function is a mathematical equation that is defined at the output layers of a neuron before being sent to the next layers of the network. The activation function aims to regulate the output of a neuron, allowing the network to learn more complex patterns [28] [29]. Some commonly used activation functions are: Sigmoid, ReLU, tanh and Softmax. Each activation function has its own characteristics and advantages, and the choice of the appropriate activation function depends on the specific problem and data set.

D. Pooling Layer

A network layer in a CNN architecture that is used to group a set of inputs into similar groups. This is achieved through the use of clustering algorithms. Pooling layer is commonly used in unsupervised learning tasks [30], such as image segmentation.

E. Fully Connected Layer

It is known as the densely connected layer. In this layer, all neurons are connected to all neurons in the previous layer, and this layer is part of a neural network layer [31]. In other words, each neuron in a fully connected layer receives an input from each neuron in the previous layer and produces an output that is passed to each neuron in the next layer. Fully connected layers are commonly used in neural networks for tasks such as image classification and natural language processing.

F. Reducing Overfitting

The overfitting technique was applied to reduce the complexity of the model and to regulate and thus prevent the model weights from becoming too large. For this case dropout was used in the neural network layers [32]. This is for the purpose of optimizing performance since overfitting is a common problem in ML in which a model is overfitted to the training data and, as a result, may exhibit performance deficiency.

G. Transfer Learning

TL is a process in which knowledge acquired in one task is used to improve performance in a different task. It is a form of ML in which the model reuses what it has learned to improve its performance rate on another task [33]. TL is especially useful in ML as it allows models to leverage what they have previously learned to improve their performance on new tasks; the process of TL is seen in Fig. 2.

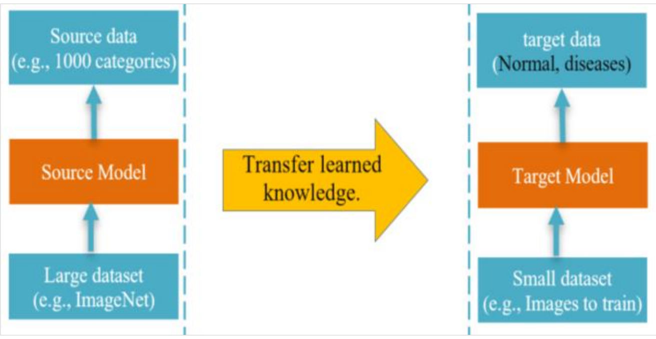


Fig. 2. Transfer learning process.

H. Model architecture

Three models were trained with images with and without disease. Each of the models used in this work is described below.

I. DenseNet-201

DenseNet-201 is characterized by the use of dense blocks, which are layered blocks in which each layer is connected to all other layers in the block [27]. This allows feature reuse and helps mitigate the problem of leakage gradients that can occur in deep neural networks. The number "201" refers to the number of convolutional layers in the network. The model architecture is shown in Fig. 3.

J. ResNet-50

ResNet-50 is a variant of the CNN ResNet. The key innovation of the ResNet-50 is the use of residual connections, which allow the network to create new residual functions for input to the layers, rather than attempting to learn the original unreferenced functions [34], this helps mitigate the problem of gradient loss that can occur in deep neural networks [30] [35]. The main use of this network is in image classification. The number "50" refers to the number of layers as presented in Fig. 4.

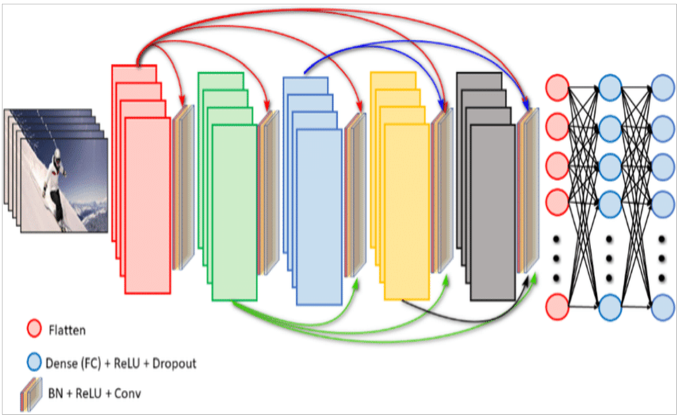


Fig. 3. Architecture of the DenseNet-2021 model.

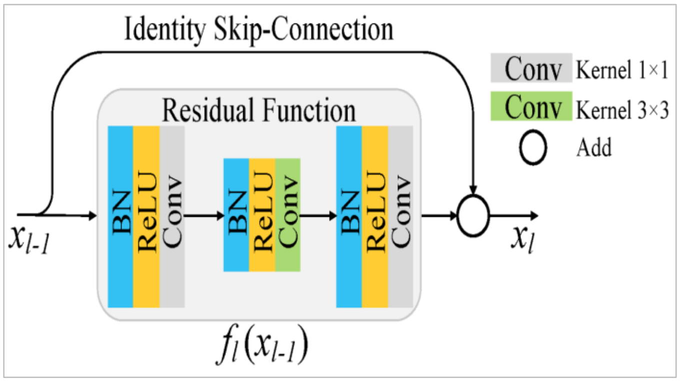


Fig. 4. ResNet-50 architecture.

K. Inception-v3

Inception-v3 is a CNN designed for object recognition in images. It uses a combination of filter layers of different sizes and depths to capture features at different scales and levels of abstraction in the input images [36] [37]. The architecture, as depicted in Fig. 5, includes an "Inception" mechanism that combines multiple filtering operations in a single layer to reduce the number of attributes and improve computational efficiency. Inception-v3 has been trained on a large set of images (ImageNet) and has been widely used in different cases of image classification and identification, object recognition, object detection, etc.

L. Understanding Data

There are a variety of diseases that affect plants, among the most common are: Mildew: caused by fungi that appear as white spots on the leaves; powdery mildew: caused by fungi that appear as a kind of powder on the leaves; Rust and Mosaic: are two diseases that are caused by fungi or insects that cause spots and deformations on the leaves, as shown in Fig. 6. These diseases can be treated with fungicides and pesticides. Also, it is important to take preventive measures such as good ventilation or good soil drainage. In this work we used an original dataset, obtained from the PlantVillage repository, and this is composed of approximately 87 thousand images of healthy and diseased crop leaves that are organized in 38 different categories. For training and validation the dataset is divided in a proportion of 80% for training and 20% for validation, and then a directory with 33 test images is created for prediction purposes.

The image set is composed of 38 disease types. Each of the diseases associated with the plant types is described below: [Tomato with blight, healthy tomato, healthy grape, orange with virus, healthy soybean, healthy peach, pumpkin with downy mildew, healthy potato, corn with blight leaf, young tomato with blight, tomato with leaf spot, strawberry with frost, apple with scab, tomato with yellow leaf, healthy cherry, healthy blueberry, healthy corn, tomato with bacterial spot, rotting apple, cherry with downy mildew, healthy apple, peach with bacterial spot, royal apple, tomato with objective spot, healthy bell pepper, grape with blight, tomato with mosaic virus, potato with blight, healthy strawberry, black rotten grape, early potato with blight, common corn, beetle grape, healthy raspberry, tomato with mildew, tomato with spider mites, bell pepper with bacterial spot].

From the types of diseases described above, the name of 14 plants is obtained, excluding healthy leaves: [corn with gray leaf spot, corn with rust, corn with fever, healthy potato, rice with leaf spot, potato with fever, healthy corn, potato, healthy rice, healthy rice, rice with hispa, rice with leaf blast, healthy wheat, young potato, red brown wheat and red yellow wheat] and 26 types of images with disease. With this, we proceed to import the dataset; then we process and perform an exploratory analysis, train the DenseNet-201, ResNet-50 and Inception-v3 models and then apply the optimizers.

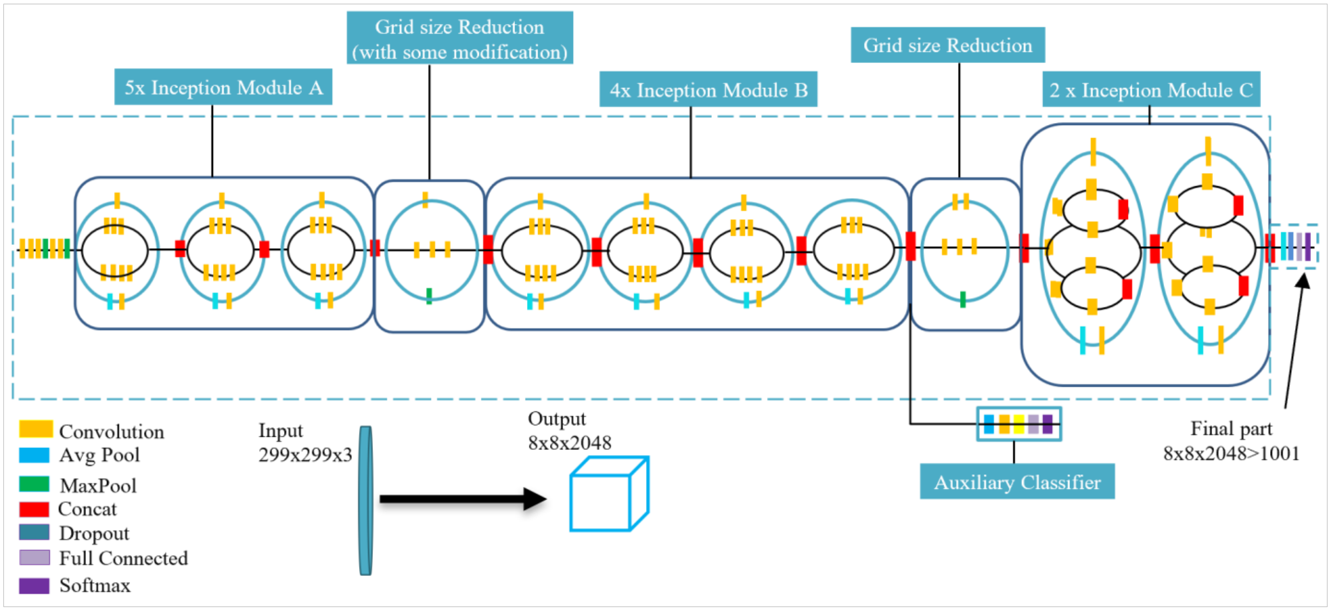


Fig. 5. Architecture of Inception-v3.

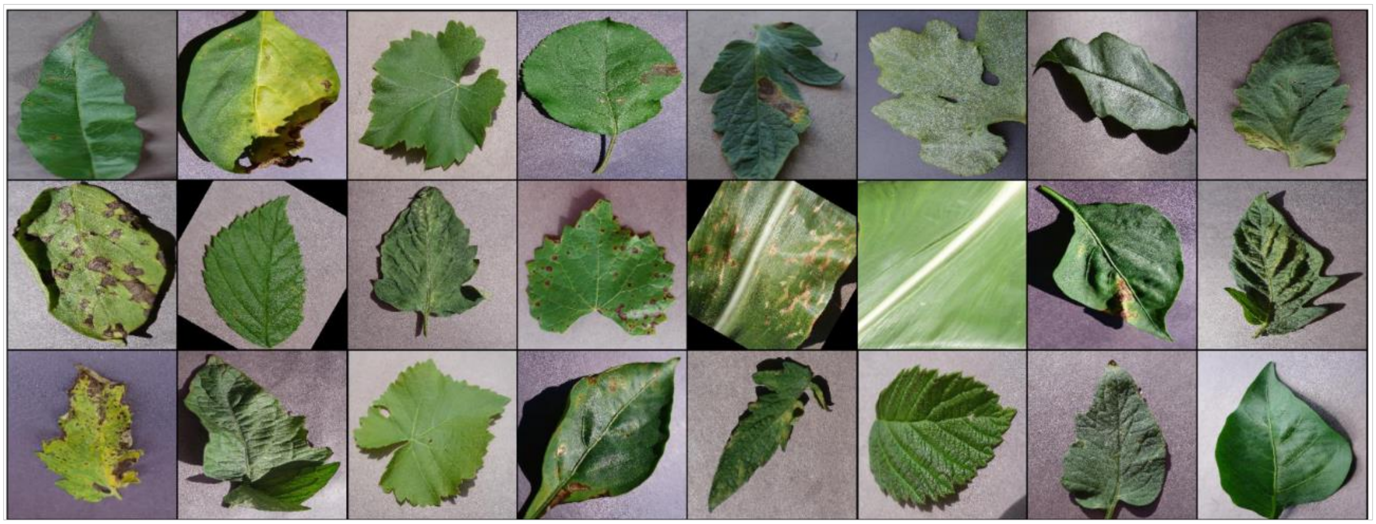


Fig. 6. Images for training.

M. Data Cleaning and Processing

The cleaning and processing process are important tasks in data analysis. Cleaning involves reviewing the data to detect and correct errors, duplicates, and missing values. Data processing involves transforming the data into a more suitable format for analysis, such as grouping, combining, or creating new variables. These tasks are essential to ensure that the results of the analysis are accurate and reliable. For the development of this work we used tools and libraries such as: the Python programming language, libraries such as numpy: for numerical calculations; pandas: to work with data frames; the pytorch module; matplotlib: to process information in graphs and images using tensors; torch.nn: to create neural networks; torch. utils.data: to load data; PIL: to load images; torch.nn.functional: function to calculate loss; torchvision.transforms: to transform images into tensors; torchvision.utils: for data verification; torchvision.datasets: to work with classes and images and torchsummary: to get the model summary.

N. Exploratory Data Analysis

Exploratory data analysis (EDA) is a process of investigating data using statistical tools and visualizations. The main objective of EDA is to understand the structure and distribution of the data, as well as to detect patterns and trends. EDA is an essential step in the data analysis process, as it provides a better understanding of the data before applying statistical and ML techniques. It can include techniques such as histograms, boxplots, scatterplots, among others, to visualize the data and detect patterns and trends. In addition, EDA can also help identify problems in the data, such as outliers or missing values, and make decisions on how to clean and process them before continuing with the analysis. Table I and Fig. 7 show the number of images for each disease.

The next step is to prepare the data for model training, this step is critical to ensure that the model performs optimally. This may include tasks such as normalization, scaling, transformation, and splitting into training and test sets. It is also very important to ensure that the data is in the correct format and is representative of the production data to avoid problems such as overfitting. It is at this stage that libraries become relevant, for example, torchvision.datasets(), is used to load the image dataset. After loading the data, the pixel values of each image (0-255) are transformed to 0 - 1, since CNNs understand much better when the data is normalized. The pixel matrix is converted to a torchTensor and then divided by 255. For example, the image can have the following form (3,256,256), where 3 is the number of channels (RGB) and 256x256 is the width and height of the image. Also, we have the batch_size library (), it allows to count the total number of images given in the CNN input. For example, if we have 1500 image samples for training and we need to set up a batch size of 150 samples from the training dataset, then what the network does is to randomly take the 150 samples and retrains the network, this process continues until all the samples in the network are propagated. Similarly, dataLoader (), the Liberian dataLoader that comes with PyTorch load data in parallel from the dataset. It also provides a convenient way to iterate over the dataset, in small batches, with the ability to shuffle the data before each epoch with the option to use multiple threads for data loading. This library is also accompanied by the num_workers function, which allows to calculate the number of processes that generate batches in parallel. Another very important point is in the configuration, the variable 'shuffle=true', this is very important so that the batches between epochs do not resemble each other.

TABLE I. NUMBER OF IMAGES PER DISEASE

Disease	No. of images
tomato blight	1851
healthy tomato	1926
healthy grapes	1692
orange greening	2010
terveellistä soijaa	2022
Squash mildew	1736
Potato healthy	1824
Northern Corn	1908
Tomato early	1920
Tomato with spots	1745
corn with leaf spot on leaves	1642
strawberry leaf with virus	1774
Peach healthy	1728
apple with scab on leaves	2016
Tomato with yellow leaves due to virus	1961
tomato with bacterial spot	1702
black apple	1987
Blueberry healthy	1816
Moldy cherry	1683
Peach with bacteria	1838
Apple Cedar	1760
tomato target point	1827
Healthy peppers	1988
grapes with leaf spot	1722
potato with Tizon	1939
tomato with virus	1790
Healthy strawberry	1824
Apple healthy	2008
rotten grapes	1888
young pope	1939
Cherry healthy	1826
Corn Common	1907
Grape black	1920
Healthy raspberry	1781
Tomato with mushroom	1882
Tomato with mites	1741
Bell pepper with bacterial spots	1913
Young corn	1859

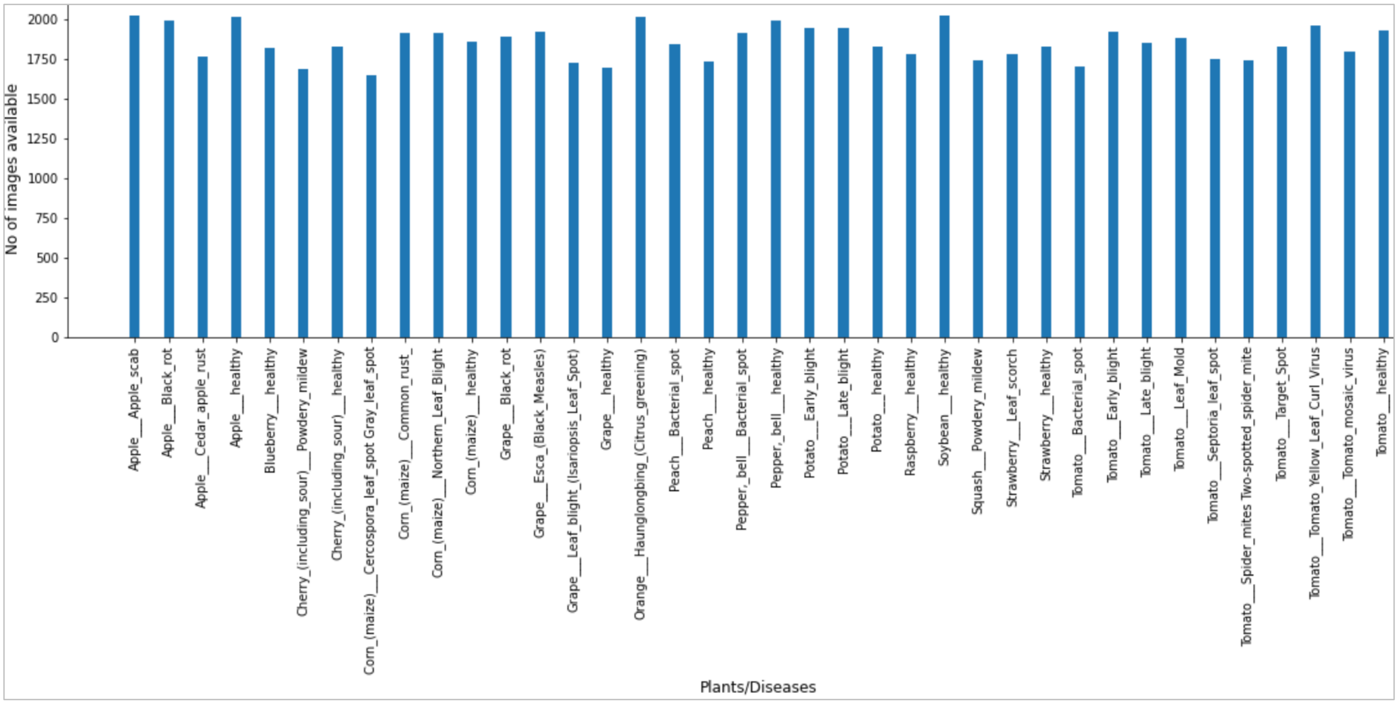


Fig. 7. Images for each type of plant disease.

O. Training and Testing of the Model

For the model training process, we worked with the training data set, in this case 80% of the images in the data set. During model training, the internal parameters were adjusted to minimize the loss function between model predictions. The training ends when it reaches a certain level of accuracy or when it reaches the maximum number of iterations. Meanwhile, to test the models, we worked with the difference of the training data set (20%). This stage is very important, since the aim is to evaluate the model's capability and to measure its accuracy. For which the following metrics were used, such as accuracy, recall Score-F1 and the performance curve (ROC). It is important to point out that both the training and testing processes are performed on different datasets, and another dataset was also used for validation. For training we chose to use GPUs instead of CPUs, considering that the volume of data is large, and a CPU is not able to respond to the demands, and GPUs are optimized to perform such tasks, as they can process multiple calculations simultaneously. In addition, they have a large number of cores to perform calculations and handle large amounts of data, which makes the memory bandwidth of a GPU the most suitable. Also, functions that help with the training are used, such as: Training_step(): this was used to find out how erroneous the model turned out to be after training. This function is not only an accuracy metric, but is necessary for the model to improve during training; Validation-setp(): this function is used to measure the accuracy across the threshold and is counted if the difference between the model prediction and the actual label is less than the threshold; Validation_epoch_end(): is used to track losses in validation accuracies and training losses after each epoch. However, it must be ensured not to be tracking the gradient; Epoch_end(): Used to print the validation accuracy losses and training losses and the learning rate after each epoch. We also use the accuracy function to calculate the overall accuracy of the models in the batch of the results. For example, in Fig. 8, we can see the training summary of the ResNet-50 model using Keras.

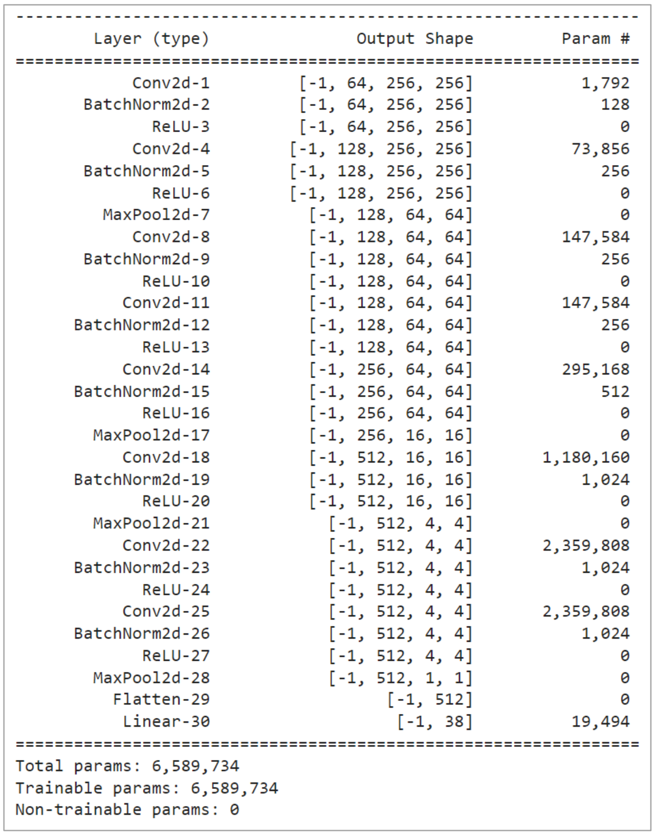

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 256, 256]           1,792
       BatchNorm2d-2         [-1, 64, 256, 256]             128
              ReLU-3         [-1, 64, 256, 256]               0
            Conv2d-4        [-1, 128, 256, 256]          73,856
       BatchNorm2d-5        [-1, 128, 256, 256]             256
              ReLU-6        [-1, 128, 256, 256]               0
         MaxPool2d-7          [-1, 128, 64, 64]               0
            Conv2d-8          [-1, 128, 64, 64]         147,584
       BatchNorm2d-9          [-1, 128, 64, 64]             256
             ReLU-10          [-1, 128, 64, 64]               0
           Conv2d-11          [-1, 128, 64, 64]         147,584
      BatchNorm2d-12          [-1, 128, 64, 64]             256
             ReLU-13          [-1, 128, 64, 64]               0
           Conv2d-14          [-1, 256, 64, 64]         295,168
      BatchNorm2d-15          [-1, 256, 64, 64]             512
             ReLU-16          [-1, 256, 64, 64]               0
        MaxPool2d-17          [-1, 256, 16, 16]               0
           Conv2d-18          [-1, 512, 16, 16]       1,180,160
      BatchNorm2d-19          [-1, 512, 16, 16]           1,024
             ReLU-20          [-1, 512, 16, 16]               0
        MaxPool2d-21            [-1, 512, 4, 4]               0
           Conv2d-22            [-1, 512, 4, 4]       2,359,808
      BatchNorm2d-23            [-1, 512, 4, 4]           1,024
             ReLU-24            [-1, 512, 4, 4]               0
           Conv2d-25            [-1, 512, 4, 4]       2,359,808
      BatchNorm2d-26            [-1, 512, 4, 4]           1,024
             ReLU-27            [-1, 512, 4, 4]               0
        MaxPool2d-28            [-1, 512, 1, 1]               0
          Flatten-29                  [-1, 512]               0
           Linear-30                   [-1, 38]          19,494
================================================================
Total params: 6,589,734
Trainable params: 6,589,734
Non-trainable params: 0
```

Fig. 8. ResNet-50 model training summary.

In the training process, some functions that are very useful should be taken into account, such as evaluate (), a function used for the validation process, and the function fit_one_cycle(), a function that performs the entire training process. Some techniques were also used such as: 1) Learning Rate Scheduling: allows using a learning rate after each training, instead of using a fixed rate; 2) Weight decay: regulation technique that prevents the weights from becoming too large to add a term to the loss function; 3) Gradient Clipping: this technique allows limiting the values of the decays to a small range to avoid undesired changes in the parameters due to large decay values, it is a simple but very effective technique. After training the models, the confusion matrix of each model is constructed, as in the case of the ResNet-50 model shown in Fig. 9. The confusion matrix is very important, since it helps to understand and compare the predicted data with the real data; however, for a better understanding, four basic concepts must be known: true and false positives and true and false negatives. That is, true positives and true negatives are simply the hits, while false positives and false negatives are the misses. This tool is important since it allows visualizing the performance of each of the models. Each column represents the number of predictions of each class, while the rows represent the actual values. Finally, the hits and misses are placed in each of the cells, Fig. 10, we can see a confusion matrix of 15 x 15, because we have 15 classes of plants. In the main diagonal we have the hits (true positives and true negatives) and in the remaining cells the number of misses (false positives and false negatives). Now, with the confusion matrix constructed, the performance of each model can be seen in detail, such is the case of the resNet-50 model in Fig. 9. This process is repeated for the other models (DenseNet-201 and Inception-v3).

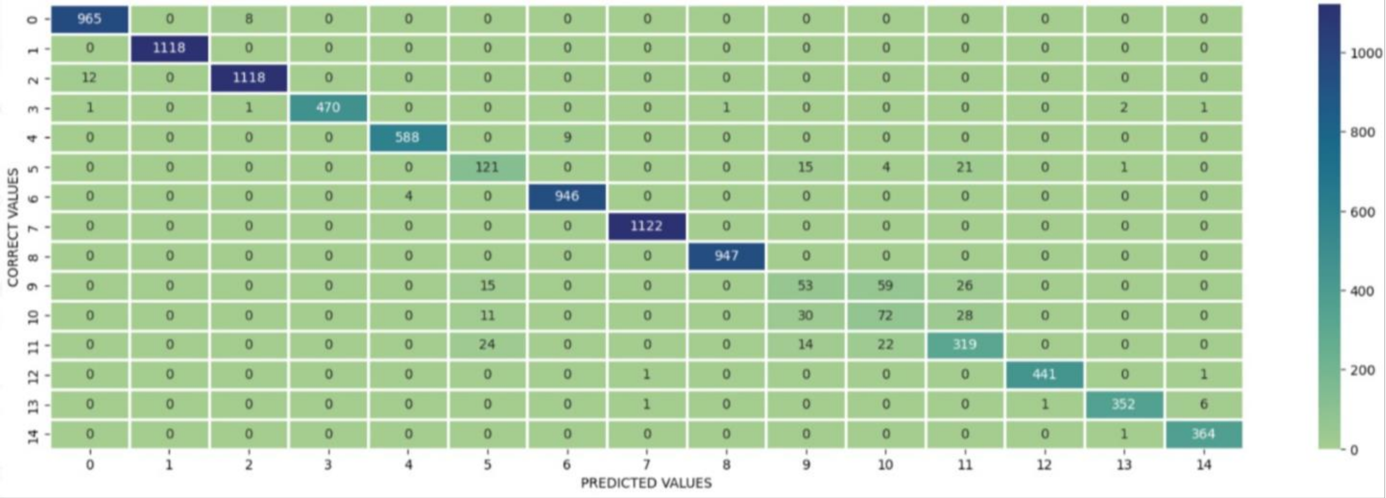


Fig. 9. Confusion matrix of the ResNet-50 model.

IV. Results and Discussion

This section presents the training results of the DenseNet-201, Inception-v3 and ResNet-50 models. For which an original dataset, obtained from the PlantVillage repository, was used, and is composed of approximately 87 thousand images of healthy and diseased crop leaves. The performance of each of the models is subject to the following metrics such as accuracy, precision, recall, F1 score and ROC curve as presented in Table II. The results of the DenseNet-201, Inception-v3 and ResNet-50 models vary depending on the dataset and the specific tasks they perform, as shown in Table II.

Table II shows that the DenseNet-201 and Inception-v3 models achieved better performance in image classification, showing slightly better accuracy than ResNet-50. In terms of accuracy, DenseNet-201, has been characterized as one of the best models for image classification, and it is thanks to its density block design that allows greater propagation of information between layers and better feature extraction. However, the DenseNeet-201 model is usually slower due to its larger number of parameters. On the contrary, the ResNte-50 and Inception-v3 models are faster due to their efficient design. A very important point to take into account is the available computational power and memory limitations. Fig. 10, Fig. 11 and Fig. 12 show the performance of the models. For example, in Fig. 10, graph (Results Loss), shows the performance of the loss and accuracy of the DenseNet-201 model, the blue line represents the training data and the red line the test, as can be seen in the first two epochs the error decreases very significantly, then the error decreases as it goes through the epochs or iterations, and in turn makes the comparison with the test data, however, as it progresses the two lines are separated, this means that it is entering an over fit, ideally the two lines should be joined. With respect to the graph (Results Accuracy) in Fig. 10, which is the result of the Accuracy, it is also evident that from the second epoch there is a very significant increase of hits, and from the third epoch onwards it has been improving its accuracy, however, the red line as the epochs progress is separating from the blue line, which is what should not happen, because it means that the test data are separated from the training. Similarly, it can be seen in Fig. 11, graph (Results Loss), shows the loss and accuracy performance of the ResNet-50 model, the blue line represents the training data and the red line the test, as can be seen in the first two epochs the error decreases very significantly, and progressively decreases the error as it advances the iterations in the epochs, and in turn makes the comparison with the test data, however, as it advances the iterations the two lines are separated, this means that it is entering in an over adjustment, the ideal is that the two lines are joined. With respect to the graph (Results Accuracy) of Fig. 11, it is the result of the Accuracy, it is also evident that from the first epoch there is a very significant increase of hits, and from the second epoch it has been improving its accuracy, however, the red line as it advances the iterations of the epochs is separating from the blue line, it is what should not happen, because it means that the test data is separated from the training. Similarly in Fig. 12, graph (Results Loss), the performance of the loss and accuracy of the Inception-v3 model is presented, where the results are very similar to the Densenet-201 and ResNet-50 models. And with respect to the graph (Results Accuracy) of Fig. 12, is the result of the Accuracy, its performance of this model is very similar to the Densenet-201 and ResNet-50 models, these similarities are seen in Fig.12 graph (Results Accuracy).

TABLE II. MODEL EVALUATION RESULTS

DenseNet-201				
	accuracy [%]	recall [%]	f1-score [%]	support
accuracy			0.98	9316
macro avg	0.93	0.93	0.93	9316
weighted avg	0.98	0.98	0.98	9316

ResNet-50				
	accuracy [%]	recall [%]	f1-score [%]	support
accuracy			0.97	9316
macro avg	0.89	0.89	0.89	9316
weighted avg	0.96	0.97	0.97	9316

Inception-v3				
	accuracy [%]	recall [%]	f1-score [%]	support
accuracy			0.98	9316
macro avg	0.93	0.93	0.93	9316
weighted avg	0.98	0.98	0.98	9316

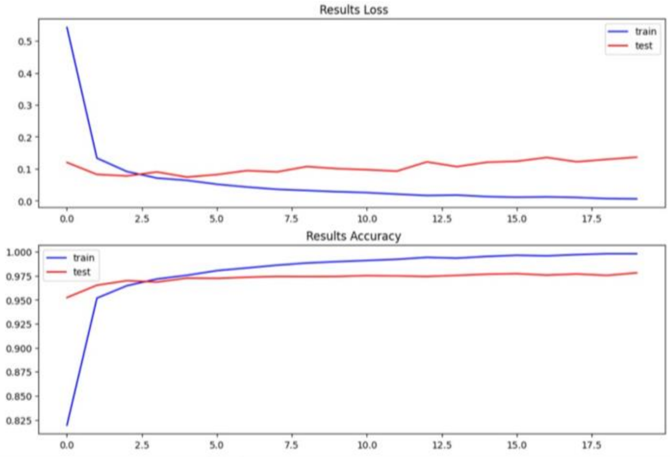


Fig. 10. Comparison of results of the denseNet-201 model between loss and accuracy.

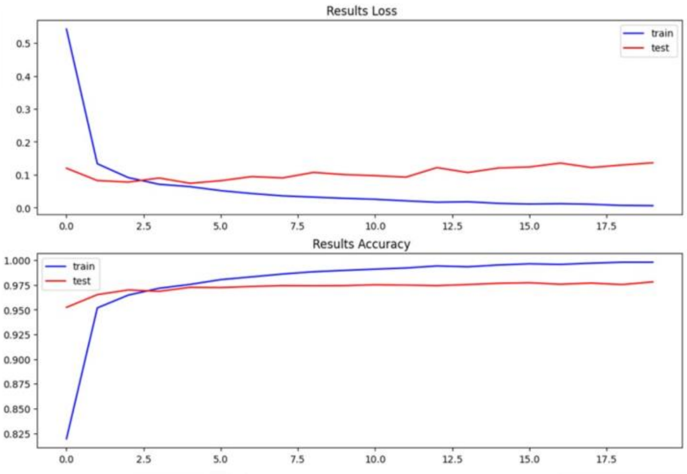


Fig. 11. Comparison of results of the ResNet-50 model between loss and accuracy.

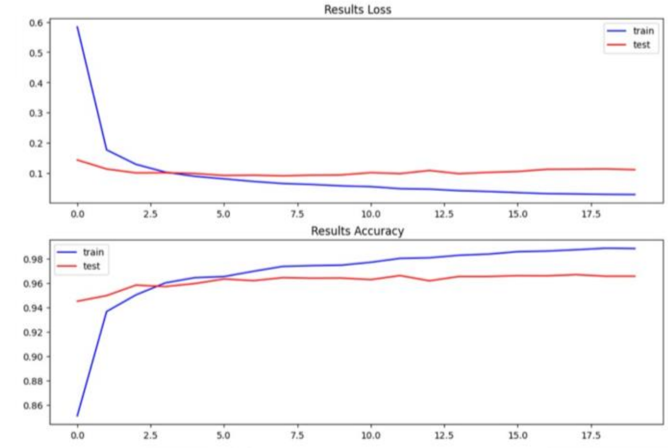


Fig. 12. Comparison of results of the Inception-3 model between loss and accuracy.

The results presented in Table II, Fig. 10, Fig. 11 and Fig. 12, allow an analysis of the performance of each of the trained models (DenseNet-201, ResNet-50 and Inception-v3). In general terms, the performance of the three models has been successful in the identification and classification of diseases in crop plants. For example, the DenseNet-201 model achieved an accuracy of 98%, slightly higher than the performance of the ResNet-50 model. This result is superior to that obtained in [20] where they used this model to classify images, it is important to specify that the accuracy rate will depend on different factors, such as the volume of data with which it was processed, as in [18] where this model achieved an accuracy of 99%, higher than the result obtained in this work. The ResNet-50 model achieved an accuracy level of 97%, slightly lower than the performance of the DenseNet-201 and Inception-v3 models. However, the level obtained does not mean that the model has weaknesses in identifying and classifying images; on the contrary, 97% accuracy is significantly very good. Considering that, for example in [24] this model obtained an accuracy of 92%, it is also considered an optimal rate for its level of accuracy, in [22], [23] used this model for image prediction where it reached an accuracy level of 97.8%, slightly higher than that obtained in this work. This indicates that the performance of the model varies depending on the volume of the data set and the quality of images, among other factors. With respect to the Inception-v3 model, in this work it reached an accuracy level of 98% as shown in Table II, slightly higher than the ResNet-50 model, the Inception-v3 model is characterized by having multiple filters and kernel sizes in each layer, it is a lighter model than DenseNet-201 and ResNet-50, it has fewer parameters, which makes it ideal for identifying and classifying diseases in planes through artificial vision. The level of accuracy obtained is considered optimal, considering that it is close to 100%, this model also obtained a good performance in [16], where it reached an accuracy of 99% in the classification of images of different species, slightly higher than that obtained in this work. Similarly, in [17] they used this model to identify and classify weeds using mixed cropland, where it reached an accuracy level of 92%, a figure much lower than that obtained in this work. However, this does not mean that the model does not perform well in these tasks; on the contrary, the accuracy will depend on the volume of data and the quality of images with which it is processed. Finally, the three models used in this work have achieved optimal performance and accuracy in the process of identification and classification of diseases in crop plants. It is important to point out that the DenseNet-201 and Inception-v3 models obtained the best results in accuracy with 98%, so they would be a viable alternative in technological terms for the identification of diseases in crop plants.

V. Conclusion

The three CNN models (DenseNet-201, ResNet-50 and Inception-v3) used in this work have demonstrated an effective and promising approach, being able to learn relevant features from the images and classify them accurately. A dataset consisting of more than 87 thousand images of healthy and diseased crop leaves, categorized into 38 different categories, was used. For the purpose of identifying and classifying crop plant diseases, using CNN with Transfer Learning. The performance of each of the models was analyzed, as shown in Table II, Fig. 10, Fig. 11 and Fig. 12. The models used are characterized by excellent performance in image identification and classification. For example. The DenseNet-201 and Inception-v3 models achieved an accuracy of 98% in the identification and classification of plant diseases, slightly higher than the ResNet-50 model, but this does not mean that the ResNet-50 model does not have an optimal performance, on the contrary, it achieved 97% accuracy. Likewise, Fig. 11 (Results Loss) shows the training results of the DenseNet-50 model, where it is evident that the blue line from the second iteration drastically reduces the error and progressively decreases as the iterations advance in the epochs. However, at the end it separates from the red line, this is not good for the prediction, because it is an indication of an over adjustment to a greater number of iterations. In the case of Results Accuracy, it is also evident that from the second epoch onwards there is a very significant increase in accuracy, and from the third epoch onwards its accuracy has been improving. The ResNet-50 model in Fig. 12 and the Inception-v3 model in Fig. 12 have a very similar behavior to the DenseNet-201 model.

Finally, the DenseNet-201 and Inception-v3 models achieved the best results in the identification and classification of diseases in crop plants; therefore, they are the two models that are recommended for implementation for this type of task. In the future, a possible complement to this work would include the development of a mobile application so that the implemented model can be consumed.

References

[1] N. Bevers, E. J. Sikora, and N. B. Hardy, “Soybean disease identification using original field images and transfer learning with convolutional neural networks,” Comput Electron Agric, vol. 203, p. 107449, Dec. 2022, doi: 10.1016/J.COMPAG.2022.107449.

[2] B. Dey, M. Masum Ul Haque, R. Khatun, and R. Ahmed, “Comparative performance of four CNN-based deep learning variants in detecting Hispa pest, two fungal diseases, and NPK deficiency symptoms of rice (Oryza sativa),” Comput Electron Agric, vol. 202, p. 107340, Nov. 2022, doi: 10.1016/J.COMPAG.2022.107340.

[3] S. Janarthan, S. Thuseethan, S. Rajasegarar, and J. Yearwood, “P2OP—Plant Pathology on Palms: A deep learning-based mobile solution for in-field plant disease detection,” Comput Electron Agric, vol. 202, p. 107371, Nov. 2022, doi: 10.1016/J.COMPAG.2022.107371.

[4] A. Hussain and P. Balaji Srikaanth, “Disease Classification and Detection Techniques in Rice Plant using Deep Learning,” 8th International Conference on Smart Structures and Systems, ICSSS 2022, 2022, doi: 10.1109/ICSSS54381.2022.9782162.

[5] V. K. Shrivastava and M. K. Pradhan, “Rice plant disease classification using color features: a machine learning paradigm,” Journal of Plant Pathology, vol. 103, no. 1, pp. 17–26, Feb. 2021, doi: 10.1007/S42161-020-00683-3/METRICS.

[6] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks,” Neurocomputing, vol. 267, pp. 378–384, Dec. 2017, doi: 10.1016/J.NEUCOM.2017.06.023.

[7] Rukhsar and S. K. Upadhyay, “Deep Transfer Learning-Based Rice Leaves Disease Diagnosis and Classification model using InceptionV3,” Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2022, pp. 493–499, 2022, doi: 10.1109/CISES54857.2022.9844374.

[8] D. Xiao et al., “Citrus greening disease recognition algorithm based on classification network using TRL-GAN,” Comput Electron Agric, vol. 200, p. 107206, Sep. 2022, doi: 10.1016/J.COMPAG.2022.107206.

[9] H. Yu, J. Liu, C. Chen, A. A. Heidari, Q. Zhang, and H. Chen, “Optimized deep residual network system for diagnosing tomato pests,” Comput Electron Agric, vol. 195, p. 106805, Apr. 2022, doi: 10.1016/J.COMPAG.2022.106805.

[10] S. Kendler et al., “Detection of crop diseases using enhanced variability imagery data and convolutional neural networks,” Comput Electron Agric, vol. 193, p. 106732, Feb. 2022, doi: 10.1016/J.COMPAG.2022.106732.

[11] K. Paul et al., “Viable smart sensors and their application in data driven agriculture,” Comput Electron Agric, vol. 198, p. 107096, Jul. 2022, doi: 10.1016/J.COMPAG.2022.107096.

[12] R. Gajjar, N. Gajjar, V. J. Thakor, N. P. Patel, and S. Ruparelia, “Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform,” Visual Computer, vol. 38, no. 8, pp. 2923–2938, Aug. 2022, doi: 10.1007/S00371-021-02164-9/METRICS.

[13] S. B. Jadhav, V. R. Udupi, and S. B. Patil, “Identification of plant diseases using convolutional neural networks,” International Journal of Information Technology (Singapore), vol. 13, no. 6, pp. 2461–2470, Dec. 2021, doi: 10.1007/S41870-020-00437-5/METRICS.

[14] G. Yogeswararao, V. Naresh, R. Malmathanraj, and P. Palanisamy, “An efficient densely connected convolutional neural network for identification of plant diseases,” Multimed Tools Appl, vol. 81, no. 23, pp. 32791–32816, Sep. 2022, doi: 10.1007/S11042-022-13053-1/METRICS.

[15] J. G. Arnal Barbedo, “Digital image processing techniques for detecting, quantifying and classifying plant diseases,” Springerplus, vol. 2, no. 1, pp. 1–12, Dec. 2013, doi: 10.1186/2193-1801-2-660/TABLES/1.

[16] P. Bedi and P. Gole, “Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network,” Artificial Intelligence in Agriculture, vol. 5, pp. 90–101, Jan. 2021, doi: 10.1016/J.AIIA.2021.05.002.

[17] X. Kang, C. Huang, L. Zhang, M. Yang, Z. Zhang, and X. Lyu, “Assessing the severity of cotton Verticillium wilt disease from in situ canopy images and spectra using convolutional neural networks,” Crop J, Dec. 2022, doi: 10.1016/J.CJ.2022.12.002.

[18] O. G. Ajayi and J. Ashi, “Effect of varying training epochs of a Faster Region-Based Convolutional Neural Network on the Accuracy of an Automatic Weed Classification Scheme,” Smart Agricultural Technology, vol. 3, p. 100128, Feb. 2023, doi: 10.1016/J.ATECH.2022.100128.

[19] A. S. Paymode and V. B. Malode, “Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG,” Artificial Intelligence in Agriculture, vol. 6, pp. 23–33, Jan. 2022, doi: 10.1016/J.AIIA.2021.12.002.

[20] S. Shrimali, “PlantifyAI: A Novel Convolutional Neural Network Based Mobile Application for Efficient Crop Disease Detection and Treatment,” Procedia Comput Sci, vol. 191, pp. 469–474, Jan. 2021, doi: 10.1016/J.PROCS.2021.07.059.

[21] X. Sun, G. Li, P. Qu, X. Xie, X. Pan, and W. Zhang, “Research on plant disease identification based on CNN,” Cognitive Robotics, vol. 2, pp. 155–163, Jan. 2022, doi: 10.1016/J.COGR.2022.07.001.

[22] R. G. Dawod and C. Dobre, “ResNet interpretation methods applied to the classification of foliar diseases in sunflower,” J Agric Food Res, vol. 9, p. 100323, Sep. 2022, doi: 10.1016/J.JAFR.2022.100323.

[23] H. Liu and J. S. Chahl, “Proximal detecting invertebrate pests on crops using a deep residual convolutional neural network trained by virtual images,” Artificial Intelligence in Agriculture, vol. 5, pp. 13–23, Jan. 2021, doi: 10.1016/J.AIIA.2021.01.003.

[24] E. M. Raouhi, M. Lachgar, H. Hrimech, and A. Kartit, “Optimization techniques in deep convolutional neuronal networks applied to olive diseases classification,” Artificial Intelligence in Agriculture, vol. 6, pp. 77–89, Jan. 2022, doi: 10.1016/J.AIIA.2022.06.001.

[25] L. C. Ngugi, M. Abdelwahab, and M. Abo-Zahhad, “A new approach to learning and recognizing leaf diseases from individual lesions using convolutional neural networks,” Information Processing in Agriculture, Oct. 2021, doi: 10.1016/J.INPA.2021.10.004.

[26] M. A. Murti, C. Setianingsih, E. Kusumawardhani, and R. Farhan, “Cedarwood Quality Classification using SVM Classifier and Convolutional Neural Network (CNN),” International Journal of Advanced Computer Science and Applications, vol. 13, no. 11, pp. 101–111, 2022, doi: 10.14569/IJACSA.2022.0131111.

[27] F. B. Mofrad and G. Valizadeh, “DenseNet-based Transfer Learning for LV Shape Classification: Introducing a Novel Information Fusion and Data Augmentation using Statistical Shape/Color Modeling,” Expert Syst Appl, p. 119261, Mar. 2022, doi: 10.1016/J.ESWA.2022.119261.

[28] A.-A. Nayan et al., “A deep learning approach for brain tumor detection using magnetic resonance imaging,” International Journal of Electrical and Computer Engineering (IJECE), vol. 13, no. 1, p. 1039, Feb. 2023, doi: 10.11591/IJECE.V13I1.PP1039-1047.

[29] N. Razmjooy, S. Razmjooy, Z. Vahedi, V. V. Estrela, and G. G. de Oliveira, “Skin Color Segmentation Based on Artificial Neural Network Improved by a Modified Grasshopper Optimization Algorithm,” Lecture Notes in Electrical Engineering, vol. 696, pp. 169–185, 2021, doi: 10.1007/978-3-030-56689-0_9/COVER.

[30] L. Nahhas, M. Albahar, A. Alammari, and A. Jurcut, “Android Malware Detection Using ResNet-50 Stacking,” Computers, Materials & Continua, vol. 74, no. 2, pp. 3997–4014, 2023, doi: 10.32604/CMC.2023.028316.

[31] T. Zheng, Q. Wang, Y. Shen, and X. Lin, “Gradient rectified parameter unit of the fully connected layer in convolutional neural networks,” Knowl Based Syst, vol. 248, Jul. 2022, doi: 10.1016/J.KNOSYS.2022.108797.

[32] F. Martínez, H. Montiel, and F. Martínez, “A Machine Learning Model for the Diagnosis of Coffee Diseases,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 4, pp. 968–974, 2022, doi: 10.14569/IJACSA.2022.01304110.

[33] O. Iparraguirre-Villanueva et al., “Text prediction recurrent neural networks using long short-term memory-dropout,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 29, no. 3, pp. 1758–1768, Mar. 2023, doi: 10.11591/IJEECS.V29.I3.PP1758-1768.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, Accessed: Jan. 17, 2023. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/.

[35] X. Cai, X. Li, N. Razmjooy, and N. Ghadimi, “Breast Cancer Diagnosis by Convolutional Neural Network and Advanced Thermal Exchange Optimization Algorithm,” Comput Math Methods Med, vol. 2021, 2021, doi: 10.1155/2021/5595180.

[36] S. Faizal, C. A. Rajput, R. Tripathi, B. Verma, M. R. Prusty, and S. S. Korade, “Automated cataract disease detection on anterior segment eye images using adaptive thresholding and fine tuned inception-v3 model,” Biomed Signal Process Control, vol. 82, Apr. 2023, doi: 10.1016/J.BSPC.2022.104550.

[37] Z. Guo, L. Xu, Y. Si, and N. Razmjooy, “Novel computer-aided lung cancer detection based on convolutional neural network-based and feature-based classifiers using metaheuristics,” Int J Imaging Syst Technol, vol. 31, no. 4, pp. 1954–1969, Dec. 2021, doi: 10.1002/IMA.22608.

Automatic Detection of Oil Palm Growth Rate Status with YOLOv5

Desta Sandya Prasvita, Dina Chahyati, Aniati Murni Arymurthy
Faculty of Computer Science, University of Indonesia, Depok, Indonesia

***Abstract*—Oil palm plantations are essential for Indonesia as a source of foreign exchange and a provider of employment opportunities. However, large-scale land clearing is considered a cause of deforestation, which harms the environment and society. So, it is necessary to manage plantations that are sustainable and still maintain the preservation of forests and biodiversity. One solution is to apply remote sensing technology. The research was conducted to develop a multi-class detection method for the growth rate of oil palm trees, with five categories: healthy palm, dead palm, yellowish palm, mismanaged palm, and smallish palm. The deep learning-based object detection method, YOLO Version 5 (YOLOv5), is used. This study compares the YOLOv5 network models, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Parameter setting is also carried out in the BCE (Binary Cross Entropy) with Logits Loss Function to handle the problem of unbalanced data distribution in each class. The YOLOv5 model with the highest mAP value is the YOLOv5l and YOLOv5x, the YOLOv5x requires longer training time. In this study, hyperparameter optimization was also carried out using hyperparameter evolution techniques. However, it has yet to provide increased results because the experiments conducted in this study are still limited.**

Keywords*—*Automatic detection; deep learning; oil palm; YOLOv5

I. INTRODUCTION

Palm oil consumption has witnessed a growth of approximately 9% annually in the last decade, and it is anticipated to rise further. This optimistic outlook for the palm oil industry is attributable to its increasing demand in domestic and global markets. Since 2004, the use of palm oil has occupied the highest position, with an average growth of 8% per year. In Indonesia, oil palm serves as a plantation commodity that has an important role in the economy as a source of foreign exchange and a provider of employment. Indonesia is also the world's largest palm oil producer [1]. Oil palm plantations, despite their positive impact on the economy, are often associated with negative impacts that can harm society, such as environmental degradation and conflicts with local communities. [2]. Oil palm plantations are considered a cause of deforestation that can damage biodiversity, with 2% of Indonesia's forests being turned into plantation areas [3]. Based on the background and problems related to oil palm plantations, especially in Indonesia, it is necessary to manage oil palm plantations sustainably so as not to harm the community. That is with an agricultural system oriented towards economic, social, and ecological balance. With proper agricultural practice and technology, it is believed that oil palm plantations will continue growing, with environmental sustainability maintained. Precision agriculture aims to optimize the use of resources to achieve the best results while protecting the environment with land management systems [4].

Remote sensing technology has been widely used in mining, agriculture, and plantations. This technology can be used for remote monitoring of plantation areas. Monitoring plantations manually by human labor is difficult, especially in oil palm plantations in Indonesia characterized by large plantation areas and difficult access. Methods for obtaining optimal results for detecting individual trees have also been developed. Based on previous research, three groups of approaches are used for tree detection. The three methods are classical digital image processing [5] [6] [7] [8], classical machine learning [9] [10], and deep learning. The tree-detection-based deep learning approach is divided into an approach based on CNN classification [11] [12] [13] [14] [15] [16] and another approach based on object detection [17] [18] [19] [20]. The first approach is based on classical digital image processing. There are four main stages to a classical digital image processing-based approach: image pre-processing, treetop detection, tree crown delineation, and post-processing. In the classical machine learning-based tree detection approach, feature extraction is required, which is then trained to build a classification model. The classification model is used to detect trees in images using the sliding window technique. The third approach is based on deep learning for tree detection. Deep learning methods are currently popular for object detection because of their object detection accuracy.

There are advantages and disadvantages to each technique for oil palm tree detection. The classical digital image processing approach has the advantage that it does not require a training process for the construction of classification or detection models that require high and expensive computational costs. Another advantage is the time it takes to detect quickly. However, this classic digital image processing approach has a weakness. It is not easy to use to detect, especially in the case of overlapping trees. On the other hand, the classic machine learning approach has the advantage of seeing overlapping trees. However, accuracy depends on the feature extraction used. Another weakness is that, while the detection process generally uses a sliding window technique, it takes a long time to detect trees. In addition, the sample size of the training image must have been determined when constructing the classification model. In contrast, the objects we detected were of various sizes both the classical machine learning approach and the deep learning approach using the CNN classification-based method share similar advantages and disadvantages. However, the CNN classification-based method does not necessitate feature extraction and has demonstrated superior performance. In contrast, the deep learning method based on object detection can detect objects that overlap with different trees or objects of various sizes and has a much faster detection process. This approach only requires a training process, which is costly and expensive in the computation, especially in cases with many classes.

From the background of the problem, multi-class detection with a deep learning method based on object detection in oil palm trees is state-of-the-art and still has exciting challenges. Zheng et al. [19], in their research, carried out multi-class detection of the growth rate of oil palm trees, with five categories: healthy palm, dead palm, yellowish palm, mismanaged palm, and smallish palm. However, it still has sub-optimal performance and needs improvement. One is the still low F1-score for the dead palm and mismanaged palm classes, namely 43.24% and 44.59%, respectively. The reason for this is that the class contains a relatively small sample of data compared to the quantity of data available in other categories. This study developed a research method based on the work of Zheng et al. to improve the detection results on the growth status of oil palm trees. Zheng et al. have published training datasets, which were used and compared in this research. This research has developed a model that utilizes YOLOv5 for detecting the growth status of oil palm trees, a technique that has already demonstrated success in detecting date palms [17] and tree damage [20]. This study has made several notable contributions, including: 1) utilizing the YOLOv5 method to detect tree growth status, 2) addressing the issue of unbalanced class distribution by adjusting the parameter of BCE with Logits Loss, and 3) conducting hyperparameter optimization experiments for YOLOv5.

II. Literature Review

In the development of deep learning methods based on object detection, such as R-CNN [21], R-CNN [22], faster R-CNN [23], and YOLO [24], the detection process increases in terms of both speed and accuracy. Therefore, an object detection-based deep learning approach is the state-of-the-art approach of this research. There were three primary references used in this study. The first was a research study conducted by Zheng et al. In this research, we developed a model to detect the growth rate of oil palm trees, categorizing them into healthy palms, mismanaged palms, smallish palms, yellowish palms, and dead palms. The data used was derived from unmanned aerial vehicle (UAV) images of oil palm plantation areas in South Kalimantan and Papua, Indonesia. The features in the UAV image data were red, green, and blue (RGB) features. There were five classes: dead palms, healthy palms, mismanaged palms, smallish palms, and yellowish palms. The method for detecting oil palm trees is called Multi-Class Oil Palm Detection (MOPAD). There are three main modules in MOPAD. The first module is the Refined Pyramid Feature (RPF) Module for feature extraction, including four steps: rescaling, integration, refinement, and disintegration. The second is the Multi-Level Region Proposal Network (RPN) with faster R-CNN to generate oil palm candidates. The third is the Hybrid Class-Balanced Loss Module to improve the detection of multi-class palm oil using Class-Balanced Cross-Entropy Loss (CBCEL) and Class-Balanced Smooth L1 Loss (CBSLL). An evaluation was carried out using the evaluation metrics recall, precision, and F1-score. Regarding the achieved performance, this study has produced in detecting oil palm trees at two sites, with F1-scores of 87.91% and 99.04%, precision values of 92.42% and 98.90%, and recall values of 83.82% and 99.19% [19]. However, the detection of multi-class oil palm trees was still low, with an average F1-score of 72.83% and with the most frequently detected classes being dead palms and mismanaged palms. A few challenges persisted in this study, one of which was the low F1-scores in detecting the growth of oil palm trees.

The latest object detection method widely used is YOLO. This method outperforms faster R-CNN in terms of detection speed. YOLOv5 was successfully implemented to detect tree damage due to snowfall automatically. The study utilized image data captured using drone technology from southeastern Norway, featuring RGB channels. The built model was then validated using precision (P), recall (R), $mAP@0.5$, and $mAP@[0.5,0.95]$ accuracy metrics. The validation results for P, R, $mAP@0.5$, and $mAP@[0.5,0.95]$ for all classes were 0.62, 0.61, 0.65, and 0.37. In this research, we employed the YOLOv5 method to detect tree damages caused by snow, which has not been attempted in previous studies. Furthermore, our findings demonstrate that YOLOv5 is capable of addressing the challenge of class imbalance, particularly when the percentage of damaged and dead trees is considerably lower than that of healthy trees. [20].

The YOLOv5 method was also applied to automatically detect date palms using drone imagery [17]. The data for the study were obtained using drones on agricultural land in the United Arab Emirates. The research experiment was carried out by comparing several versions of the YOLOv5 network, including YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra-large). The initial model weights were the pre-training weights from training on the COCO dataset to recognize 80 classes, which were then adjusted for hyperparameters to obtain optimal parameters. The hyperparameter tuning method used was to use a genetic learning algorithm. The test results showed that YOLOv5 had a higher mAP value than SSD300, YOLOv3, and YOLOv4. The detection speeds of YOLOv5s, YOLOv5m, and YOLOv5l were also higher than those of SSD300, YOLOv3, and YOLOv4, but YOLOv5x had the longest average detection time. The YOLOv5 method, recognized for its speed, is an efficient object detection approach that surpasses other CNN-based methods such as R-CNN, fast R-CNN, and faster R-CNN in terms of speed.

This study developed a research method based on the work of Zheng et al. to improve the detection results on the growth status of oil palm trees. Zheng et al. have published training datasets, which were used and compared in this research. This research built a model for detecting the growth status of oil palm trees using YOLOv5, which has been successfully applied in detecting date palms and tree damage.

III. Method

This research went through three main stages: 1) data preparation, 2) development of the YOLOv5 model, and 3) model testing. The data used were the image data provided in a study by Zheng et al. During the data preparation stage, we regenerated the data and separated it into three parts: training data, validation data, and testing data. In the training phase, we concurrently utilized the training and validation data. We employed the training data to construct the model, while the validation data was used to test the model at each iteration and determine the best possible model. In the testing phase, we tested the model created during the training stage using the testing data and performed an evaluation. Fig. 1 illustrates the stages of the research methods.

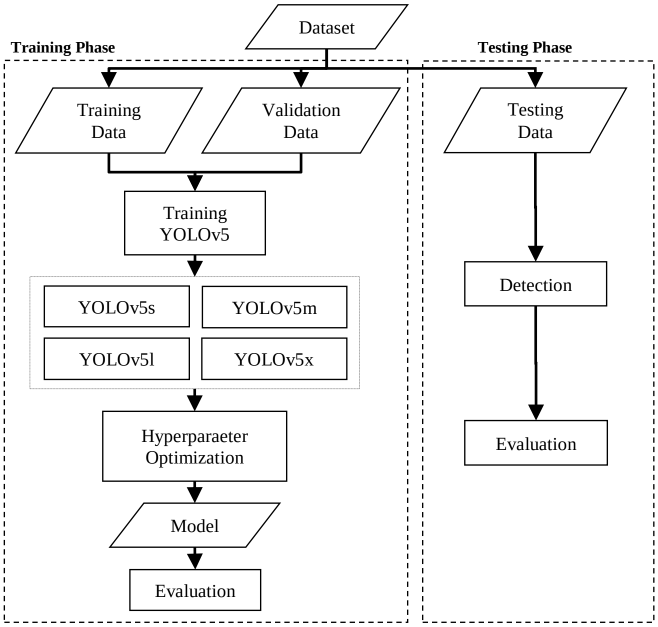


Fig. 1. Research method.

A. Data and Preparation

The dataset used in this study was composed of data from previous studies [19], available at https://github.com/rs-dl/MOPAD. The data acquisition location was Papua, Indonesia (140◦29′ 17′′E, 6◦57′ 42′′S), and the data acquisition was executed using a Skywalker X8 air bridge and a Sony a6000 camera. A spatial resolution of 8 cm with three RGB bands was used. This area had various types of land cover, such as oil palm plantations, rivers, buildings, other vegetation, etc.

For further processing for the development of the YOLOv5 model, the data obtained is carried out at the data preparation stage, namely data regeneration and reformatting. There were 2,303 image data, each measuring 1,024 x 1,024 pixels. Data regeneration was performed using the roboflow online tool. There were three stages in data regeneration: train/split test, pre-processing the data by scaling, and generating YOLOv5 format compliant data. In the train/test split process, the data were divided into training, validation, and testing data, each totaling 1,600 data, 135 data, and 136 data (70%, 15%, and 15%). The rescaling process resized the data from 1,024 x 1,024 pixels to 416 x 416 pixels. Next, necessary formatting adjustments were made to YOLOv5.

TABLE I. THE NUMBER OF IMAGES FOR TRAINING, VALIDATION, AND TESTING DATA

Type	Training Data	Validation Data	Testing Data	Total
Healthy palm	79,988	17,453	17,401	114,842
Dead palm	193	45	57	295
Mismanaged palm	407	68	51	526
Smallish palm	24,486	4,832	4,721	34,039
Yellowish palm	1,339	270	275	1,884
Total	106,413	22,668	22,505	151,586

The dataset included annotations of oil palm trees classified into five labels based on their growth rate. Specifically, the study detected five growth stages of oil palm trees, including dead palms, healthy palms, mismanaged palms, smallish palms, and yellowish palms. Overall, the dataset contained 151,586 oil palm tree annotations for all classes. Table I shows the number of annotations for each category and training, validation, and testing data.

B. The YOLOv5 Network Architecture

The object detection neural network architecture consists of three main components: the backbone extracts image features, the neck incorporates the features extracted from the previous layers, and the head predicts object classes and bounding boxes. The YOLOv5 architecture can be seen in Fig. 2.

YOLOv5 has many key components in each part of its networks, such as focus, Conv (convolution), C3, and Spatial Pyramid Pooling (SPP). The focus module divides the input image into four parallel slices (S) to create a feature map using the convolution module. The convolution module is a basic module that uses convolution operations combined with batch normalization and a leaky-ReLU activation function for feature extraction. The C3 module is designed based on a Cross-Stage Partial (CSP) connection network that is used to improve model learning capabilities. The backbone model uses the SPP module for mixing and unifying spatial features. It down-samples the input features through three parallel max pooling layers and then aggregates them to the featured initial.

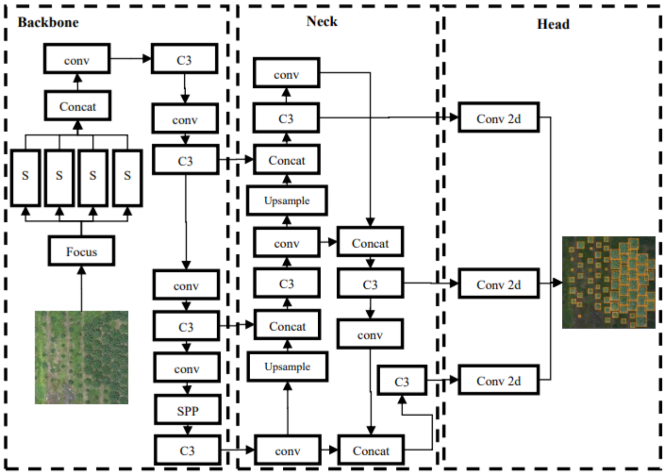


Fig. 2. The network architecture of YOLOv5.

C. Dealing with Imbalanced Data

An imbalance in class distribution can lead to the model better at predicting classes with more data and worse at predicting classes with fewer data. The dataset used in this study has a highly unbalanced distribution. Fig. 3 shows that the healthy oil palm class had a wide data distribution, as marked with a yellow bar. However, there was very little data on yellowish, mismanaged, and dead palm classes depicted in red bars.

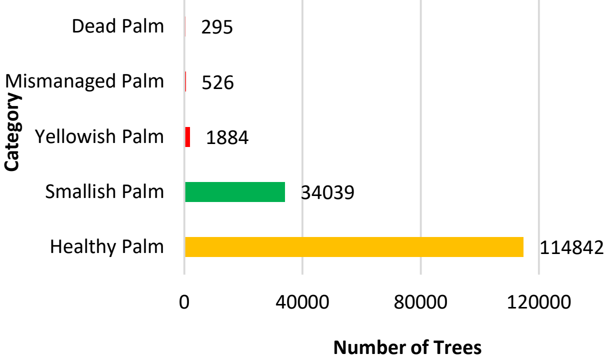


Fig. 3. Data distribution for each class.

The loss function for objectivity in YOLOv5, class probability, was calculated using BCE with Logits Loss Function. The parameters in BCE with Logits Loss were adjusted to overcome the problem of the unbalanced proportion of data in the class. Setting BCE with the Logits Loss parameter made trading of recall and precision possible by adding weight to positive examples. In the case of multi-label classification, the loss ($l_c(x,y)$) can be explained in equation (1).

$$l_c(x,y) = l_c = \{l_{1,c}, \dots, l_{N,c}\}^T,$$
$$l_{n,c} = -w_{n,c} p_c y_{n,c} . log\sigma(x_{n,c}) + (1 - y_{n,c}) . \log 1 - \sigma xn, c \quad (1)$$

Where c is the class number, n is the sample size in the batch, and p_c is the weight of the positive for class c. For example, if the data set contains 5 positive and 200 negative examples of a class, then the pos_weight for that class must equal $\frac{200}{5} = 40$. The loss will act as if the data set contains 40×5=200 positive examples [25].

D. Training the Network

Model training was performed utilizing transfer learning from pre-trained weights of 100 epochs trained on the large Common Objects in Context (COCO) dataset. The pre-trained model was trained to recognize 80 types of objects. The training was carried out for 300 epochs to train the model to recognize objects corresponding to the dataset's five oil palm growth status categories. At each epoch, validation data were executed using the $mAP@[0.5,0.95]$ evaluation metric value. The best model was chosen with the highest $mAP@[0.5,0.95]$ value. The training image data input size was 416 x 416 pixels, with a batch size of 32.

This research built training models for four different versions of YOLOv5, namely, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The difference between the four YOLOv5 versions lies in the number of feature extraction modules in the convolution layer. YOLOv5s has the least feature extraction modules and kernels, and YOLOv5x has the highest feature extraction and kernels. The bigger the model, the better the result, but it has more parameters, requires more CUDA memory, and takes longer to train.

YOLOv5 has around 30 hyperparameters which are used for various training settings. In addition to using hyperparameters in the pre-trained model COCO dataset, this research also optimized the hyperparameters. A hyperparameter tuning process was carried out using hyperparameter evolution [26]. Hyperparameter evolution is a powerful optimization method that uses a genetic algorithm (GA) to optimize hyperparameters in various stages. The first stage is hyperparameter initialization, which utilizes the default parameters of YOLOv5 COCO. The second stage is defining fitness, where the objective function is set to measure how well the model performs. The third stage is evolution, where the GA is utilized to find the optimal set of hyperparameters. Finally, the visualization stage provides a graphical representation of the optimization process. The metric evaluation value of $mAP@[0.5,0.95]$ determines the fitness value. The number of evolutions performed is 300, with each epoch being 10. Visualizations are performed to evaluate the evolution results. Considering the cost of the YOLO training process, the model used for hyperparameter evolution in this study was YOLOv5s only.

E. Evaluation

The model was tested using both validation and testing data. Validation data was used during each epoch of the training process to test the model, while testing data was used to evaluate the final YOLOv5 model. Testing of the object detection model was carried out using the mean average precision (mAP) evaluation metric. The mAP is the average precision (AP) in all detected classes. The AP value is only for each category. The average of 11 interpolation points on the precision-recall curve was calculated to obtain the AP value. According to the 2017 COCO challenge evaluation guidelines, mAP was calculated by the average AP over 80 object classes and all 10 IoU thresholds from 0.5 to 0.95 with a step size of 0.05, i.e., $mAP@[0.5,0.95]$. The calculation of the mAP value can be seen in equations (2) and (3).

$$mAP = \frac{1}{n}\sum_{k=1}^{n} AP_k\,,\ where\ \ n = the\ number\ of\ classes \quad (2)$$

$$AP_k = the\ AP\ of\ class\ k \quad (3)$$

To obtain the AP value, precision and recall values were needed to build a precision-recall curve. Precision is the ratio of the correct prediction for positive data compared to the overall positive predicted result. The precision value can be calculated using the following equation, TP (true positive) is the positive data correctly predicted to the positive class, and FP (false positive) is predicted as an object but false. The precision value can be obtained by using equation (4).

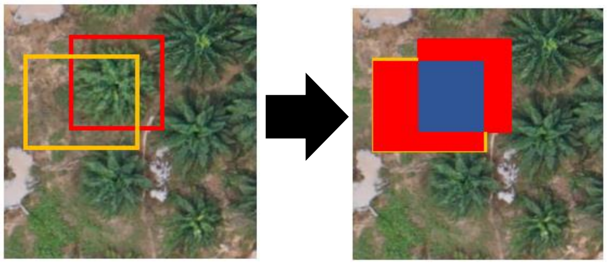
$$Precision = \frac{TP}{TP+FP} \quad (4)$$

A recall is the ratio of true positive predictions to the total number of correct positive data. The recall value can be calculated using the following equation, TP (true positive) is the positive data correctly predicted to the positive class, and FN (false negative) incorrectly predicts the object there. The recall value can be obtained using equation (5).

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

In object detection, TP and FP values are determined using the IoU value. The IoU value is used as the threshold. If the IoU threshold value is 0.5, and the IoU value for a prediction is 0.7, then the prediction result is expressed as TP. Meanwhile, if the IoU threshold is 0.3, it is expressed as FP. Equation (6) is used to calculate the IoU value. An illustration of the calculation of the IoU value can be seen in Fig. 4.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{6}$$



(a) (b)

Fig. 4. Illustration of calculation of IoU value; (a) The ground truth; (b) The red color is the area of overlap, and the blue area is the area of union.

The expected model is a model with high precision and recall values. Therefore, the ideal to measure the model is F1-Score. Mathematically it can be expressed in equation (7).

$$F_1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{7}$$

IV. Results and Discussion

A. BCE with Logits Loss Setting Parameters

One positive class weight vector was chosen as the parameter to manage imbalanced classes in the BCE with Logits Loss function. Positive classes were the class categories in the dataset that you want to detect, namely, the healthy, smallish, yellowish, mismanaged, and dead palm classes. The formula for determining class weight vectors can be seen in equation (8), where c is the detected object category. The number of positive classes is the amount of data in class c, and the number of negative classes is the amount of data in classes other than class c.

$$pos\_weight[c] = \frac{number\ of\ negative\ classes}{number\ of\ positive\ classes} \tag{8}$$

Table II provides the result of setting the BCE with Logits Loss parameter and its calculations.

TABLE II. BCE with Logits Loss Parameter Setting

Class	Count	BCEWithLogitsLoss Setting
Healthy Palm	114842	$\frac{34039+1884+526+295}{114842} = 0.32$
Smallish Palm	34039	$\frac{114842+1884+526+295}{34039} = 3.45$
Yellowish Palm	1884	$\frac{114842+34039+526+295}{1884} = 79.46$
Mismanaged Palm	526	$\frac{114842+34039+1884+295}{526} = 287.19$
Dead Palm	295	$\frac{114842+34039+1884+526}{295} = 512.85$

To set the BCE with the Logits Loss parameter in YOLOv5, it can be found in the file yolov5/utils/loss.py. There were two variables, namely BCEcls, and BCEobj. BCEobj was for objects or backgrounds, and BCEcls was for object classes. The parameter was defined in h['cls_pw'] by changing the value of the vector.

```
# Define criteria
BCEcls = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([h['cls_pw']],
device=device))
BCEobj = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([h['obj_pw']],
device=device))
```

The parameter was defined in h['cls_pw'] by changing the value of the vector as follows.

```
# Define criteria
BCEcls =
nn.BCEWithLogitsLoss(pos_weight=torch.tensor(torch.tensor([[512.,
287., 0.3, 3.5, 79]]), device=device))
BCEobj = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([h['obj_pw']],
device=device))
```

B. Results of the Training Process

The training was conducted with two experiments to obtain the optimal model for detecting oil palm trees. The initial experiment employed a pre-trained model, whereas the subsequent one involved leveraging the evolution method to optimize hyperparameters. Notably, the training procedure was executed utilizing Google Colab Pro+.

- Training with Pre-Trained Default Models

The experiment in the training process was to compare all YOLOv5 networks, including YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The training used 300 epochs with a batch size of 16. Table III compares evaluation metrics and training time for all YOLOv5 models.

YOLOv5l produced the highest mAP value among all YOLOv5 models, with $mAP@[0.5{,}0.95]$ of 0.90. Although the YOLOv5x model had a deeper network, it did not improve the results and even had the lowest average F1-score of 0.95. YOLOv5s, YOLOv5m, and YOLOv5l had a training time of 3 hours, and YOLOv5x had a training time of five hours with 300 epochs.

TABLE III. COMPARISON OF EVALUATION METRICS AND TRAINING TIME OF THE YOLOV5 MODEL

Model YOLOv5	P	R	mAP @$[0.5, 0.95]$	Average F1-score	Time
YOLOv5s	0.95	0.96	0.86	0.96	3.005 hours
YOLOv5m	0.95	0.97	0.89	0.96	3.185 hours
YOLOv5l	0.96	0.98	0.90	0.97	3.652 hours
YOLOv5x	0.95	0.95	0.90	0.95	5.550 hours

- Training with the Model by Tuning the Hyperparameter Evolution

Hyperparameter optimization experiments were carried out using the YOLOv5 model with the smallest network, namely, YOLOv5s. The number of evolutions used was 300, each using ten epochs. Fig. 5 is a visualization of the hyperparameter tuning process evolve. The yellow color indicates higher concentrations. The vertical distribution suggests that the parameter had been deactivated and not mutated.

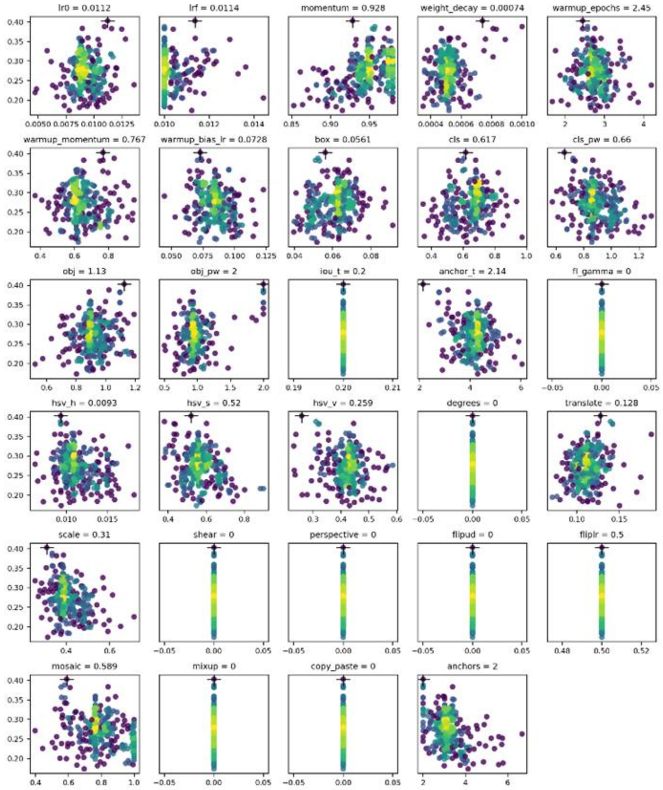


Fig. 5. Visualization of hyperparameter optimization results.

The evolved model did not improve yields compared to the default models. Fig. 6 compares the $mAP@[0.5,0.95]$ and F1-scores for detecting oil palm trees. The process of hyperparameters optimization with evolution required high costs, and evolutions required GPU processing that took days. In other words, some experimental limitations still existed at this stage, with a limited number of evolutions, namely, 300, and 10 epochs for each evolution.

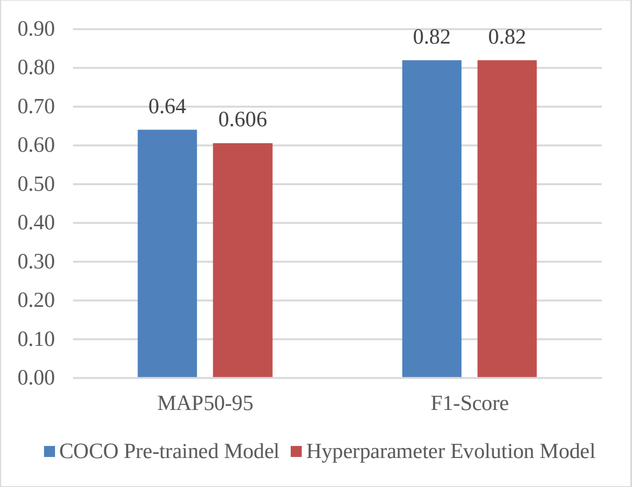


Fig. 6. Model comparison with hyperparameter evolution model and pre-trained models.

C. Model Testing Process

We compared the performance of YOLOv5 on small, medium, large, and extra-large networks using testing data. To evaluate the results, we used the F1-score and $mAP@[0.5,0.95]$ metrics on the testing data. The difference in F1 score, precisely in the mismanaged palm class, was quite far, with the lowest F1-score obtained using YOLOv5s. The F1-score for the mismanaged palm class increased significantly using YOLOv5l, in which case the increase amounted to 11.3%. Results showed that the detection of the mismanaged oil palm class had the lowest F1-score. The highest F1-score was found in the dead palm class using YOLOv5l. The F1-score comparison for each YOLO-v5 model can be seen in Fig. 7.

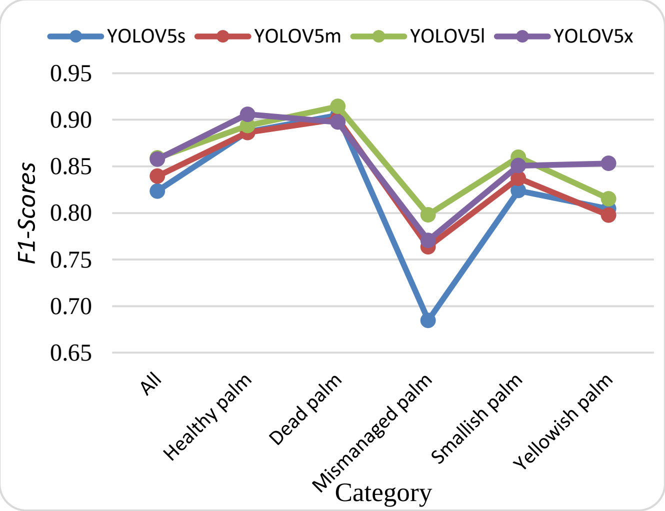


Fig. 7. Comparison of F1-Scores across YOLOv5 models.

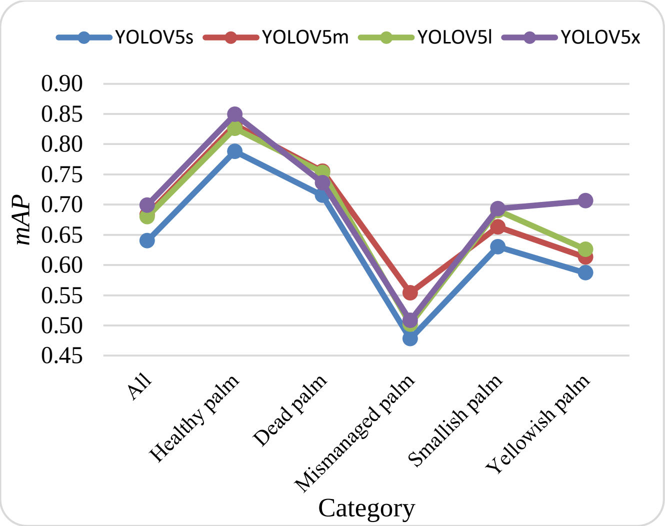


Fig. 8. Comparison of $mAP@[0.5,0.95]$ values across YOLOv5 models.

The comparison of $mAP@[0.5,0.95]$ for each YOLOv5 model can be seen in Fig. 8. The evaluation of the detection of oil palm growth rate showed that the model is suitable for detecting healthy oil palm trees. The highest $mAP@[0.5,0.95]$ value for the healthy palm oil class was 0.849, in which YOLOv5x was used. The class that was difficult to detect using YOLOv5 was the mismanaged oil palm tree class. The lowest $mAP@[0.5,0.95]$ value was 0.478, obtained with YOLOv5s, and the highest was 0.554, obtained with YOLOv5m. Detection of smallish and yellowish palms using YOLOv5 models was still difficult, but the $mAP@[0.5,0.95]$ values for these classes were better than that of the mismanaged class. For the dead palm class, the $mAP@[0.5,0.95]$ value was also quite good, which was above 0.7.

Based on the evaluation of $mAP@[0.5,0.95]$, the YOLOv5s model had the lowest scores for all classes. YOLOv5m, YOLOv5l, and YOLOv5x were in the range of $mAP@[0.5,0.95]$ evaluation values, which were not much different. YOLOv5x was sufficient to increase the value of $mAP@[0.5,0.95]$ by 0.04 compared to other YOLOv5 models. If we look at the results of the $mAP@[0.5,0.95]$ evaluation, using the YOLOv5 model with medium and large networks generated good detection results. Using the YOLOv5 extra-large network, there was not much increase in the detection results. Only the yellowish palm class saw a significant increase in the $mAP@[0.5,0.95]$ value by around 0.4.

D. Evaluation of Detection Results

The $mAP@[0.5,0.95]$ evaluation metric shows that the model was very good at detecting healthy palms and quite good at detecting dead palms. It can be seen in Fig. 9, an example of a randomly selected detected image. The detection results circled in red in Fig. 9(b), 9(c), and 9(d) are the results of mismanaged class errors (FP for mismanaged classes). The objects should have been detected as healthy oil palm trees but came out as ambiguous detection results. For instance, the objects were detected more than twice as mismanaged and healthy palms. The detection results for the mismanaged class also showed ambiguity in that one object was detected more than once, as shown in Fig. 9(a). A tree of the mismanaged palm class was detected more than once. As seen in the image with a yellow circle, the oil palm tree objects overlapped. False positive (FP) detection errors in the yellowish palm class can also be seen in Fig. 9. The yellowish class detection results are marked with a green box and a blue circle. Some objects should have been detected as members of the healthy palm class, but they were detected as part of the yellowish palm class instead. From the analysis of detection results, it was acknowledged that the model still had FP errors in detecting mismanaged and yellowish palms. From the characteristics of the image, it can be analyzed that FP errors in both the mismanaged and yellowish palm classes occurred in tree objects that overlapped each other.

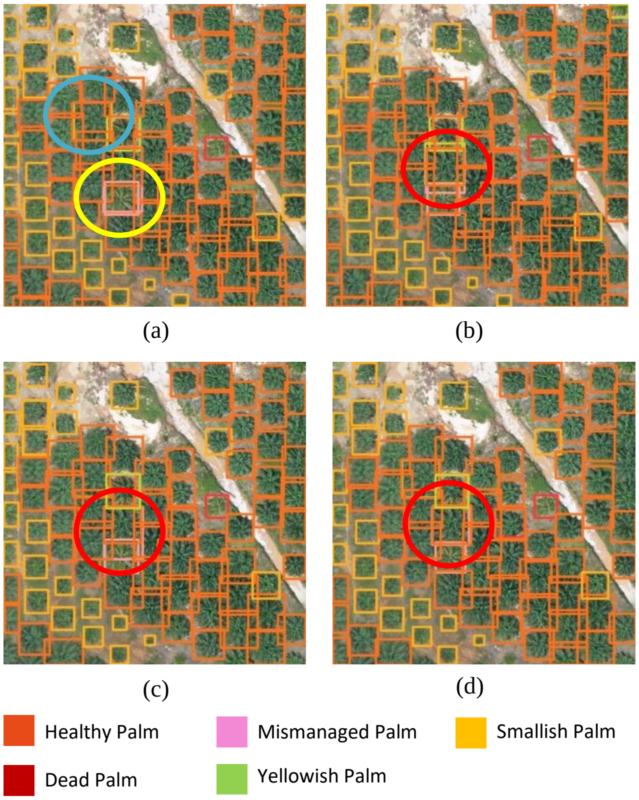
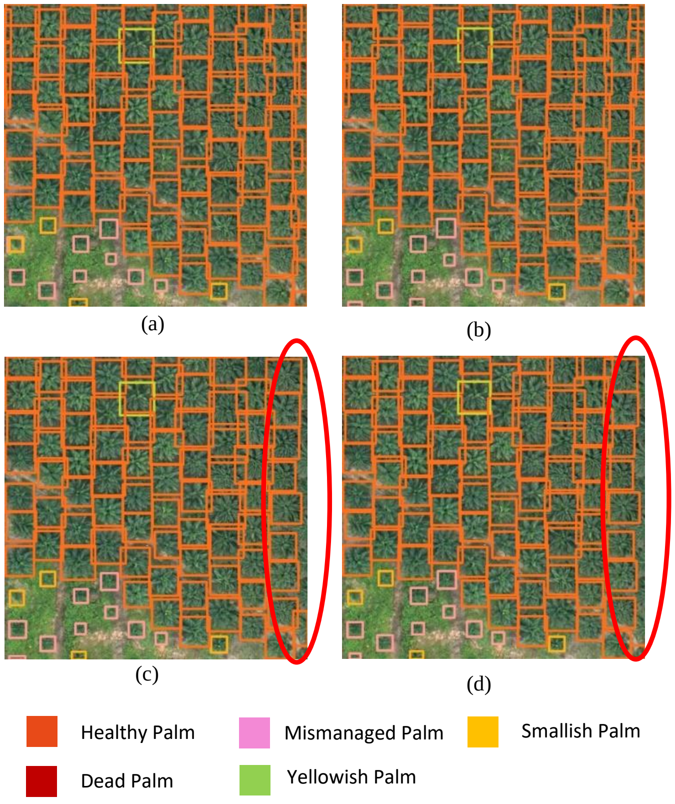


Fig. 9. Detection results for the first image sample: (a) YOLOv5s Model; (b) YOLOv5m Model; (c) YOLOv5l Model; (d) YOLOv5x Model.

Fig. 10 is an example of another detection result; there were trees in the mismanaged palm category that did not overlap, which are marked with purple bounding boxes. In contrast to the previous image, this image provides precise detection results, where oil palm trees of the mismanaged palm class were successfully detected without any false positive (FP) detection. Fig. 10 also compares YOLOv5 in each network (small, medium, large, and extra-large). In the area marked with blue circles, YOLOv5s, and YOLOv5m showed FP in the healthy palm class, where the background was detected as healthy palms. The YOLOv5l and YOLOv5x models could improve the detection results, as the red circles show.



(a) (b) (c) (d)

Fig. 10. Detection results for the second image sample: (a) YOLOv5s model; (b) YOLOv5m model; (c) YOLOv5l model; (d) YOLOv5x model.

E. Comparison of the Proposed Method with Previous Research

A comparison between the method proposed in previous studies and YOLOv5 was made. In the research conducted previously by [19], the evaluation of detection results was carried out using the F1-score. Zheng et al. named the detection method MOPAD, with an average F1-score of 72%. In addition, they also conducted experiments with other methods, such as CNN and Faster R-CNN, with an average of 28% and 56% for the two methods. As seen in Table IV, this study has improved F1-score results even with the YOLOv5s network, with an average F1-score of 72%. The highest F1-score was obtained using YOLOv5x, with an average F1-score of 86%.

The highest average F1-Score is the YOLOv5x model, but the difference in the average F1-Score is only 1% compared to the YOLOv5l. The training time for the YOLOv5x model is longer than the YOLOv5l, with a difference in training time of about two hours. It can be seen in Table IV. The analysis results show that the best oil palm growth rate detection model is YOLOv5l. Besides having a faster training time, YOLOv5l outperforms its competitors in detecting five different classes related to the growth rate of oil palm trees. Specifically, three categories - dead palm, mismanaged palm, and smallish palm - achieve the highest F1-score with YOLOv5l. The evaluation of each class was as follows: healthy palms on YOLOv5x, dead palms on YOLOv5l, mismanaged palms on YOLOv5l, smallish palms on YOLOv5l, and yellowish palms on MOPAD.

TABLE IV. COMPARISON OF THE PROPOSED METHOD WITH PREVIOUS RESEARCH

Method	Healthy	Dead	Mis-managed	Smallish	Yellowish	Average F1-score
CNN (ResNet-101) [19]	0.75	0.07	0.03	0.36	0.19	0.28
Faster R-CNN [19]	0.90	0.06	0.42	0.66	0.74	0.56
MOPAD [19]	**0.91**	0.55	0.51	0.77	**0.88**	0.72
YOLOv5s	0.89	0.90	0.68	0.82	0.80	0.82
YOLOv5m	0.89	0.90	0.76	0.84	0.79	0.84
YOLOv5l	0.89	**0.91**	**0.80**	**0.86**	0.81	0.85
YOLOv5x	**0.91**	0.90	0.77	0.85	0.85	**0.86**

V. CONCLUSION

This study found several contributions, including: 1) The application of the YOLOv5 method for detecting the growth level of oil palm trees. 2) A comparison of the YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x models. 3) Handling unbalanced class distributions by setting positive class vector parameters in the BCE with the Logits Loss function. 4) Conducting experiments to optimize hyperparameters with evolutions.

Based on the experiments conducted in this study, the YOLOv5 model improved the results of detecting the growth level of oil palm trees. The highest F1-score was obtained with the YOLOv5x model, but it took the longest learning time. The YOLOv5l model is the best because the training time is shorter, and the difference in F1 scores is slightly smaller than the YOLOv5x model by 1%. In addition to outperforming other YOLOv5 models' F1-score, the evaluation results for each class show YOLOv5l achieved the highest F1-score for 3 out of 5 classes. The detection results showed that the model still had errors in predicting trees of the mismanaged and yellowish palm classes, and the FP value for overlapping trees was still relatively high.

Setting positive class vector parameters in the BCE with the Logits Loss function could solve the unbalanced class distribution problem; where there was an imbalance in the class distribution, there the imbalance was extreme. Still, adjusting BCE with Logits Loss function parameters could produce an evaluation value for each class that was still quite good. Hyperparameter optimization by evolutions required high costs, where the training process could take days to complete. This study was limited to hyperparameter optimization experiments on the YOLOv5s model with 300 evolutions and ten epochs for each evolution. The hyperparameter optimization with evolutions results did not show a significant increase in the model.

This study applied a deep learning method based on object detection, a state-of-the-art method in oil palm tree detection research. This research still faced challenges, especially in detecting multi-class oil palm trees. It is insufficient to only feature RGB images for detecting tree growth and health level, providing only color and texture information. To better detect the growth level of oil palm trees, it is advisable to add remote sensing sensors.

ACKNOWLEDGMENT

This research is partially supported by an internal publication grant from the Faculty of Computer Science, University of Indonesia.

REFERENCES

[1] Kementrian Perindustrian RI, Tantangan dan Prospek Hilirisasi Sawit Nasional Analisis Pembangunan Industri. 2021.

[2] Ngadi and M. Noveria, “Keberlanjutan Perkebunan Kelapa Sawit di Indonesia dan Prospek Pengembangan Perbatasan,” J. Masy. Indones., vol. 43, no. 1, pp. 95–111, 2017.

[3] E. Meijaard et al., Kelapa sawit dan Keanekaragaman Hayati Analisis situasi oleh Satuan Tugas Kelapa Sawit IUCN. 2018.

[4] E. N. Ginting and D. Wiratmoko, “Potensi dan Tantangan Penerapan Precision Farming dalam Upaya Membangun Perkebunan Kelapa Sawit yang Berkelanjutan,” War. PPKS, vol. 26, no. 2, pp. 55–65, 2021.

[5] H. Z. M. Shafri, N. Hamdan, and M. I. Saripan, “Semi-automatic detection and counting of oil palm trees from high spatial resolution airborne imagery,” Int. J. Remote Sens., vol. 32, no. 8, pp. 2095–2115, 2011, doi: 10.1080/01431161003662928.

[6] J. Yang, Z. Kang, S. Cheng, Z. Yang, and P. H. Akwensi, “An Individual Tree Segmentation Method Based on Watershed Algorithm and Three-Dimensional Spatial Distribution Analysis from Airborne LiDAR Point Clouds,” IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 13, pp. 1055–1067, 2020, doi: 10.1109/JSTARS.2020.2979369.

[7] T. Yun et al., “Individual tree crown segmentation from airborne LiDAR data using a novel Gaussian filter and energy function minimization-based approach,” Remote Sens. Environ., vol. 256, no. December 2020, p. 112307, 2021, doi: 10.1016/j.rse.2021.112307.

[8] A. Harikumar, P. D’Odorico, and I. Ensminger, “A Fuzzy Approach to Individual Tree Crown Delineation in Uav Based Photogrammetric Multispectral Data,” in International Geoscience and Remote Sensing Symposium, 2020, pp. 4132–4135.

[9] Y. Wang, X. Zhu, and B. Wu, “Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier,” Int. J. Remote Sens., vol. 40, no. 19, pp. 7356–7370, 2019, doi: 10.1080/01431161.2018.1513669.

[10] A. P. D. Corte et al., “Forest inventory with high-density UAV-Lidar: Machine learning approaches for predicting individual tree attributes,” Comput. Electron. Agric., vol. 179, no. April, p. 105815, 2020, doi: 10.1016/j.compag.2020.105815.

[11] D. S. Prasvita, M. M. Santoni, R. Wirawan, and N. Trihastuti, “Klasifikasi Pohon Kelapa Sawit Pada Data Fusi Citra Lidar Dan Foto Udara Menggunakan Convolutional Neural Network,” JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 6, no. 2, pp. 406–415, 2021, doi: 10.29100/jipi.v6i2.2437.

[12] W. Li, H. Fu, L. Yu, and A. Cracknell, “Deep learning based oil palm tree detection and counting for high-resolution remote sensing images,” Remote Sens., vol. 9, no. 1, 2017, doi: 10.3390/rs9010022.

[13] W. Li, H. Fu, and L. Yu, “Deep convolutional neural network based large-scale oil palm tree detection for high-resolution remote sensing images,” in International Geoscience and Remote Sensing Symposium (IGARSS), 2017, vol. 2017-July, pp. 846–849, doi: 10.1109/IGARSS.2017.8127085.

[14] N. A. Mubin, E. Nadarajoo, H. Z. M. Shafri, and A. Hamedianfar, “Young and mature oil palm tree detection and counting using convolutional neural network deep learning method,” Int. J. Remote Sens., vol. 40, no. 19, pp. 7500–7515, 2019, doi: 10.1080/01431161.2019.1569282.

[15] W. Li, R. Dong, H. Fu, and L. Yu, “Large-Scale Oil Palm Tree Detection from High-Resolution Satellite Images Using Two-Stage Convolutional Neural Networks,” Remote Sens., vol. 11, no. 1, 2019, doi: 10.3390/rs11010011.

[16] S. R. Aliandra and D. S. Prasvita, “Application of Median Filter Method for Classification of Oil Palm Tree on LiDAR Images,” pp. 441–444, 2022.

[17] T. Jintasuttisak, E. Edirisinghe, and A. Elbattay, “Deep neural network based date palm tree detection in drone imagery,” Comput. Electron. Agric., vol. 192, no. November 2021, p. 106560, 2022, doi: 10.1016/j.compag.2021.106560.

[18] Z. Hao et al., “Automated tree-crown and height detection in a young forest plantation using mask region-based convolutional neural network (Mask R-CNN),” ISPRS J. Photogramm. Remote Sens., vol. 178, no. May, pp. 112–123, 2021, doi: 10.1016/j.isprsjprs.2021.06.003.

[19] J. Zheng et al., “Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images,” ISPRS J. Photogramm. Remote Sens., vol. 173, no. August 2020, pp. 95–121, 2021, doi: 10.1016/j.isprsjprs.2021.01.008.

[20] S. Puliti and R. Astrup, “Automatic detection of snow breakage at single tree level using YOLOv5 applied to UAV imagery,” Int. J. Appl. Earth Obs. Geoinf., vol. 112, no. August, p. 102946, 2022, doi: 10.1016/j.jag.2022.102946.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.

[22] R. Girshick, “Fast R-CNN,” Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[23] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[25] The PyTorch Foundation, “BCEWITHLOGITSLOSS,” 2022. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html.

[26] G. Jocher, “Hyperparameter Evolution,” https://docs.ultralytics.com/, 2022. https://docs.ultralytics.com/tutorials/hyperparameter-evolution/.

Hybrid Approach Used to Analyze the Sentiments of Romanized Text (Sindhi)

Irum Naz Sodhar[1], Suriani Sulaiman[2], Abdul Hafeez Buller[3], Anam Naz Sodhar[4]
Post-Doctoral Fellow, Department of Computer Science-Kulliyyah (Faculty) of Information and Communication Technology, International Islamic University, Malaysia[1]
Assistant Professor, Department of Computer Science-Kulliyyah (Faculty) of Information and Communication Technology, International Islamic University, Malaysia[2]
Post-Doctoral Fellow, Department of Civil Engineering-Kulliyyah (Faculty) of Engineering, International Islamic University, Malaysia[3]
Postgraduate Student, Quaid-e-awam University of Engineering, Science & Technology, Nawabshah, Sindh, Pakistan[4]

***Abstract*—Sentiment analysis is an important part of natural language processing (NLP). This study evaluated the sentiment of Romanized Sindhi Text (RST) using a hybrid approach and ground truth values. The methodology of sentiment analysis involves three major steps: input data, process on tool, analysis of data and evaluation of results. One hundred RST sentences were used in this study's sentiment analysis, which can be positive, neutral, or negative. The statements in the corpus of this study are simple to understand and are used in everyday life. This research used an online Python tool to process a text and get results in the form of outcomes. The results showed that 86% of the sentences have neutral sentiments, 9% of the total results of sentiment analysis have negative sentiments, and only 5% of sentences of Romanized Sindhi Text have positive sentiments. The accuracy of the RST was measured on an online calculator and the value was 87.02% on the basis of ground truth values. An error ratio of 12.98% was calculated on the basis accuracy found on the online calculator of confusion matrix.**

***Keywords*—Sentiment analysis; natural language processing; hybrid approach; python tool; Romanized Sindhi**

I. INTRODUCTION

Sentiment analysis is the most important task of NLP, in which it analyses the community's opinions about social actions such as social media apps, academic activities, and technology [1, 2]. Sentiment analysis is the analysis of opinions about users [3, 4]. The principle part of artificial intelligence (AI) and man-made brainpower in NLP is to measure the content and investigate the importance of the content [5]. The information or text utilized for the Natural Language Processing looks like unstructured and organized information or text [6]. Text investigation is the cycle of changing unstructured content information over to organized content information as significant. Text examination utilized apparatuses for a few contents and measured factual information by utilizing artificial intelligence calculations [7]. Text investigation is additionally used to assess the client's assessment, and item audits with criticism are used to give a better reaction to future assignments. Text is utilized to recognize examples, and the fundamental thought of the examination comes from various wellsprings of data. Sentiment analysis is mostly used for the analysis of comments about any product or any other social activity [8, 9].

This research focused on the Romanization of Sindhi language by using hybrid model for the evaluation/analysis of sentiments. After the analysis of results on tool, author evaluates the results on the basis of ground truth / reality basis. The significance of this research study is to evaluate the results of tool and actual aim of the sentence in selected language.

Sentiment Analysis applied in this study project on 100 sentences of RST. Sentiment Analysis is done on the online Python tool and it may the result in Positive, Neutral and Negative Sentiments. After the task performed on tool, all sentiments were compared on ground truth values and accuracy was measured as 87.02%.

II. STRUCTURE OF ROMANIZED SINDHI TEXT

Structure of sentiment analysis of RST is same as English language [10]. The structure of Romanized Sindhi Text depend upon the three main attribute of grammar as like: subject à Verb à Object, same as like an English sentence [11]. The structure of Romanized Sindhi Text is easy to understand by the tool and it recognized the sentence by using above three attributes and it may give better output [12].

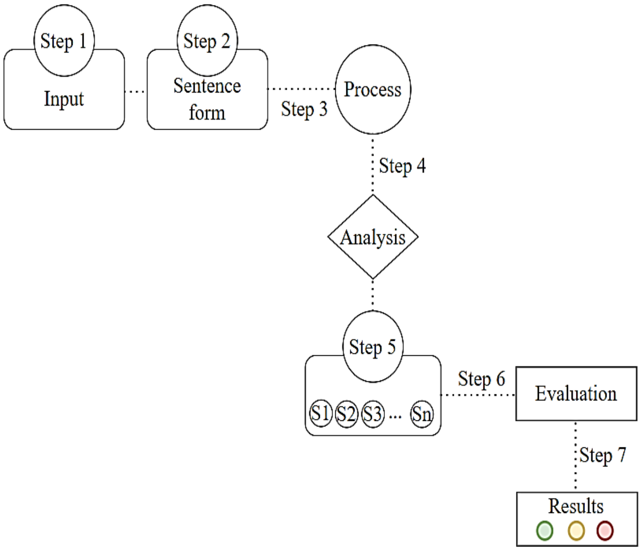
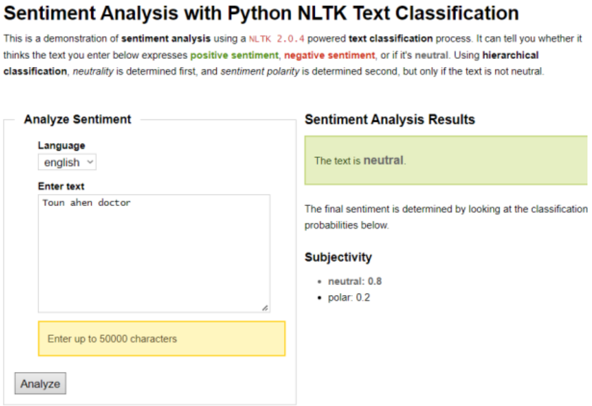


Fig. 1. Research methodology of sentiment analysis.

Fig. 1 show the methodology of sentiment analysis of RST depends on three major steps such as: Input data, Process on tool, analysis of data and evaluation of results. According to the first step of the methodology, input data is collected. The data in the RST is the input data in the form of sentences. After this step, the data is processed on a tool in the shape of sentences, but the sentences may be single or multiple. After the second step of processing, tool is to analyze the data and it may the output results in the shape of sentiments. These sentiments are positive, neutral and negative.

III. Analysis of Romanized Sindhi Text Analysis

One Hundred RST sentences were used in this study's sentiment analysis. A text's sentiment can be positive, neutral, or negative [13-15]. The statements in the corpus of this study are simple to understand. These sentences are used in everyday life. A data set is a fundamental element of research and is used as input towards the online Python tool (as shown in Fig. 2), which then processes a text and gets results in the form of outcomes. Sentiment analysis results, on the basis of ground truth value Data set of this research, is shown in below Table I.



Sentiment Analysis with Python NLTK Text Classification

This is a demonstration of **sentiment analysis** using a NLTK 2.0.4 powered **text classification** process. It can tell you whether it thinks the text you enter below expresses **positive sentiment**, **negative sentiment**, or if it's neutral. Using **hierarchical classification**, *neutrality* is determined first, and *sentiment polarity* is determined second, but only if the text is not neutral.

Analyze Sentiment

Language: english

Enter text:

```
Toun ahen doctor
```

Enter up to 50000 characters

Analyze

Sentiment Analysis Results

The text is **neutral**.

The final sentiment is determined by looking at the classification probabilities below.

Subjectivity

- neutral: 0.8
- polar: 0.2

Fig. 2. Input/output view of online Python tool.

TABLE I. Sentiment Analysis of Romanized Sindhi Sentence

#	Sindhi Sentence	Romanized Sindhi Sentence	Sentiment Analysis	Ground through value
1	تون آهين ڊاڪٽر	Toun ahen doctor	The text is neutral.	Neutral
2	مان آهيان شاگرد	Maa'n ahiyan shagrid	The text is neutral.	Neutral
3	مان کيڏان راند	Maa'n khedan rand	The text is neutral.	Neutral
4	مان پيان پاڻي	Maa'n piyan pani	The text is neutral.	Neutral
5	علي ماريو نانگ	Ali maryo nang	The text is neutral.	Neutral
6	ڊاڪٽر سڏيو مريض کي	Doctor sadiyo mariz khe	The text is neutral.	Neutral
7	هي آهي بدصورت	He ahy bad surat	The text is neg.	Negative
8	مان لکيو خط	Maa'n likhyo khat	The text is neutral.	Neutral
9	مان پڙهيو ڪتاب	Maa'n parhyo kitab	The text is neutral.	Neutral
10	تو ساڙيو گهر	Toun sadyo ghar	The text is neutral.	Negative
11	مان ڪيو ڪم	Maa'n kayo kam	The text is neutral.	Neutral
12	تو ٻڌو گانو	Toun budho gano	The text is neutral.	Neutral
13	حنا ٿوڙيو ڪلاس	Hina tourdyo glass	The text is neutral.	Negative
14	مان کوليو دروازو	Maa'n kholiyo darwazo	The text is neutral.	Neutral
15	مان سکي انگلش	Maa'n sikhi English	The text is neutral.	Neutral
16	تو ٿاهي چانهن	Toun thai chanhe	The text is neutral.	Neutral
17	تون آهين خوبصورت	Toun ahen khoubsorat	The text is neutral.	Neutral
18	تون آهين ڪنو	Toun ahen kino	The text is neutral.	Negative
19	آهين تون ٺيڪ	Ahen toun thek	The text is neutral.	Neutral
20	ڪتو ڀونڪي ٿو	Kuto bhonke tho	The text is neutral.	Neutral
21	مان لکيو آرٽيڪل	Maa'n likhyo article	The text is neutral.	Neutral
22	هتي ارام ويهه	Irum waihu hite	The text is neutral.	Neutral
23	مون کي آهي ڏک	Moun khe ahe dukh	The text is neutral.	Neutral
24	مان آهيان خوش	Maa'n ahiyan khush	The text is neutral.	Neutral
25	هتي سارا اُٺي بيهه	Sara uthee beehu hite	The text is neutral.	Neutral
26	تون آهين استاد	Toun ahen ustad	The text is neutral.	Neutral
27	مان سکان ٿي سنڌي	Maa'n sikhan thi Sindhi	The text is neutral.	Neutral
28	مان سکان ٿي ميٿ	Maa'n sikhan thi math	The text is neutral.	Neutral
29	علي ماريو واڳون	Ali maryo wagoon	The text is neutral.	Negative
30	ڪرسي علي ويهه تي	Ali waihu kursi te	The text is neutral.	Neutral
31	انعم اچ	Anam achu	The text is neutral.	Neutral
32	ڏاڍو خراب آهي بيڊ	Dadho kharab ahy bad	The text is neg.	Negative
33	گندگي آهي بيڊ تي	Gandagi ahy bad te	The text is neg.	Negative
34	مان آهيان استاد	Maa'n ahiyan ustad	The text is neutral.	Neutral
35	چا آهين تون ڊاڪٽر	Cha ahin toun doctor	The text is neutral.	Neutral
36	مان هلائي مشين	Maa'n halai machine	The text is neutral.	Neutral
37	انعم اچ ويهه هتي	Anam ach hite waihu	The text is neutral.	Neutral

#	Sindhi Sentence	Romanized Sindhi Sentence	Sentiment Analysis	Ground through value
38	مان وڃان ٿو اسڪول	Maa'n wanjan tho school	The text is neutral.	Neutral
39	آهين تونشاگرد	Ahin toun shagrid	The text is neutral.	Neutral
40	آهيان مان ڊاڪٽر	Ahyan Maa'n doctor	The text is neutral.	Neutral
41	هوءَ آهي استاد	Huoaa ahe ustad	The text is neutral.	Neutral
42	مون کي آهي ڪم لنڊن ۾	Moun khe ahe kam London maen	The text is neutral.	Neutral
43	تون رهين ٿو نواب شاهه ۾	Toun rahen tho Nawabshah maen	The text is neutral.	Neutral
44	تو کاڌو انب	Toun khado amb	The text is neutral.	Neutral
45	هوءَ ڊوڙي ٿي روڊ تي	Huoa dore thi road te	The text is neutral.	Neutral
46	مان ڪيان ٿي پسند پيزا	Maa'n kayan thi pasand pizza	The text is neutral.	Neutral
47	هي ڪري ٿو نظر انداز مون کي	He kre tho nazarandaz moun khe	The text is neutral.	Negative
48	هي آهي هوشيار	He ahy hushiyar	The text is neutral.	Positive
49	مان ڪيان ٿي پسند ڪيلا	Maa'n kayan thi pasand kela	The text is neutral.	Neutral
50	ڪيان ٿي مان پسند ڪيلا	Kayan thi Maa'n pasand kela	The text is neutral.	Neutral
51	مان کيڏان ٿي فٽ بال	Maa'n khedan thi football	The text is neutral.	Neutral
52	مان کيڏان ٿي ڪرڪيٽ	Maa'n khedan thi circket	The text is neutral.	Neutral
53	مان پڙهان ٿو ڪتاب	Maa'n parhan tho kitab	The text is neutral.	Neutral
54	تون مدد ڪندي منهنجي	Toun madad kande mounhje	The text is neutral.	Positive
55	تون هليو وڃ هاڻي	Toun halyo wanj hanne	The text is neutral.	Neutral
56	ۂ گهر ۾ بيڊ تي ڪچرو آهي	Kichro ahy bad te aen ghar mean	The text is neg.	Negative
57	بيڊ کي ٿوڙين ٿو	Torin tho bad khe	The text is neg.	Negative
58	تون پهنجي گهر وارن جي لاء پازيٽو سوچيندو آهين	Toun pahinje ghar waran je parhae je lae Positive sochindo ahin	The text is pos.	Positive
59	علم حاصل ڪرڻ هڪ پازيٽو رستو آهي	Ilim hasil karan hik Positive rasto ahy zindagi guzarin jow	The text is pos.	Positive
60	علم سکي ڪري بين کي سيکارڻ پازيٽو ڳاله آهي	Ilim sikhi kry biyan khe saikharinn Positive galh ahy	The text is pos.	Positive
61	توهان جو بيڊ ء ڪرسي ڀڳل آهي	Tawha jow ahy bhaghal bad aen kursi	The text is neg.	Negative
62	تون هليو وڃ هاڻي مهرباني ڪري	Toun halyo wanj hanne maherbani kre	The text is neutral.	Neutral
63	علي سمهي ٿو	Ali sumhe tho	The text is neutral.	Neutral

#	Sindhi Sentence	Romanized Sindhi Sentence	Sentiment Analysis	Ground through value
64	سارا کائي ٿي انب	Sara khae thi amb	The text is neutral.	Neutral
65	ڪير آهين تون	Kair ahen Toun	The text is neutral.	Neutral
66	پيءُ جو نالو	Peeu joww nalo	The text is neutral.	Neutral
67	ماءُ جو نالو	Maau joww nalo	The text is neutral.	Neutral
68	ڀاءُ جو نالو	Bhau joww nalo	The text is neutral.	Neutral
69	ڀيڻ جو نالو	Bheen joww nalo	The text is neutral.	Neutral
70	مان ڪيان ٿي پسند چانور	Mai kayan thi pasand chanwar	The text is neutral.	Neutral
71	صنم آهي هتي	Sanam ahe hete	The text is neutral.	Neutral
72	هي آهي منهنجي ماءُ	He ahy mounhje maau	The text is neutral.	Neutral
73	هي آهي منهنجو پيءُ	He ahy mounjoww peeu	The text is neutral.	Neutral
74	علي آهي هتي	Ali ahe hete	The text is neutral.	Neutral
75	سارا آهي استاد	Sara ahe ustad	The text is neutral.	Neutral
76	اقرا آهي ڊاڪٽر	Iqra ahy doctor	The text is neutral.	Neutral
77	ماڻهو پسندڪن ٿاچانور	Mannho pasand Kan tha chanwar	The text is neutral.	Neutral
78	ماڻهو پسند ڪن ٿا اڀاريل چانور	Mannho pasand kan tha umarial chanwar	The text is neutral.	Neutral
79	ماڻهو پسند ڪن ٿا چانهن	Mannho pasand kan tha chanhe	The text is neutral.	Neutral
80	توهان ڀڳل بيڊ تي ويٺا آهيو	Tawha waitha ahiyo bhaghal bad te	The text is neg.	Negative
81	توهان بدسورت آهيو	Tawha bad sourat aahiyo	The text is neg.	Negative
82	اسان سڀني کي هڪ ٻئي جي لاء پازيٽو سوچ رکڻ گهرجي	Asan sabhni khy hik biay je lae Positive soach rakhan ghurjy	The text is pos.	Positive
83	توهان جو ننڍو ڀاء ڏاڍو بدڪردار آهي	Tawha jow nandho bhau dadho bad kirdar ahy	The text is neg.	Negative
84	مسجد ۂ گهر کي ساف رکڻ پازيٽو ڳاله آهي	Masjid aen ghar khy saff rakhan Positive soach ahy	The text is pos.	Positive
85	تون کيڏين ٿي مون سان	Toun kheden thi moun san	The text is neutral.	Neutral
86	ڊبو آهي ڳاڙهو	Dabho ahy gahro	The text is neutral.	Neutral
87	مان ڏيني ٻلي	Maa'n detho bhli khe	The text is neutral.	Neutral
88	ڪتو آهي هتي	Kuto ahe hete	The text is neutral.	Neutral
89	تون ڳالهين ٿو انگلش	Toun galhaeen tho English	The text is neutral.	Neutral
90	اسان کي نه وساريو	Naa wesaryo asann khe	The text is neutral.	Neutral

#	Sindhi Sentence	Romanized Sindhi Sentence	Sentiment Analysis	Ground through value
91	نه ڊرڇو	Naa drejow	The text is neutral.	Neutral
92	تون ڳالهايو ڪوڙ	Toun galhayo koor	The text is neutral.	Negative
93	کٽو چانهن	Khanno chahen	The text is neutral.	Neutral
94	نه اچو هتي	Naa acho hete	The text is neutral.	Negative
95	ٻلي ڊورڙي ٿي	Bilii dorhe thi	The text is neutral.	Neutral
96	هي آهي ٿوم	He ahy tom	The text is neutral.	Neutral
97	وڻ آهي وڏو	Wanu ahe wadho	The text is neutral.	Neutral
98	سج آهي گرم	Sijh ahy garam	The text is neutral.	Neutral
99	مان کاڌو ڪيڪ	Maa'n khado cake	The text is neutral.	Neutral
100	مان استاد کان نه ٿو ڊرجان	Maa'n Naa tho drjan tho ustad khan	The text is neutral.	Negative

Table I shows the data set of the research, which depends on one hundred sentences of RST. The data was obtained from the RST, which was obtained from different sources. Sentiment Analysis RST was done on the online Python tool. The obtained results of sentiment analysis are in the form of positive text, neutral text, and negative text. From the output results of the Python tool, 573 sentences have neutral sentiments, 8 sentences have positive sentiments, and 12 sentences have negative sentiments. After the task was performed on the tool, all sentiments were compared based on ground truth values.

IV. Algorithm for the Sentiment Analysis of Romanized Text (Sindhi)

```
Start
Take input: In sentence form of RST
Apply: apply one by one sentence till 100 sentences.
Analysis on tool: Analysis on the tool results (Neu, Neg and Positive) based on input sentence.
Evaluation: based on the ground truth values.
Process: Sentence run on tool Repeat steps 2 to 6 when get appropriate
End
```

V. Evaluation of Romanized Sindhi Text

Results are evaluated based on RST's sentiment analysis, which is carried out using a web-based Python tool and ground truth values. The one hundred sentences in the RST are subjected to a sentiment analysis examination. The evaluation of RST is based on two key elements of the research, the first being the ground truth values and the second being the online Python tool. The results came from tool are in three categories of the sentiments positive, neutral and negative as shown in Table II and Fig. 3. According to the results of the Python tool, 573 sentences had neutral, 12 sentences had negative, and 8 sentences had positive sentiments on the sentences of RST. According to the results of ground truth values 323 sentences had neutral meaning, 64 sentences had negative meaning, and 208 sentences had positive meaning in the RST.

TABLE II. SENTIMENTS ANALYSIS OF RST PYTHON VS GROUND TRUTH VALUES.

S. No.	Sentiment	Analysis By Python	Ground Truth Values
1	Neutral	86	76
2	Negative	9	17
3	Positive	5	7
Total		100	100

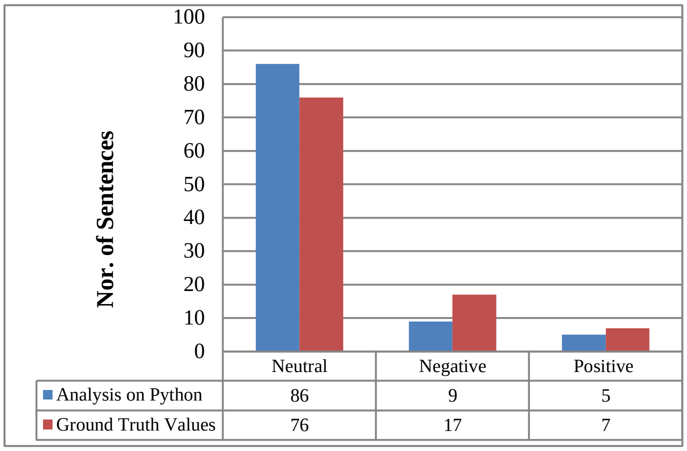


Fig. 3. Comparison of sentiment analysis (python tool and ground truth values).

According to the results of RST on Python, 86% of the sentences have neutral sentiments, 9% of the total results of sentiment analysis have negative sentiments, and only 5% of the total sentences of Romanized Sindhi Text have positive sentiments, as shown in Fig. 4.

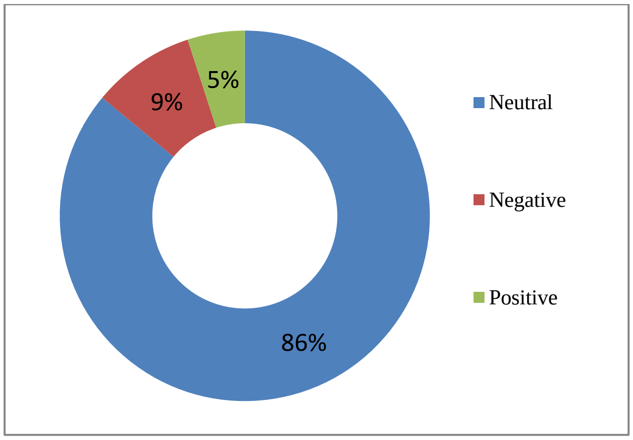


Fig. 4. Analysis of sentiments on online python tool.

As the results of RST on Ground Truth Values of sentiment analysis show, 76% of the sentences have neutral sentiments, 17% of the total results of sentiment analysis have negative sentiments, and only 7% of the total sentences of RST have positive sentiments. Analysis of sentiments for RST on Python Tool is shown in Fig. 5.

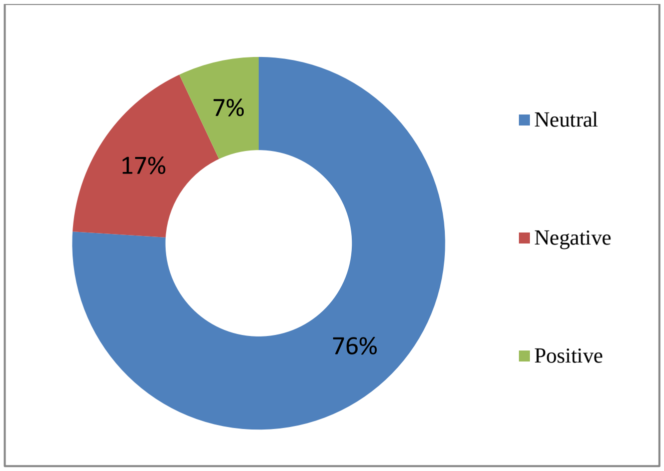


Fig. 5. Analysis of sentiments on ground truth values.

VI. Accuracy of Sentiment Analysis of Romanized Sindhi Text

Sentiment analysis of RST has been done using the online Python tool for 100 sentences. In sentiment analysis, positive, negative, and neutral sentiments were measured. Output results of the sentiment from tool were compared with the Ground Truth Value of the sentences. After performing the task of sentiment analysis, the accuracy of the output results from the tool was measured on the basis of the ground truth values of the sentences [16, 17].

For the accuracy evaluation, a confusion matrix has been created on the basis of ground truth values, as shown in Table III, True Negative (TNeg), True Positive (TP), True Neutral (TNeu), False Neutral (FNeu), False Positive (FP), and False Negative (FNeg) [16]. After calculating the values of the parameters from the confusion matrix, the accuracy of the RST was measured on an online calculator, and the value is 87.02% on the basis of ground truth values as shown in Fig. 6. Also, error ratio of 12.98% was calculated on the basis of accuracy found on online calculator of confusion matrix.

TNeg = Both values (Ground truth value and tool (Python) are negative.

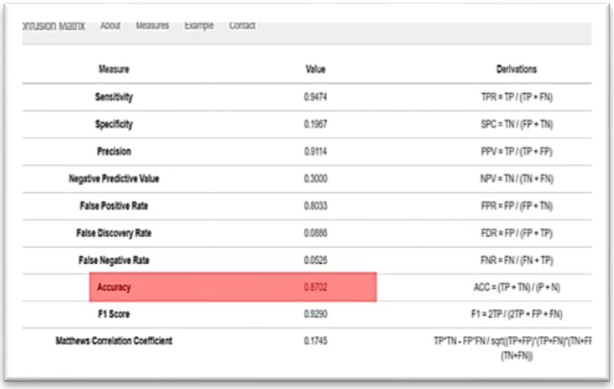
TP = Both values (Ground truth value and tool (Python) are positive.

FP = Ground truth values are negative or neutral and values from tool (Python) are positive.

FNeg = Ground truth values are positive or neutral and values from tool (Python) are negative.

TABLE III. Confusion Matrix on the Basis of Ground Truth Values

S. No.	FNeg	FP	TP	TNeg
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	0	1	0
5	0	0	1	0
6	0	0	1	0
7	0	0	0	1
8	0	0	1	0
9	0	0	1	0
10	0	1	0	0
. . .	. . .	. . .	. . .	. . .
Total	2	8	81	9



Measure	Value	Derivations
Sensitivity	0.9474	TPR = TP / (TP + FN)
Specificity	0.1967	SPC = TN / (FP + TN)
Precision	0.9114	PPV = TP / (TP + FP)
Negative Predictive Value	0.3000	NPV = TN / (TN + FN)
False Positive Rate	0.8033	FPR = FP / (FP + TN)
False Discovery Rate	0.0886	FDR = FP / (FP + TP)
False Negative Rate	0.0526	FNR = FN / (FN + TP)
Accuracy	0.8702	ACC = (TP + TN) / (P + N)
F1 Score	0.9290	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.1745	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FF (TN+FN))

Fig. 6. Accuracy of confusion matrix on online calculator.

VII. Issues of Sentiment Analysis of Romanized Sindhi Text

Sentiment analysis of RST has been done on the online Python tool for 100 sentences. But during, before, and after performing the task of sentiment analysis on RST, faced issues with the completion of this task [18, 19]. While performing the task of sentiment analysis on RST, positive sentences were not identified by the tool (Python), but after the characters of the Romanized text were changed, and then the results came. Other issues are discussed below:

1) Even when a single word (positive or negative) is used for sentiment analysis on tool, output result was a neutral sentiment.

2) Input sentences were interrogative used for sentiment analysis, but the results were neutral sentiment.

3) When Punctuation was used as input in sentences, the results were mostly neutral sentiment.

4) Input sentences were used as Negative, but the results were neutral sentiments. But for the negative sentences of English word, bad is used in Romanized Sindhi sentences, the result was negative.

5) Input sentences were used as Positive, but the results were neutral sentiments. But for the positive sentences of English word, positive were used in Romanized Sindhi sentences, the result was positive.

6) When country name comes in any sentences output comes as neutral.

7) When (ignore) word of English comes in sentences with subject+verb+Object of Romanized Sindhi text used on tool and the result came neutral.

8) Neutral output comes when sentences without subject are used on tool.

VIII. Conclusion

In this research, sentiment analysis has been done on Romanized Sindhi text using a machine learning tool (a hybrid approach) and ground truth values. The machine learning tool is a Python online tool that is freely available to perform different tasks of NLP by using input text. In this task, we used a data set of RST. Sentiment Analysis of RST has been done on 593 sentences, and the sentiments of the sentences are positive, negative, and neural. According to the results for the sentences of RST on the Python tool of sentiment analysis, 86% of the total sentences have neutral sentiments and as per Ground Truth Values of Sentiment Analysis 76% of the sentences have neutral sentiments. The overall accuracy of the sentiment analysis was measured from the confusion matrix, and the accuracy is 87.02%. Sentiment analysis of RST has been done using the online Python tool for the one hundred sentences. In sentiment analysis, positive, negative, and neutral sentiments were measured. Output results of the sentiment from tool were compared with the Ground Truth Value of the sentences. After performing the task of sentiment analysis, the accuracy of the output results from the tool was measured on the basis of the ground truth values of the sentences [16, 17].

Acknowledgments

The corresponding author would like to thank Dr. Suriani Sulaiman (Assistant Professor, IIUM Malaysia) for assistance in research and designed the simulated work.

References

[1] Mabunda J.G., Jadhav A., Ajoodha R. Sentiment analysis of student textual feedback to improve teaching. Interdisciplinary Research in Technology and Management 2021 Sep 14; (pp. 643-651). CRC Press.

[2] Al-Garadi M.A., Yang Y.C., Cai H, Ruan .Y, O'Connor K., Graciela G.H., Perrone J., Sarker A. Text classification models for the automatic detection of nonmedical prescription medication use from social media. BMC medical informatics and decision making. 2021 Dec;21(1):1-3.

[3] Alatabi H.A., Abbas A.R. Sentiment analysis in social media using machine learning techniques. Iraqi Journal of Science. 2020 Jan 27:193-201.

[4] Alowisheq A., Alhumoud S, Altwairesh N, Albuhairi T. Arabic sentiment analysis resources: a survey. InSocial Computing and Social Media: 8th International Conference, SCSM 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17–22, 2016. Proceedings 8 2016 (pp. 267-278). Springer International Publishing.

[5] S Anjali Devi and S Sivakumar, "A Hybrid Ensemble Word Embedding based Classification Model for Multi-document Summarization Process on Large Multi-domain Document Sets" International Journal of Advanced Computer Science and Applications(IJACSA), 12(9), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120918.

[6] Rehman Z., Anwar W., Bajwa U.I. Challenges in Urdu text tokenization and sentence boundary disambiguation. InProceedings of the 2nd workshop on south southeast asian natural language processing (WSSANLP) 2011 Nov (pp. 40-45).

[7] Kyle K. Natural language processing for learner corpus research. International Journal of Learner Corpus Research. 2021 Mar 1;7(1):1-6.

[8] Asif M., Ishtiaq A., Ahmad H., Aljuaid H., Shah J. Sentiment analysis of extremism in social media from textual information. Telematics and Informatics. 2020 May 1;48:101345.

[9] Yoo S., Song J., Jeong O. Social media contents based sentiment analysis and prediction system. Expert Systems with Applications. 2018 Sep 1;105:102-11.

[10] Sodhar I.N., Jalbani A.H., Buller A.H., Channa M.I., Hakro D.N. Sentiment analysis of Romanized Sindhi text. Journal of Intelligent & Fuzzy Systems. 2020 Jan 1;38(5):5877-83.

[11] Pavan K., Tandon N., Varma V. Addressing challenges in automatic language identification of romanized text. InProceedings of 8th International Conference on Natural Language Processing (ICON-2010) 2010.

[12] Sodhar I.N., Jalbani A.H., Channa M.I., Hakro D.N. Parts of speech tagging of Romanized Sindhi text by applying rule based model. IJCSNS. 2019 Nov;19(11):91.

[13] Lakizadeh A., Zinaty Z. A novel hierarchical attention-based method for aspect-level sentiment classification. Journal of AI and data mining. 2021 Jan 1;9(1):87-97.

[14] Divyapushpalakshmi M., Ramalakshmi R. An efficient sentimental analysis using hybrid deep learning and optimization technique for Twitter using parts of speech (POS) tagging. International Journal of Speech Technology. 2021 Jun;24:329-39.

[15] Salur M.U., Aydin I. A novel hybrid deep learning model for sentiment classification. IEEE Access. 2020 Mar 23;8:58080-93.

[16] Chicco D., Tötsch N., Jurman G. The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData mining. 2021 Dec;14(1):1-22.

[17] Rabab Emad Saudy, Alaa El Din M. El-Ghazaly, Eman S. Nasr and Mervat H. Gheith, "A Novel Hybrid Sentiment Analysis Classification Approach for Mobile Applications Arabic Slang Reviews" International Journal of Advanced Computer Science and Applications(IJACSA), 13(8), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130849.

[18] Sodhar I.N., Jalbani A.H., Channa M.I. Identification of issues and challenges in romanized Sindhi text. International Journal of Advanced Computer Science and Applications. 2019 Sep;10(9).

[19] Sodhar, I. N., Buller, A. H., Sulaiman, S., & Sodhar, A. N. (2022). Word by Word Labelling of Romanized Sindhi Text by using Online Python Tool. International Journal of Advanced Computer Science and Applications, 13(8). http://dx.doi.org/10.14569/IJACSA.2022.0130831.

Consolidated Definition of Digital Transformation by using Text Mining

Mohammed Hitham M.H[1], Hatem Elkadi[2], Neamat El Tazi[3]
PhD Student[1] , Associate Professor[2, 3]
Faculty of Computers & Artificial Intelligence, Cairo University, Cairo, Egypt[1, 2, 3]

***Abstract*—Digital transformation has become essential for the majority of organizations, in both public and private sectors. The term "digital transformation" has been used (and misused), so frequently that it is now somewhat ambiguous. It has become imperative to give it some conceptual rigor. The objective of this study is to identify the major elements of digital transformation as well as develop a proper definition for DT in the public and private sectors. For this purpose, 56 different definitions of DT collected from the available literature were analyzed, and we found that they extracted elements from definition of DT manually. So, text mining (TF-IDF and Fp-tree) techniques are used to identify the major constituents and finally consolidate in generic DT definitions. The approach consists of five phases: 1) collecting and classifying DT definitions; 2) detecting synonyms; 3) extracting major elements (terms); 4) discussing and comparing DT elements; 5) formulating DT definitions for different business categories. An evaluation tool was also developed to assess the level of DT elements coverage in various definitions found in the literature, and, as a validation, it was applied to the formulated definitions.**

Keywords—Digital transformation; text mining; association rules; FP tree

I. Introduction

In a world of emerging and continuous change, digital transformation (DT) has become a necessity for most organizations, both in the private and public sectors. The word "digital transformation" has been used in a broad sense to include many ideas that lead to widely divergent viewpoints. Few attempts have been made to define DT. Based on a review of 56 definitions, we could identify two fundamental approaches to defining DT: One is based on the scope of the study [1-37], [39-42] and the other is based on the perspective of expert(s) interview as [38] in the private sector or in [43] for the public sector. According to [1], the phrase "digital transformation" does not have a generally accepted definition. Without properly defining the DT, proper assessment and proposition of DT solutions (Framework, Model, or Architecture) are not possible. In a recent study [1], an effort was made to define digital transformation, but, this study had two limitations: a) it did not classify the prior definitions and b) it extracted the manually DT elements (based on their frequency). To the best of our knowledge, no study has so far defined DT elements using text mining techniques. To go beyond these limitations, we propose a comprehensive approach, using text mining algorithms to objectively extract the DT elements. We categorize the prior DT definitions into two groups: in the public sector and in the private sector. In this study, text mining is used to answer the research question: What are the key elements of the DT definitions in the public and private sectors as well as in general (all definitions)? The rest of this paper is organized as follows: In Section II, the proposed an approach is described. In Section III, results of text mining techniques are presented. In Section IV; results of digital transformation elements are discussed. In Section V, definitions of DT are proposed. In Section VI, we present a tool to asses various DT definitions. Finally, the conclusion, limitations and future work are presented.

II. Proposed Approach for Defining DT Elements

Our general approach for defining major elements in digital transformation definitions is outlined in Fig. 1. We will give a brief description of each phase as follows:

A. Phase One: Collecting and Classifying DT Definitions

The first phase is responsible for gathering existing definitions from recent literature specialized academic literature as well as from the websites of specialized private companies such as IBM, Google, and Oracle... (Our data set included 56 definitions).

After reviewing them, it has been found:

- Several publications [1-20] in the literature do not specify to which type their definitions applied. We will try to define the appropriate type for them later.
- 21 Private sector definitions from the companies' perspectives (14 definitions) ([21] [23-31] [37] [39-41], and from researchers' perspectives (7 definitions) [22], [32-36] [38].
- 15 Public sector definitions from researchers' perspectives (15 definitions) [42-56].

B. Phase Two: Replacing Synonyms

Analysis of the acquired dataset revealed the existence of specific bigrams and n-grams (e.g. big data, business process, business model, etc) that must appear as block. Thus, the synonym identification phase was proposed, where these n-grams are ligated and replaced in the dataset, for example big data replace with (Bigdata). We also replace some words like "artificial intelligence" and "internet of things" with their shorthand (AI, IOT).

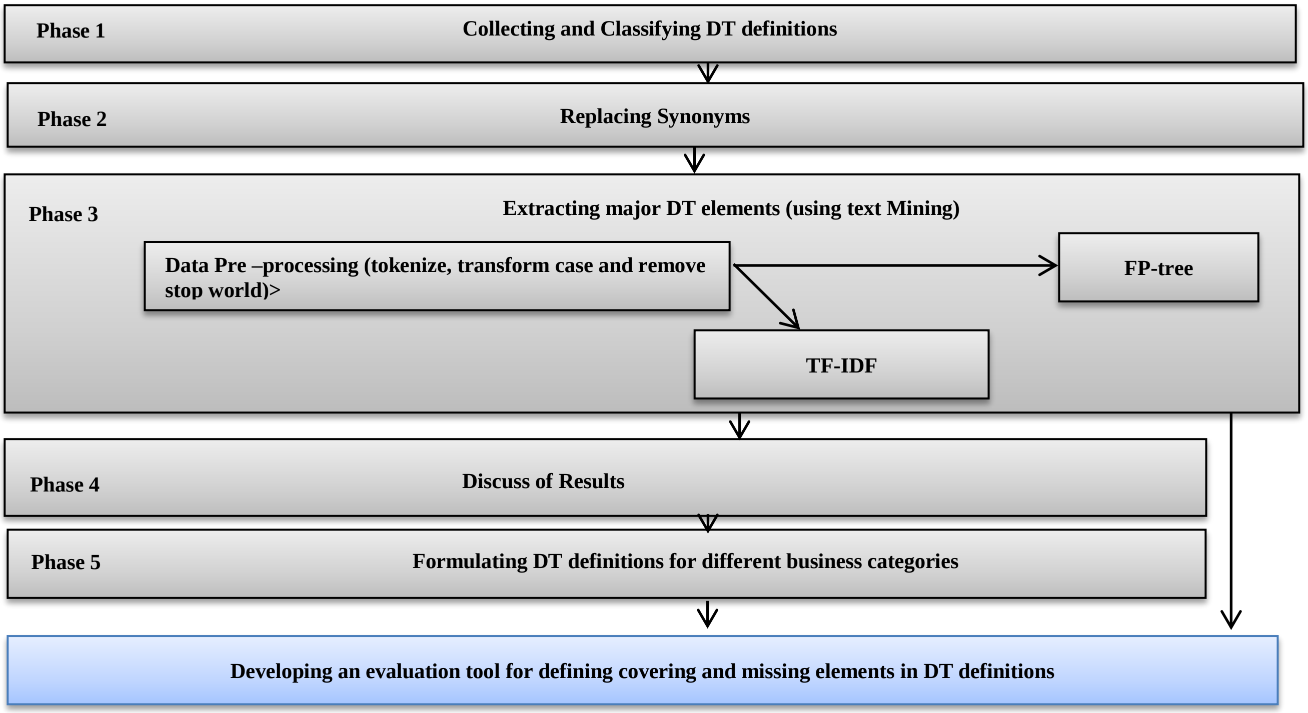

Phase 1
Collecting and Classifying DT definitions
Phase 2
Replacing Synonyms
Phase 3
Extracting major DT elements (using text Mining)
Data Pre –processing (tokenize, transform case and remove stop world)>
FP-tree
TF-IDF
Phase 4
Discuss of Results
Phase 5
Formulating DT definitions for different business categories
Developing an evaluation tool for defining covering and missing elements in DT definitions

Fig. 1. Proposed approach for defining DT elements in DT definitions.

C. *Phase Three: Extracting Major DT Elements (Using Text Mining)*

The main elements (terms) of the definitions of digital transformation are extracted from the 56 collected definitions using traditional text mining techniques. This requires proper preprocessing of the acquired text (tokenizing, removal of stop words, stemming, and case transformation). The TF-IDF method, being the most widely used method [60] in the literature, was used to identify the most frequently used terms and according to [61], Fp-tree algorithm offered good results for extracting association rules from text. Therefore, in this work we used the TF-IDF method (one gram) to extract frequently occurring terms from DT definitions and used Fp trees to extract association DT elements.

III. Results of Text Mining Techniques (Results of Phase Three)

Following results are obtained on laptop running Dell-core i7, Windows 10. The approach was implemented using Python 3.7.4 and RapidMiner Studio-9.10.1. Table I, shows the experimental parameters for text mining algorithms. The confidence in the Fp-tree in all DT categories is 1.

We will discuss the results of applying the TF-IDF and FP tree algorithms to DT definitions elements as follows:

TABLE I. Parameters Settings of Text Mining Algorithms

Category of definitions	Frequency in Fp-tree	Min acceptable weight in TF-IDF
Public	3	0.14
Private	3	0.14
General (All definitions)	4	0.14

A. *Method 1: Applying TF-IDF*

Firstly, we use TF-IDF to define the most frequently used words.

- The main elements in the public definitions.

Table II shows the results when applying TF-IDF on DT definitions in public definitions.

TABLE II. TF-IDF for DT Definitions in Public

Words	Weight
businessprocess	1
government	0.78
service	0.67
digitaltechnology	0.50
digital	0.39
citizen	0.33
digitaltransformation	0.28
bigdata	0.22
businessmodel	0.22
egovernment	0.22
Public sector	0.22
leverage	0.22
datamining	0.17
change	0.17

- The main elements in the private definitions

Table III shows the results when applying TF-IDF on DT definitions in private definitions.

- The main elements in the all definitions (general).

Table IV shows the results when applying TF-IDF on DT definitions in all definitions.

TABLE III. TF-IDF FOR DT DEFINITIONS IN PRIVATE

Words	Weight
digitaltechnology	0.52
business	0.48
businessprocess	0.43
digital	0.38
businessmodel	0.33
change	0.33
customer	0.24
operation	0.24
cloudcomputing	0.19
customerexperience	0.19
iot	0.19
innovation	0.14
transformation	0.14
organization	0.14

TABLE IV. TF-IDF FOR DT DEFINITIONS IN ALL

Words	Weight
businessprocess	0.60
digitaltechnology	0.42
change	0.36
business	0.36
digital	0.34
businessmodel	0.32
organization	0.30
government	0.25
service	0.21
customer	0.16
process	0.14
value	0.14
digitaltransformation	0.14

B. *Method 2: Applying Association Rules*

In this sub-section, we will apply association rules (Fp-tree) to each category of DT definitions as follows:

- The main elements in the public definitions.

The results of running the Fp-growth algorithm are shown in Table V. It can be seen that the final set contains three words that appear to be associated with one another: businessmodel, businessprocess and digitaltechnology.

TABLE V. ASSOCIATION RULES IN PUBLIC DEFINITIONS

Premises	Conclusion	Support	Confidence
businessmodel	businessprocess	0.22	1
Citizen	businessprocess	0.22	1
bigdata	digitaltechnology	0.22	1
service, citizen	businessprocess	0.167	1
leverage	businessprocess	0.167	1
government, citizen	businessprocess	0.167	1
leverage	businessmodel	0.167	1

- The main elements in the private definitions.

The results of running the Fp-growth algorithm are shown in Table VI. We would be able to see that the final set contains three words that appear to be associated with one another: customerexperience, businessmodel and businessprocesses.

- The main elements in general DT definitions (all definitions).

The results of running the Fp-growth algorithm are shown in Table VII. We would be able to see that the final set contains four words that appear to be associated with one another: business-model, businessprocess, DigitalTechnology, and digital.

TABLE VI. ASSOCIATION RULES IN PRIVATE DEFINITIONS

Premises	Conclusion	Support	Confidence
BusinessProcess	DigitalTechnology	0.25	1
Improve	BusinessModel	0.15	1
Operation ,Customerexperience	BusinessModel	0.143	1
businessmodel, operation	customerexperience	0.143	1
Change, Organization	BusinessProcess	0.143	1
Digital, Customer	BusinessProcess	0.143	1

TABLE VII. ASSOCIATION RULES IN GENERAL DEFINITIONS

Premises	Conclusion	Support	Confidence
Createvalue	BusinessModel	0.073	1
createvalue	Leverage	0.073	1
businessmodel, createvalue	Leverage	0.073	1
BusinessProcess	BusinessModel	0.073	1
Createvalue	BusinessModel,BusinessProces	0.073	1
BusinessProcess, customerexperiences	BusinessModel	0.073	1
BusinessProcess, createvalue	BusinessModel	0.073	1
People	DigitalTechnology	0.073	1
BusinessModel, DigitalTechnology	Digital	0.073	1
BusinessModel, customerexperiences	BusinessProcess	0.073	1
BusinessModel, Leverage	BusinessProcess	0.073	1
BusinessModel, createvalue	BusinessProcess	0.073	1
Createvalue, Leverage	BusinessModel, BusinessProcess	0.073	1

IV. PHASE FOUR: DISCUSSION OF RESULTS

Based on the results of applying the TF-IDF and Fp-tree to DT definitions, we can note that in the private definitions, the most important technologies are IoT and cloud computing, compared to data mining and big data in the public definitions. We also found that definitions in the private sector focus on business, the customer, and innovation, while in the public sector they focus on services, government and citizens. We can define intersecting elements between DT definition categories as shown in Fig. 2. In general digitaltechnology, digital, businessmodel, and businessprocess are intersecting elements across all DT definition categories. This suggests these are minimum elements to define DT. It can be noted that all intersection elements originated from method one (TF-IDF), except (BusinessProcess->BusinessModel) l which identified from method two (Fp-tree) in the intersection between public and general definitions. So we didn't draw a Venn diagram in FP-tree.

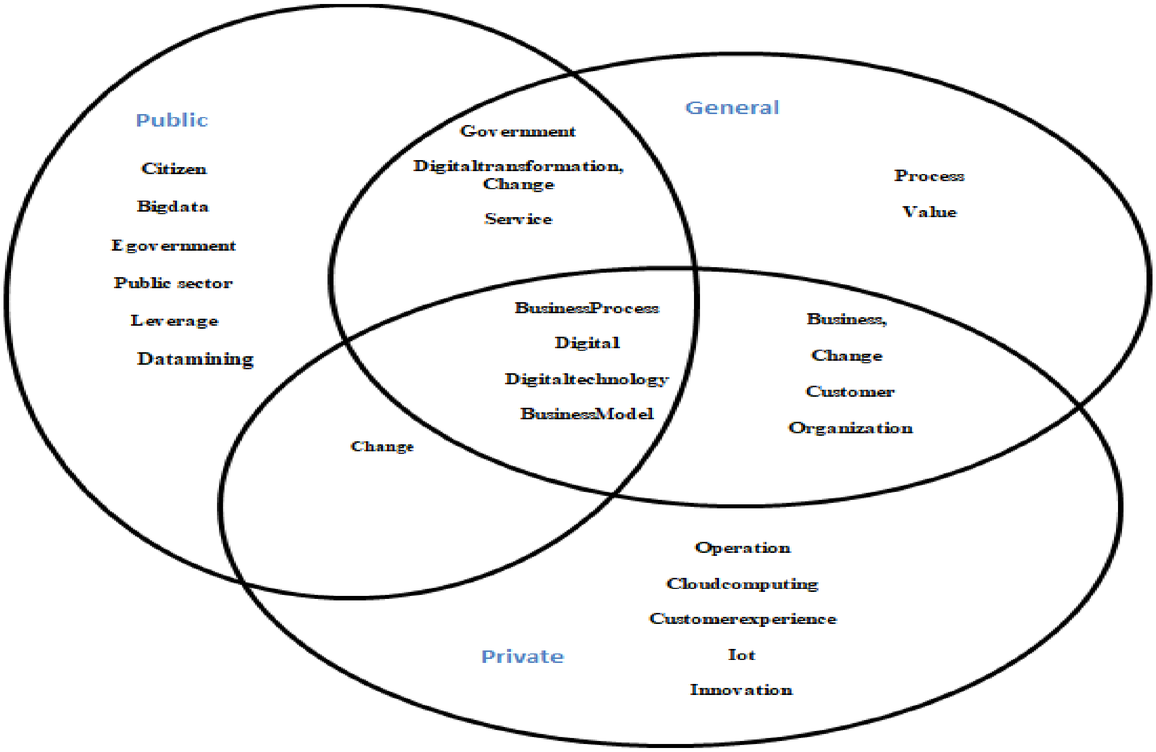


Fig. 2. Intersection between DT Elements in DT Definitions (TF-IDF).

V. Phase Five: Proposing DT Definitions

Based on the DT elements that have been defined before, we can define DT in three categories as follows:

- DT Definition in General.

DT is a process that leverages digital technologies to change an organization of government or business, business model and business processes, to create value for consumer (customers or citizens).

- DT Definition in Public.

DT is a process that leverages digital technologies (bigdata, data mining), to change government, business process, business model, services and citizens.

- DT Definition in Private.

DT is a process that leverages digital technologies (big data, cloud computing and IOT) to change an organization, business model, business processes, to create value for customer.

VI. Proposing a Tool to Asses Various DT Definitions

As mentioned before, we have two main categories of DT definitions (public, private) and combined between them to create a new category called "general". Each category contains a set of elements, as discussed above. In the following, we will try to find the percentage of covered elements by each definition in each category, as well as identify the percentage of missing elements for each definition. For this purpose, we developed algorithm 1. There are two inputs to this algorithm. The first input contains dataset that includes reference numbers and definitions (text). The second input contains a dictionary (data) where key is category: (Public, Private, and General) and value is DT elements for each category that contains two lists: List 0 contains words that appear in definitions in sequence as a block (come from the TF-IDF); whereas List 1 contains words that appear in definitions in sequence but not as a block (come from the Fp-tree algorithm).We follow several steps to calculate the percentage of covering and missing DT elements, which are:

A. For Each Algorithm, the Percentage of Words Covered is Calculated in Each Definition as Follows

- TF-IDF

$$CTF_{IC} = \frac{NW_{IC}}{TW_c} \qquad (1)$$

Where CTF is the percentage of words covered in definitions I (1 ...59) in term frequency, c is category (c set of index: private, public, general), NW is Number of words covered by each definition I in each category c, and TW is total number of words in each category according to TF- IDF.

- FP-tree

$$CTF_{IC} = \frac{NW_{IC}}{WT_c} \quad (2)$$

Where CFP is the percentage of words covered in definition I in Fp-tree, c is category (c set of index: private, public, general), NW is Number of words covered by each definition I in each category c and TW is total number of words in each category according to FP algorithm.

B. The Total Percentage of Words Covered is Calculated in Each Definition as Follows

$$TC_{IC} = \frac{CTF_{IC}+CFP_{IC}}{2} \quad (3)$$

Where TC is total covered words in each definition I, in-TF and Fp-tree.

C. The Percentage of Words Missing in Each Definition is Calculated as Follows

$$MP_{IC} - 1 - TC_{IC} \quad (4)$$

Where MP is missing percentage in each defilation I in each category c.

VII. Discussion of Results (Applying Our Tool)

Table VIII shows an example of results of applying algorithm 1 to 56 definitions from the literature and our proposed definitions (3); see the link in the appendix for the complete results.

A. When Applying our Algorithm to Define Category of First 20 Definitions (from 1 to 20),

- It has been found that 80% of definitions are classified as private definitions.
- It has been found that 15% of definitions are classified as public definitions.
- It has been found that 5% of definitions are classified as general definitions.
- The definition that covered the most elements is [8] in general with (35.1%), in public with (23.85%) and in private with (29.65%).
- The definition that covered the lowest elements in public is [15] with (3.35%) and [4] in private with (7.9%) and in general with (6.5%).

B. When Applying our Algorithm to the Private Category (21 Definitions), it has been found that

- Proposed algorithms agree with (66.66%) in the classification of private sector definitions and differ (28.95%) as they were classified as definitions in the public sector and (4.76%) as a general category.
- The definition that covered the most elements is [34] with (28.95%).
- The definitions that covered the lowest elements are [24] and [30] with (7.9%).
- It has been found only three definitions covering elements in Fp-tree which are [23], [26], and [36] as shown in Table VIII.

C. When Applying our Algorithm to the Public Category (15 Definations), it has been found that

- Proposed algorithm agrees with (86.66%) in the classification of the public definitions and disagree (13.33%) as they were classified as private definitions.
- The definition that covered the lowest elements is [42] with (6.5%).
- The definition that covered the most elements is [47] with (26.5%).
- The definitions that covered most elements in TF are [47] with (53.3%) compared to [53] in Fp-tree with (7.7).

D. When Applying our Algorithm to All Definitions it has been found that

- Proposed algorithm agreed with 75% of the previous studies' classification of DT definitions private (21) and public (15) while disagreeing with 25%.

E. When Applying our Algorithm to our Definitions it has been found that

It can be seen that our definitions cover the largest percentage of the DT elements in general (all definitions) [57] with (42.5%), in the public definitions [58] with (38.5%), and in the private definitions [59] with (38.7%). Overall, our definitions have achieved the highest percentages in (TF-IDF, Fp), which gives us an indication that our definitions are more comprehensive.

Algorithm1: Algorithm to find the covering and missing percentage in DT definitions

```
Dictionary [] ←0 //empty dictionary
Dictionary← Loading dataset // (reference#, definitions)
DefCode←reference#  ,val←definitions
DictData [] ←0 //empty dictionary
DictData← Loading data
key← DT definitions category, vlaue← DT elements in each category(private, public, general)
Function check_definitions (DefCode, definitions, DictData). res[]←0 //empty list
  For key, value in DictData   Then
```

```
    For index, listData in value   Then    // loop in  list0 and list 1
         For val in listData Then // loop in each  item in list 0 and list 1
          If index ==0  and val in definitions  Then  // check if words as appear as block in definitions
              res. append ([DefCode, index, key])
              Else   Then //when index ==1
                res.append ([DefCode, index, key])
          Endif
       Endfor
    Endfor
  Endfor
End Function
res[]←0
For  DefCode, definitions in  Dictionary  Then
  res+=check_definitions (DefCode, definitions ,DictData)  //// return list of every word is true in definitions
Endfor
Function calculate covering missing percentage(DefCode, definitions ,DictData)
 for DefCode
   Calculate  Covering percentage  in each category  using equation 1,2
   Calculate  Total Covering percentage in  each category  using equation 3
   Calculate  Missing percentage  in each category  using equation 4
 Endfor
End Function
Total cover in private(tcp)[]←0
Total cover in public(tcpb)[]←0
Total cover in general(tcg)[]←0
Function  define category( DefCode, TCPPR, TCPPB, TCG)
 for DefCode
   if  (   TCPPR >   TCPPB )&(    TCPPR >    TCG )  Then
     category as private
    else if (     TCPPB >    TCPPR )&(    TCPPB >    TCG )  Then
      category as public
   else if (      TCG >    TCPPB )&(     TCG >     TCPPR )Then
      category as general
  else if (   TCG =   TCPPB =   TCG ) //  tcg ,  tcp ,  tcpb >0  Then
     category as general
   else   Then     // when    TCG =    TCPPB =   TCG =0
    category as NA
 End if
End for
End Function
```

TABLE VIII. EXAMPLE RESULT OF APPLYING ALGORITHM TO DEFINE COVERING / MISSING PERCENTAGE

Category	**Reference#**	**Cover in (TF-IDF) %**			**Cover in (Fp-tree)%**			**Total Cover in (TF-IDF and FP-tree)%**			**Missing Elements (not covered in both)%**			**Category Based on Previous Studies**	**Category Based on our Approach**
		General	**Private**	**Public**	**General**	**Private**	**Public**	**General**	**Private**	**Public**	**General**	**Private**	**Public**		
Not define	1	30.4	36.8	20	0	0	0	15.2	18.4	10	84.8	81.6	90	NA	private
	4	13	15.8	13.3	0	0	0	6.5	7.9	6.65	93.5	92.1	93.35	NA	private
	8	43.5	52.6	40	26.7	6.7	7.7	35.1	29.65	23.85	64.9	70.35	76.15	NA	General
	15	17.4	26.3	6.7	0	0	0	10.5	14.7	3.6	89.5	85.3	96.65	NA	public
	20	21.7	26.3	20	0	0	0	10.85	13.15	10	89.15	86.85	90	NA	private
Private	22	21.7	26.3	13.3	0	0	0	10.85	13.15	6.65	89.15	86.85	93.35	private	private
	23	26.1	42.1	26.7	0	0	7.7	13.05	21.05	17.2	86.95	78.95	82.8	private	private
	26	21.7	31.6	26.7	0	6.7	7.7	10.85	19.15	17.2	89.15	80.85	82.8	private	private
	34	43.5	57.9	33.3	0	0	0	21.75	28.95	16.65	78.25	71.05	83.35	private	private
	36	43.5	47.4	33.3	0	0	7.7	21.75	23.7	20.5	78.25	76.3	79.5	private	private
	38	17.4	15.8	20	0	0	0	8.7	7.9	10	91.3	92.1	90	public	public
Public	42	4.3	0	13.3	0	0	0	2.15	0	6.65	97.85	100	93.35	public	public
	51	26.1	21.1	40	0	0	0	13.05	10.55	20	86.95	89.45	80	public	public
	53	21.7	15.8	33.3	0	6.7	7.7	10.85	11.25	20.5	89.15	88.75	79.5	public	public
	54	26.1	21.1	40	0	0	0	13.05	10.55	20	86.95	89.45	80	public	public
	56	13	15.8	13.3	0	0	0	6.5	7.9	6.65	93.5	92.1	93.35	public	private
Our Proposed Definitions															
General	57	57	63.2	60	27.3	6.7	0	42.15	34.95	30	57.85	65.05	70	-	General
Public	58	39.1	36.8	66.7	0	0	10	19.55	18.4	38.35	80.45	81.6	61.65	-	public
private	59	56.5	68.4	53.3	13.3	9	0	34.9	38.7	26.65	65.1	61.3	73.35	-	private

VIII. CONCLUSION

Although digital transformation is a hot topic right now, there is no generally accepted definition, which has implications for both researchers and practitioners. Consequently, the goal of this study was to learn more about the concept of digital transformation. According to the analysis of previous definitions of digital transformation, we can divide them into two groups: in the private sector, in the public sector and create a new group called in general. We propose a comprehensive approach to defining major elements in DT definitions in each category as well as in general (all definitions). This approach consists of five phases. The first phase is used for collecting and classifying DT definitions. The second phase is responsible for synonyms and defining the words that must appear together. The third phase is responsible for extracting major DT elements in each category using text mining methods (Fp trees, TF-IDF). The fourth phase is used to discuss and compare DT elements. The fifth phase is used to propose new definitions of DT in the private, public, and general. In the end, we propose an assessment tool (algorithm) to identify the percentage of covered elements for each definition in each category and define the percentage of missing. The results of applying TF-IDF in general showed that: digitaltechnology, digital, businessmodel, businessprocess and change are common elements across all DT definition categories. This suggests these minimum elements to define either in private or in public. In the private category, our algorithm classified 66.66% of them as private, compared to 28.95% classified as public and 4.76% classified as general. While there are 86.66% of people who classify DT definitions in the public domain, and our algorithm puts them in that category compared to 13.13% in the private domain. The assessment tool agreed 75% with the previous classification of definitions and did not agree with 25% of them.

We also use the assessment tool to identify categories of definitions [1–20] that were not previously classified. The assessment tool classified 80% of them as private definitions, while classifying 15% as public definitions and 5% as general. Overall, when using our assessment tool to define category to all defilations (56), it has been found that the most definitions classified as private with 57.14% followed by public category with 39.28% and general 3.57%. This indicates that most definitions of digital transformation focus more on the private sector than others. It can also be noted that our proposed DT definitions covered the largest percentage of the DT elements in general (all definitions) with 42.15%, in private with 38. 7%, and in public with 38.35%. This shows that our suggested definitions are more thorough. This study was limited by the small number of definitions that were examined (56), and this shortcoming will be overcome in future study. We are looking forward to doing a lot of experiments using other text mining algorithms as well as trying to apply our approach to other domains.

REFERENCES

[1] Gong, Cheng, and Vincent Ribiere. "Developing a unified definition of digital transformation." Technovation 102 (2021): 102217.

[2] Kurmann, Philipp, and Brian Arpe. "Managing Digital Transformation-How organizations turn digital transformation into business practices." (2019).

[3] Vial, Gregory. "Understanding digital transformation: A review and a research agenda." The journal of strategic information systems 28.2 (2019): 118-144.

[4] Janssen, Marike Susan, and Jonas Merk. "How Digital Transformation Changes Work Design: A Butterfly Emerging from its Chrysalis?." (2019).

[5] Reis, João, et al. "Digital transformation: a literature review and guidelines for future research." World conference on information systems and technologies. Springer, Cham, 2018.

[6] Schallmo, Daniel, Christopher A. Williams, and Luke Boardman. "Digital transformation of business models—best practice, enablers, and roadmap." Digital Disruptive Innovation (2020): 119-138.

[7] Morakanyane, Resego, Audrey A. Grace, and Philip O'Reilly. "Conceptualizing Digital Transformation in Business Organizations: A Systematic Review of Literature." Bled eConference 21 (2017).

[8] Arribas, Veronica, and José A. Alfaro. "3D technology in fashion: from concept to consumer." Journal of Fashion Marketing and Management: An International Journal (2018).

[9] Morakanyane, R., Grace, A. A., & O’Reilly, P. (2017). Conceptualizing Digital Transformation in Business Organizations: A Systematic Review of Literature.

[10] Kohli, Rajiv, and Nigel P. Melville. "Digital innovation: A review and synthesis." Information Systems Journal 29.1 (2019): 200-223.

[11] Morakanyane, R., Grace, A. A., & O’Reilly, P. (2017). Conceptualizing Digital Transformation in Business Organizations: A Systematic Review of Literature, 427–443. https://doi.org/10.18690/978-961-286-043-1.30.

[12] Kotarba, Marcin. "Digital transformation of business models." Foundations of Management 10.1 (2018): 123-142.

[13] Demirkan, H., Spohrer, J. C., & Welser, J. J. (2016). Digital Innovation and StrategicTransformation

[14] Parviainen, Päivi, et al. "Tackling the digitalization challenge: how to benefit from digitalization in practice." International journal of information systems and project management 5.1 (2017): 63-77.

[15] Vial, Gregory. "Understanding digital transformation: A review and a research agenda." The journal of strategic information systems 28.2 (2019): 118-144.

[16] Solis, Brian. "The six stages of digital transformation maturity." Altimeter Cognizant (2016).

[17] Schuchmann, Daniela, and Sabine Seufert. "Corporate learning in times of digital transformation: a conceptual framework and service portfolio for the learning function in banking organisations." International Journal of Corporate Learning (iJAC) 8.1 (2015): 31-39.

[18] Mićić, Ljubiša. "Digital transformation and its influence on GDP." Economics 5.2 (2017): 135-147.

[19] Fitzgerald, Michael, et al. "Embracing digital technology: A new strategic imperative." MIT sloan management review 55.2 (2014): 1.

[20] Osmundsen, Karen, Jon Iden, and Bendik Bygstad. "Digital Transformation: Drivers, Success Factors, and Implications." MCIS. 2018.

[21] Gartner,2022<]https://www.gartner.com/en/information-technology/glossary/digital-transformation>.

[22] ar Negreiro and Tambiama Madiega, – June 2019 https://www.europarl.europa.eu/RegData/etudes/BRIE/2019/633171/EPRS_BRI(2019)633171_EN.pdf.

[23] IDC InfoBrief,Decmeber 2019. https://media.bitpipe.com/io_14x/io_149461/item_2091898/IDC_Mind%20the%20Gap%20on%20Your%20DX%20Journey.pdf.

[24] Deloitte,2022https://www2.deloitte.com/za/en/pages/digital/topics/digital-transformation.html.

[25] The 9Lenses Team,2021 <https://offers.9lenses.com/hubfs/Digital%20Transformation%20in%20the%20Consulting%20Industry.pdf>.

[26] Digital Adoption Team,2022< https://www.digital-adoption.com/digital-transformation/#what-is-digital-transformation >

[27] By Ouriel Lancry, Nate Anderson, Greg Caimi, Laurent Colombani, and Lucy Cummings, February 01, 2019. < https://www.bain.com/insights/scaling-your-digital-transformation/>.

[28] Martin Danoesastro, Grant Freeland, and Tom Reichert,May, 2017https://www.bcg.com/publications/2017/digital-transformation-digital-organization-ceo-guide-to-digital-transformation.

[29] Microsoft,2021<https://docs.microsoft.com/en-us/learn/modules/enable-digitaltransformation/#:~:text=Digital%20transformation%20is%20a%20business,manage%2C%20and%20transform%20their%20business>.

[30] forrester,2022<https://www.forrester.com/blogs/category/digital-transformation/#:~:text=Digital%20transformation%20means%20applying%20the,demands%20and%20new%20customer%20requirements>.

[31] Redhat,Published March 16, 2018 <https://www.redhat.com/en/topics/digital-transformation/what-is-digital-transformation>

[32] Pucihar, A., et al. "The Impact of Cloud-Based Digital Transformation on ICT Service Providers’ Strategies." 30TH Bled eConference: Digital Transformation–From Connecting Things to Transforming Our Lives (2017): 111.

[33] Abdallah, Yasser Omar, Essam Shehab, and Ahmed Al-Ashaab. "Towards Managing Digital Transformation in Manufacturing Industry: Theoretical Framework." Advances in Manufacturing Technology XXXIV. IOS Press, 2021. 21-26.

[34] Berghaus, Sabine, and Andrea Back. "Stages in Digital Business Transformation: Results of an Empirical Maturity Study." MCIS. 2016.

[35] Henriette, Emily, Mondher Feki, and Imed Boughzala. "Digital Transformation Challenges." MCIS. 2016.

[36] Agushi, Getoar. Understanding the digital transformation approach-a case of Slovenian enterprises: master's thesis. Diss. Univerza v Ljubljani, Ekonomska fakulteta, 2019.

[37] Oracle,2021<https://www.oracle.com/cloud/digital-transformation.html>

[38] Balakrishnan, Radhakrishnan, and Satyasiba Das. "How do firms reorganize to implement digital transformation?." Strategic Change 29.5 (2020): 531-541.

[39] SAP,2021<https://www.sap.com/mena-ar/insights/what-is-digital-transformation.html#:~:text=Digital%20transformation%20involves%20integrating%20digital,deliver%20customer%20experiences%20and%20benefits>

[40] Google,2021<https://cloud.google.com/learn/what-is-digital-transformation>

[41] IBM,2021<https://www.ibm.com/topics/digital-transformation#:~:text=Resources-,Defining%20digital%20transformation,experiences%20to%20processes%20and%20operations.&text=And%20ultimately%2C%20it%20changes%20customer%20expectations%20and%20creates%20new%20business%20opportunities>.

[42] European Commission (2013). Powering European public sector innovation: Towards a new architecture. Retrieved from

[43] Mergel, Ines, Noella Edelmann, and Nathalie Haug. "Defining digital transformation: Results from expert interviews." Government information quarterly 36.4 (2019): 101385.

[44] Akatkin, Yury, and Elena Yasinovskaya. "Data-centricity as the key enabler of digital government: Is Russia ready for digital transformation of public sector." International Conference on Electronic Governance and Open Society: Challenges in Eurasia. Springer, Cham, 2018.

[45] Koo, Eunjin. Digital transformation of Government: from E-Government to intelligent E-Government. Diss. Massachusetts Institute of Technology, 2019.

[46] Bertot, John Carlo, Elsa Estevez, and Tomasz Janowski. "Digital public service innovation: Framework proposal." Proceedings of the 9th International Conference on Theory and Practice of Electronic Governance. 2016.

[47] Carcary, M., Doherty, E., and Conway, G. 2016. "A Dynamic Capability Approach to Digital Transformation: A Focus on Key Foundational Themes," 10th European Conference on Information Systems Management: ECISM 2016: Academic Conferences and publishing limited, p. 20.

[48] Alhaqbani, A., Reed, D. M., Savage, B. M. and Ries, J. (2016) ‘The impact of middle management commitment on improvement initiatives in public organisations’, Business Process Management Journal, 22(5), pp. 924–938. doi: 10.1108/BPMJ-01-2016-0018

[49] Mićić, Ljubiša. "Digital transformation and its influence on GDP." ECONOMICS-Innovative and Economic Research 5.2 (2017): 135-147.

[50] Lemke, Florian, et al. "Stage models for moving from e-government to smart government." International Conference on Electronic Governance and Open Society: Challenges in Eurasia. Springer, Cham, 2019.

[51] Glybovets, Andrii, and Alhawawsha Mohammad. "E-government versus smart government: Jordan versus the United States." EUREKA: Social and Humanities 3 (2017): 3-11.

[52] Borras, John A. "The transformational government framework." Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance. 2011.

[53] Bjerke-Busch, Linn Slettum, and Arild Aspelund. "Identifying Barriers for Digital Transformation in the Public Sector." Digitalization. Springer, Cham, 2021. 277-290.

[54] [Nachit, Hicham, et al. "Digital Transformation in the Moroccan Public Sector: Drivers and Barriers." Available at SSRN 3907290 (2021).

[55] https://granicus.com/dictionary/digital-government-transformation/

[56] Schwab, Klaus. The fourth industrial revolution. Currency, 2017.

[57] Anderson, Russell K. Visual data mining: the VisMiner approach. John Wiley & Sons, 2012.

[58] D. J. Litman, “Natural language processing for enhancing teaching and learning.” in AAAI, 2016, pp. 4170–4176

[59] G. Gupta and S. Malhotra, “Text documents tokenization for word frequency count using rapid miner (taking resume as an example),” Int. J. Comput. Appl, pp. 0975–8887, 2015

[60] L.-P. Jing, H.-K. Huang, and H.-B. Shi, “Improved feature selection approach tfidf in text mining,” in Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on, vol. 2. IEEE,2002, pp. 944–946.

[61] Chaiwuttisak, Pornpimol. "Text Mining Analysis of Comments in Thai Language for Depression from Online Social Networks." Soft Computing for Biomedical Applications and Related Topics. Springer, Cham, 2021. 301-313.

APPENDIX

The below link content detail information about our work https://drive.google.com/drive/folders/1-sX2_8tKFmkXhMgLgCovVTRAD81Bo4O4.

Fast Hybrid Deep Neural Network for Diagnosis of COVID-19 using Chest X-Ray Images

Hussein Ahmed Ali[1], Nadia Smaoui Zghal[2], Walid Hariri[3], Dalenda Ben Aissa[4]
Microwave Electronics-Research laboratory Faculty of Sciences of Tunis, University Tunis El-Manar, Tunis El-Manar, Tunis[1, 4]
Control and Energy-Management Laboratory-(CEM Lab) ENIS, University of Sfax, Sfax, Tunisia[2]
LABGED Laboratory-Computer Science Department-Badji Mokhtar, Annaba University, Annaba, Algeria[3]

Abstract—**In the last three years, the coronavirus (COVID-19) pandemic put healthcare systems worldwide under tremendous pressure. Imaging techniques, such as Chest X-Ray (CXR) images, play an essential role in diagnosing many diseases (for example, COVID-19). Recently, intelligent systems (Machine Learning (ML) and Deep Learning (DL)) have been widely utilized to identify COVID-19 from other upper respiratory diseases (such as viral pneumonia and lung opacity). Nevertheless, identifying COVID-19 from the CXR images is challenging due to similar symptoms. To improve the diagnosis of COVID-19 using CXR images, this article proposes a new deep neural network model called Fast Hybrid Deep Neural Network (FHDNN). FHDNN consists of various convolutional layers and various dense layers. In the beginning, we preprocessed the dataset, extracted the best features, and expanded it. Then, we converted it from two dimensions to one dimension to reduce training speed and hardware requirements. The experimental results demonstrate that preprocessing and feature expansion before applying FHDNN lead to better detection accuracy and reduced speedy execution. Furthermore, the model FHDNN outperformed the counterparts by achieving an accuracy of 99.9%, recall of 99.9%, F1-Score has 99.9%, and precision of 99.9% for the detection and classification of COVID-19. Accordingly, FHDNN is more reliable and can be considered a robust and faster model in COVID-19 detection.**

Keywords—COVID-19; Chest X-ray (CXR); Deep Learning (DL); Convolutional Neural Network (CNN)

I. Introduction

The novel coronavirus disease (COVID-19) is a current pandemic that has sparked widespread concern worldwide. This virus causes severe respiratory illness [1]. Currently, COVID-19 is recognized as the worst disease on the earth. Usually, chest X-ray (CXR) imaging is used for radiographic examination of the suspected cases. In severely impacted or limited resource locations, CXR imaging is favored because of its accessibility, cheap cost, and speedy findings. Via Lenard and Crookes tubes, the first X-ray was coined by Wilhelm Conrad in 1895. The rapid development of the x-ray system and its low cost made it available in most clinical testing [2]. Patients can quickly get CXR images in their homes or quarantine facilities since CXR facilities are available even in the most distant locales. Recently, CXR images based on Artificial Intelligence (AI) have been utilized widely to detect COVID-19 instances [3, 4]. Given the fast spread of COVID-19, however, such testing might reduce the effectiveness of pandemic prevention and control. Therefore, AI techniques, such as Deep Learning (DL), are viable solutions for autonomous diagnosis due to their promising processing of visual data and a vast array of medical images [4].

This section investigates supreme papers that utilized Convolutional Neural Networks (CNN) and other DL architectures to diagnose COVID-19 via CXR images. The CXR image testing factor is determined by (DL) plans and consists mostly of measurements such as COVID-19 segmentation, data correction, and model training [5]. Manually, radiologists often perform COVID-19 classification. However, it is time-consuming, error-prone, and exhaustive since radiologists are expected to diagnose many COVID-19 patients. Furthermore, despite the dramatic increase in COVID-19 patients, the lack of qualified radiologists to make reliable diagnosis continues to be a major problem [6].

On the other hand, healthcare professionals take a sample from the nose to find whether it is a COVID-19 case. Nevertheless, the manual outcome of the test could be a false negative depending on the timing and quality of the test sample. The maximum accuracy, of this testing method, for COVID-19 infections, is 71% [6].

CXR is an alternate important diagnostic method for COVID-19 detection. However, if the CXR images require clarification, COVID-19 is sometimes diagnosed as another disorder. Due to faulty diagnoses, patients frequently receive incorrect medicines, complicating their health. This critical health issue to diagnose COVID-19 has inspired researchers to create more precise and automated CXR-based diagnostic approaches.

Machine learning (ML) methods and the most intense learning have been considered more accurate in identifying COVID-19 [7, 8]. The authors in [9] used DL methods-based CXR images to diagnose COVID-19. Although the presented models in the literature were validated with several CXR images, particularly the positive case images of COVID-19 cases, there was a class imbalance issue in previous studies. Additionally, the authors utilized the transfer learning approach in their model. The scholars applied dataset preprocessing to convert it into a gray level. Next, change the image intensity, resize and extract the best features, and use the transfer learning approach. The authors in [9] focused on getting high COVID-19 classification accuracy. This means applying the same preprocessing on CXR images developing them and then designing a new model in DL for COVID-19 diseases. Several authors (such as in [4, 10]) utilized only CNN to classify COVID-19 patients and didn’t achieve more accurate accuracy.

However, CNN alone is insufficient for reliable COVID-19 detection [6, 11]. Therefore, researchers explored the limitations of the existing models to propose crucial solutions that can increase detection accuracy.

Since the outbreak of COVID-19, the majority of state-of-the-art methods have primarily focused on using transfer learning techniques to implement their systems as shown in the recent survey published in [3]. One of the main challenges in using transfer learning techniques for detecting COVID-19 is the amount of time it takes to apply these methods to large datasets, especially when dealing with CXR images. This is because the model needs to be fine-tuned by retraining it on the new data, which can be time-consuming even with powerful computing resources. To address this issue, a study proposed using several image preprocessing techniques before inserting deep CNN. These techniques involve converting the images to grayscale, adjusting the image intensity, resizing and extracting the best features, and expanding features. This preprocessing transforms the data from two dimensions to one dimension, which can speed up implementation. Additionally, a unique fast deep learning proposal design was introduced to improve COVID-19 classification accuracy while reducing execution time and storage requirements.

Going forward, this article is organized as follows. Section II reviews some of the important proposed models in COVID-19 identification. Section III presents our proposed methodology. Section IV describes the utilized dataset. Section V explains the conducted preprocessing operations. Section VI introduces Linear Discriminant Analysis (LDA) that we use to extract CXR features. Section VII describes feature expansion as a last preprocessing for applying the deep proposal CNN. Section VIII presents the Fast Hybrid Deep Neural Network (FHDNN) and its summary layers. Section IX evaluates the proposal's performance and compares it with related counterparts. Finally, Section X concludes the paper and identifies some future directions.

II. Related Works

Recently, the DL technique has been adopted for analyzing medical images because of its reliable outcomes [12]. Due to no constraint on the dataset employed, DL is suitable for making predictions based on training data and information. [13]. Currently, the existing methods suffer from some crucial issues, such as being time-consuming and expensive. Therefore, developing DL technology to detect COVID-19 faster and more accurately has become necessary. CXRs and CT scans can help achieve this. However, CXRs are less expensive than CT scans, thus, they are preferable.

To classify COVID-19 cases, Aslan, M. F. et al. [14] suggested a DenseNet-SVM structure and utilized binary classification techniques. While the model achieved an accuracy of 96.29%, the author also used eight models of SVM-based CNN and an average accuracy of 95.21%. Alaska, T. B. et al. [15] utilized the LSTM model and validated their work using a 10-fold cross-validation strategy. The results of the models showed excellent classification performance and achieved an accuracy greater than 84%. Additionally, the recall and AUC scores reached 99.43% and 0.625, respectively. Furthermore, using Holdout validation, the CNN-LSTM achieved better performance, where the accuracy, recall, and AUCs cores went 92.3%, 93.68%, and 0.90 respectively. For detecting and diagnosing human lung disorders, four models were proposed by Ibrahim et al. [16]. The models have been used to classify three cases: COVID-19, lung cancer, and Pneumonia. CNN-VGG19 (GRU with ResNet152V2) achieved an accuracy of 98.05% (96.09%). The VGG-inspired model identified the COVID-19 patients successfully from other cases (regular and pneumonia). For multi-class and binary classification, evaluation metrics accuracy, F1-score, recall, precision, and AUC were 97%, 0.95, 95%, 95% and 0.95, and 98%, 0.97, 97%, 97%, and 0.97, respectively [16].

As mentioned earlier, CXR is one of the essential methods to diagnose pneumonia worldwide [17] since it is a fast, cheap [18], and standard clinical method [19-21]. In [22-24], scholars suggested a DL method-based dataset of CXR images. The authors utilized domain DL within environment-deep CNN. The author proposed that the technique was developed specifically for identifying COVID-19 patients from CXR data. For three classes, an overall accuracy of 96.48% was obtained. However, for four class classifications, the COVDC-Net method achieved accuracy reached 90.22%. The authors used 3,884 (1,414) images for three (four) classes. Using image analysis, early detection of the COVID-19 system was proposed in [25]. Using DL method-based CXR image; the authors calculated the impacts of COVID-19 on people. They utilized five pre-trained CNN-based models: ResNet152, ResNet101, ResNet50, Inception-ResNetV2, and InceptionV3. The highest classification accuracy was determined by ResNet50, reaching 96.1% for the dataset (comprised of two classes).

Similarly, the authors in [26] used transfer learning methods: VGGNet-19, VGG16, VGG19 LeNet-5, MobileNetV2, and the Fusion model. The researchers collected CXR images from various sources (from different classes of chests humans) to detect COVID-19 disease. On the other hand, the study [27] used a pre-trained approach called VGG16-CNN to detect COVID-19 cases using CXR images. While the study achieved an accuracy of 97.50% for multiple classifications, the authors tested the models and found varying proportions, such as an F1-score rate, precision rate, recall rate, and overall accuracy. The authors utilized CNN and pre-trained models for feature extraction and achieved several classification accuracies. Accordingly, the alone use of CNN is insufficient for better identifying COVID-19 cases. Fewer images and no preprocessing steps are the causes of poor performance. One of the major difficulties in employing transfer learning approaches to detect COVID-19 is the time required to apply these techniques to big datasets. Due to the model's need for fine-tuning to alter its parameters, this process takes a long time when dealing with big amounts of CXR picture data. This requires retraining the model on the new data, which can take a significant amount of time, even with powerful computing resources. As shown in the survey [4], many methods served to evaluate pre-trained deep CNN models including AlexNet, VGG-16, GoogleNet, MobileNet-V2, SqueezeNet, ResNet-34, ResNet-50, and Inception-V3 to distinguish COVID-19 from normal cases. Many hyperparameters tuning have been applied to find the best batch size, learning rate, optimizer, and a number of epochs. The obtained results were less than our fast HFDNN with high time consumption. Finally, the highest classification accuracy of recent papers reached 99%, obtained using the InstaCovNet-19 DL method [28]. Based on the studies mentioned above, they suffer from some significant limitations:

- Imbalance classes in the utilized datasets.
- Applied classification images using DL without image preprocessing techniques (such as Linear Discriminant Analysis (LDA)). For example, equalization of the intensity (use Histogram Equalization (HE)), removing noise from the images, resizing it, extracting the best features, then feature expansion.
- The evaluation metrics (i.e., accuracy, precision, recall, and F1-score) must improve.
- DL models typically need a lot of training time.
- More hardware is required for DL model execution.

Therefore, before inserting deep CNN, we applied several image preprocessing techniques to tackle the above drawbacks in this study. In the first step of dataset preprocessing, we convert the images into the gray level. Next, the image intensity is adjusted, and the best features are resized and extracted. Lastly, perform features expansion. This processing may ensure that the data is transformed from two dimensions to one dimension, which will speed up implementation. Furthermore, a unique fast deep learning proposal design can improve COVID-19 classification accuracy in less execution time and storage.

III. Methodology

The COVID-19 outbreak has necessitated rapid action against a massive threat to humans. To diagnose COVID-19, understanding and classifying CXR images are crucial. Deep CNN technologies, which aid radiologists, improve the effectiveness of imaging tools. Our goal in this research is to design a novel CNN model capable of achieving an almost perfect classification accuracy with fast execution time and low storage use. Here, we describe the architecture of our proposed method, namely FHDNN as illustrated in Fig. 1.

The proposed CNN model was built and trained on the COVID-19 four class dataset using the Python programming language and the Keras library. The work was developed using an I7- 4710H Intel processor, 2.50 GHz CPU, 8.0 GB RAM, and an NVIDIA Quadro K2100M graphic card.

After extracting features using LDA and feature expansion, the proposed FHDNN is designed to work with eighteen one-dimensional features. It takes a package of input features, processes them, and classifies them into specific categories. The FHDNN consists of multiple layers that are interlinked and arranged in a way that enhances its performance. When processing each input data set, the eighteen features pass through a series of convolution layers, pooling layers, and fully connected layers (FC) to classify objects with probabilistic values between zero and one.

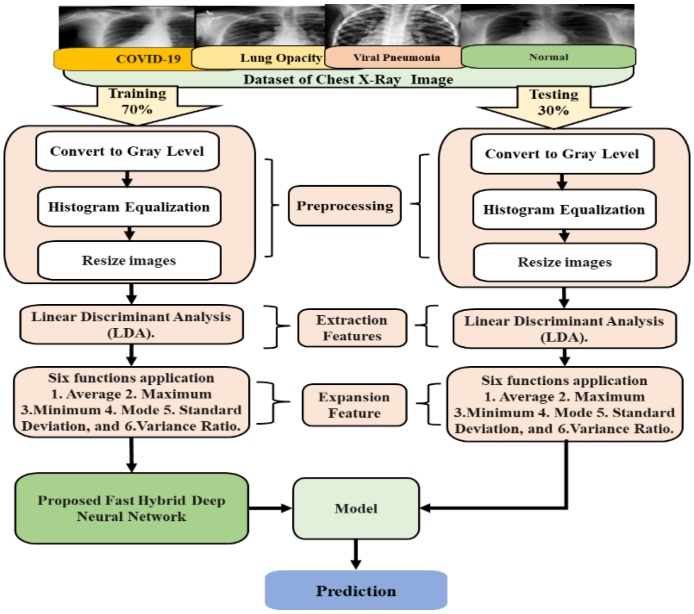

COVID-19
Lung Opacity
Viral Pneumonia
Normal
Dataset of Chest X-Ray Image
Training 70%
Testing 30%
Convert to Gray Level
Histogram Equalization
Resize images
Preprocessing
Convert to Gray Level
Histogram Equalization
Resize images
Linear Discriminant Analysis (LDA).
Extraction Features
Linear Discriminant Analysis (LDA).
Six functions application 1. Average 2. Maximum 3.Minimum 4. Mode 5. Standard Deviation, and 6.Variance Ratio.
Expansion Feature
Six functions application 1. Average 2. Maximum 3.Minimum 4. Mode 5. Standard Deviation, and 6.Variance Ratio.
Proposed Fast Hybrid Deep Neural Network
Model
Prediction

Fig. 1. The flowchart of the proposed system for COVID-19 diagnosis.

To conduct the training process, we divided the dataset into two groups: 70% for training and 30% for testing. Next, we applied the training to adjust the model's parameters and prevent overfitting. This ensures that the model has a high generalization power to accurately classify unseen images during testing.

On the other hand, the images used in our experiments were obtained from publicly available datasets on Kaggle called COVID-19 Radiography Database. The dataset used in our study is already labeled into four classes: COVID, normal, lung-opacity, and pneumonia. This large dataset is designed to assist researchers in evaluating their proposed methods in terms of accuracy and time consumption. A detailed description of this dataset, including the number of samples in each class, is presented in Section IV (Dataset). Accordingly, to solve this problem we are designing FHDNN to predict COVID-19 diagnosis. Before using the model, the dataset passes some steps to reduce storage size, increase implementation speed, and improve classification accuracy:

- Converting the dataset into the gray level and ensuring all images are converted from three channels to 1 channel.
- Adjust image intensity using the HE method.
- Resizing all dataset images for speed execution. The operation extracts the best features using LDA for the best classification.
- Performing feature expansion after preprocessing from CXR images is crucial in improving the accuracy of the algorithms. The expanded features' approaches are Average, Maximum, Minimum, Mode, Standard deviation, and Variance ratio.

Lastly, we implement FHDNN to provide fast and highly accurate diagnostics for COVID-19 diseases from CXR images. In the, following sections, we will detail the different parts of the developed methodology.

IV. Dataset

Although there are numerous COVID-19 infections across the world, a large number of CXR images are freely accessible online. Many authors published a Kaggle database [24] to make the database open in front of researchers worldwide. Usually, the CXR images of COVID-19, viral pneumonia, and lung opacity are available in the database. Here, we have four classes of images in the dataset as follows:

- Class 0: 3616 confirmed COVID-19 cases.
- Class 1: 10,192 normal cases.
- Class 2: 6,012 lung- Opacity cases.
- Class 3: 1345 pneumonia disease cases.

Lastly, Fig. 2 shows the viral pneumonia chest CXR. Across the dataset, the seam size (299*299) images are consistent.

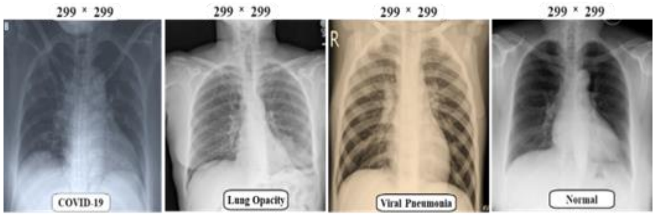


Fig. 2. Sample of CXR images for each class.

V. Dataset Preprocessing

Dealing with enormous amounts of data necessitates the use of the most efficient features for minimizing hardware requirements, such as memory and processing CPU. As mentioned earlier, the experiments' dataset of (21,165) images is classified into four classes. In the beginning, convert the images to one gray level (i.e., one color level). Grayscale is a monochromatic (gray) shade ranging from pure white on the lightest end to pure black on the darkest end. Less information must be provided for each pixel is the reason for differentiating images from any other color ones. Typically, the grayscale intensity is kept as an 8-bit integer, allowing 256 shades of gray ranging from black to white. The soothing impact on the eyes increases the gift of the green color, reduces the contribution of red color, and puts blue between these two [29] (formulated as follows):

$$\text{Grayscale} = \big((0.3 * Red) + (0.59 * Green) + (0.11 * Blue)\big) \quad (1)$$

During this stage, no image of RGB on the dataset has been identified, and the same color levels will all images have. Next, the second stage eliminates noise. CXR images appeared to have a gray level. However, there is an overlap in the image to ensure that it is converted to one group. Fig. 3 shows sample CXR images converted from multi-colors to grayscale.

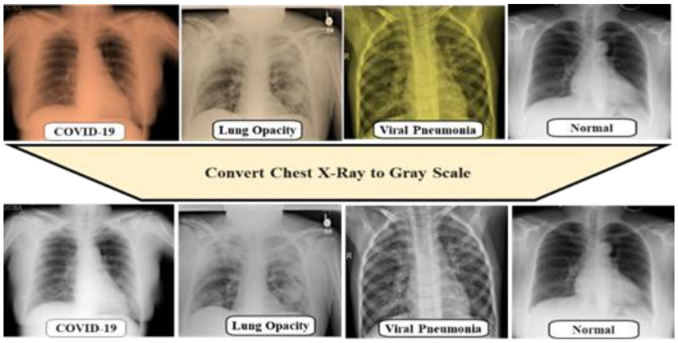


Fig. 3. CXR image samples from different datasets converted to gray level.

CXR images need to distribute lighting over the image by HE adequately. HE is a crucial topic in digital image processing used for contrast adjustment via an image's histogram (i.e., HE is used to enhance contrast). Furthermore, it modifies the image strength to increase opposition [30]. Image improvement aims to improve its quality, highlight desired elements, and make it less intrusive. In an image's histogram, the gray level values are plotted versus the number of pixels at that value. Mathematically, equilibrium can be formulated as follows.

$$\boldsymbol{G(x) = \left(\left(\frac{T(x)}{n}\right) * L\right) - 1} \qquad (2)$$

Where the new image after equalization is denoted as G(x), L is the number of gray levels, n is the number of pixels, and T(x) refers to the cumulative sum of each gray level. Fig. 4 shows an application of HE on CXR images.

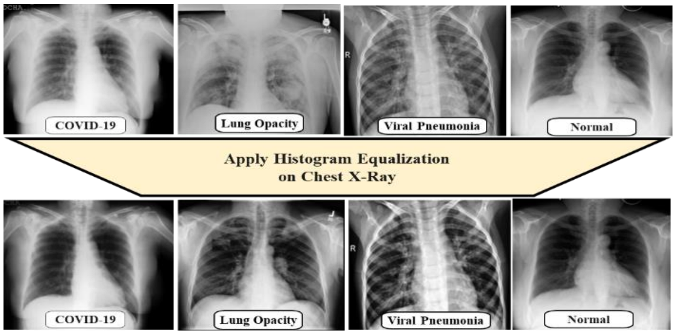


Fig. 4. Applying a Histogram of a CXR image before and after equalization (images are from different datasets).

Image speed and storage size are crucial in image processing, where the image plays an influential role in processing the data in terms of speed. Therefore, we reduced the number of images (of each dataset) to increase processing speed with low storage capacity. At this point, image size must be reduced for faster implementation and less storage space. For giving different resolutions and aspect ratios, resizing images have appeared in many applications. We resized the images from (299 x 299) to (50 x 50). Fig. 5 shows four samples resized from the original 299 x 299 to size 50 x 50. The required formula for resizing is written as follows [31]:

$$\boldsymbol{Ratio = W \,/\, H} \qquad (3)$$

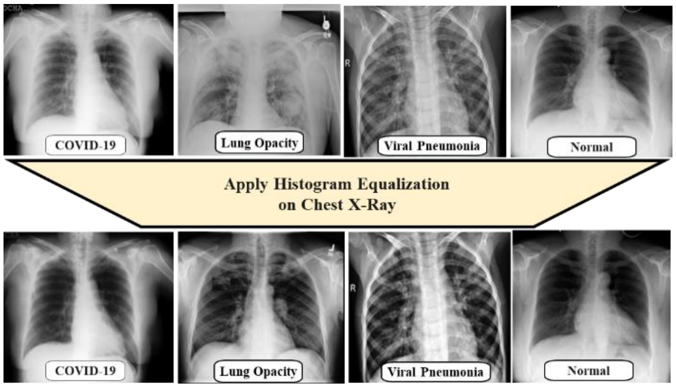


Fig. 5. Sample of resizing four images of the adopted classes.

VI. Features Extraction

The accuracy of the utilized algorithms (ML or DL) can be increased by extracting features (from CXR images) because of an increase in the infection probability. In the literature, since it obtains the best features of image characteristics [32], LDA is the most favored method for extracting discriminative feature pattern classification [33]. LDA starts after completing the earlier stages of image processing. However, when dealing with high-dimensional and small sample-size data, classical LDA suffers from the singularity problem. Under supervision, the LDA reduces dimensionality inside a single classification and separates the classifications so we can select the best image type [34]. The main pros of LDA are that it separates two or more classes and models the group differences in groups by projecting the spaces in a higher dimension into space with a lower extent. Both features create a new axis that minimizes the variance and maximizes the two variables' class distance. Calculate each class's average from the CXR image within the dataset by LDA Algorithm. The mathematical equations used in LDA are similar in action are identical in action [33]. Image preprocessing techniques are applied before model training to achieve better detection accuracy and reduce training duration. Specifically, preprocessing will ensure that the image is transformed from two dimensions to one dimension, consequently speeding up the implementation. Additionally, a propionate fast deep learning proposal design must be utilized for further detection accuracy and higher speed. After implementing LDA, the stages are gray level, histogram equalization, stages are resized, and the best features are extracted.

The development of DL algorithms products is a significant contribution to dataset reprocessing. In the beginning, the images are converted to a single channel with suitable resizing and highlighting desired elements extracting the features. Next, we perform classification between classes and reduce dimensionality within a single category. Accordingly, database preprocessing is essential for enhancing data quality to speed up deep learning with optimal performance.

VII. Feature Expansion

This section explains a feature expansion approach for the one-dimension. As mentioned earlier, LDA is implemented after completing preprocessing on a dataset of CXR images. The output of the LDA got three components from CXR images [35].

The proposed feature expansion is capable of increasing the number of features from three to eighteen. This procedure applies a window that slides through the pattern to expand the high and low elements. The window is a (one × ten) shaped matrix and will be transformed with the input pattern matrix. For data classification, feature expansion can improve classification accuracy by enabling deep CNN application on any dataset from CXR images. Furthermore, feature expansion is helpful if there is a variance in the CXR or a change in the x-ray equipment, such as reducing noise in an image. According to the literature, experiment results demonstrated that the feature expansion method improves the classification performance of CNN algorithms for detecting COVID-19. Finally, the expanded features' approaches (as explained below) are average, maximum, minimum, mode, standard deviation, and variance ratio.

A. Average

The average (or mean) of a set of numbers is calculated by dividing their summations by how many they are. Mathematically, the mean format $\bar{x}$ of a data set is the sum of all data values divided by the count or size n. The following equation can be used to determine the average or mean of a window within the dataset [35].

$$mean = x^{-} = \frac{\sum_{i=1}^{n} x^{i}}{n} \qquad (4)$$

B. Maximum

Finding the maximum value within a window and checking with all data is important. A real-valued function f defined on a domain (X) has a global (or absolute) maximum point at

$$x_i.if\ f(x_i) \geq f(X) for\ all\ x\ in\ X \qquad (5)$$

The value of the function at the top ended the total weight of the process [36].

C. Minimum

The minimum value of a function is determined by finding the lowest value within a window. The value of the process at a minimum point is called the minimum value of the position [36]. Symbolically, this can be written as follows:-

$$x_i.if\ f(x_i) \leq f(X) for\ all\ x\ in\ X \qquad (6)$$

D. Mode

Mode is the number (or value) that appears most frequently in a data set. For data without any repeating values, there is no mode at all. Furthermore, the mode value depends on the given dataset. It is formulated as follows [37].

$$Mode = L + h\ \frac{(\boldsymbol{f_m - f_1})}{(\boldsymbol{f_m - f_1}) + (\boldsymbol{f_m - f_2})} \qquad (7)$$

Where L the lower limit of the modal class is, h is the size of the class interval, and f_m denotes the frequency of the modal class. f_1 (f_2) .This is the frequency of the class preceding (succeeding) the modal classes.

E. Standard Deviation

Standard deviation is a formula used to determine the values of scattered data. It is the deviation of the values or data from an average to a mean. A smaller (larger) standard deviation indicates that the results are nearer to (far from) their average. It is important to mention that no negative value can standard deviation be. It is formulated as follows [38].

$$the\ Standard\ deviation = \sqrt{\frac{\sum(\boldsymbol{X}-\boldsymbol{\mathcal{X}}^{-})^2}{\boldsymbol{n}-\boldsymbol{1}}} \quad (8)$$

Where X is each value in the sample t, $\mathcal{X}^{-}$ refers to sample means, and n is the number of values in the sample.

F. Variance Ratio

Statistically, variance (symbolized by σ^2) refers to the spread between numbers in a data set. In other words, variance calculates how far every number is from the average (i.e., from other numbers in the group). Both traders and analysts use it to find volatility through the dataset. Variance is calculated as follows [39]:

$$\boldsymbol{\sigma^2} = \frac{\sum_{i=1}^{n}(\,x_i - x^{-}\,)^2}{N} \quad (9)$$

Where $\mathcal{X}^{-}$ refers to the mean of all values in the dataset, $\boldsymbol{x_i}$ is each value in the dataset, and $\boldsymbol{n}$ is the number of values in the dataset.

VIII. Fast Hybrid Deep Neural Network

A. Preliminaries

In the state-of-the-art, researchers and authors who use transfer learning for COVID-19 detection need to consider the trade-offs between accuracy and computational resources, as well as the time required to fine-tune the model on large datasets. It is important to carefully balance these factors to ensure that transfer learning can be applied effectively in the state of the art for COVID-19 detection. These trade-offs are not easy to ensure. Although the ensemble model can enhance the efficiency of each model separately, its implementation requires fine-tuning the aforementioned deep models which is a very challenging task since it requires powerful machines.

Alternative methods have been proposed to extract deep features and feed them into an SVM classifier without utilizing transfer learning, as demonstrated in [40]. However, this approach is less time-consuming, but it lacks a high generalization power because the extracted features are tailored to the ImageNet dataset initially used to train the model. Consequently, the SVM classifier may not achieve a very high accuracy level.

To solve this problem, the proposed HFDNN aims to extract features using LDA and apply some feature engineering to feed a DNN for detection. This strategy will avoid fine-tuning existing models to large datasets, and considerably speed up the training process. Before the dataset was enlarged, it underwent preprocessing, and the best features were extracted using LDA. Next, we altered it from two dimensions to one dimension to shorten training time and hardware needs. By using this method, existing models won't need to be adjusted for large datasets, and the training process will be greatly accelerated. In this section, we explain our proposal, the fast hybrid deep Convolutional Neural Network model to detect COVID-19 from CXR images.

B. Modeling

A rapid diagnosis of COVID-19 is necessary to treat and control the disease. CXR is an essential tool that can be used to find COVID-19. Therefore, this study aims to improve classification accuracy and reduce training time and hardware requirements. Furthermore, it should accurately identify COVID-19 cases so doctors can treat patients appropriately. Therefore, we proposed to utilize a novel deep-learning method to predict COVID-19 using CXR images based on deep CNN. The proposed model implements the preprocessing on the dataset from CXR images in some steps before applying the proposed deep CNN for the COVID-19 classification. As explained later, our model's (FHDNN) objectives are the automatic identification of four classes: 1) normal, 2) lung opacity, 3) viral pneumonia, and 4) COVID-19. As it is named, FHDNN is a quick deep CNN used to classify CXR images for detected COVID-19. The proposed FHDNN deals with features (eighteen features) that must be one-dimensional. FHDNN takes an input package of features, processes it, and classifies it under specific categories. It consists of several layers intertwined with each other and arranged in a way that makes it more highly. For each input data set, eighteen features pass through a series of convolution layers with filters, Pooling, and Fully Connected layers (FC) to classify objects with probabilistic values between (zero and one). The architecture of FHDNN for COVID-19 detection is depicted in Fig. 6. FHDNN comprises twenty-seven layers categorized as follows:

- Eight Conv1D layers: Eight convolutional layers are used for feature extraction of type 1D.
- Seven MaxPooling 1D layers.
- Three Dense layers: They are divided into 1) One layer representing a fully connected layer in a convolutional neural network; and 2) two dense layers are placed before the Flatten to obtain robust features closer to them.
- Eight LeakyReLU layers: The Activation function is LeakyReLU.
- One Flatten layer: The full connection step in CNNs.

C. Layers of FHDNN

As mentioned earlier, the FHDNN is expected to achieve accurate classification results due to dealing with one-dimension CXR images (after preprocessing datasets). Table I lists and explains the FHDNN layers in terms of Filters, Strides, and Padding. We listed detailed explanations and clarification of each layer with the activation and setting of all parameters in each layer. The difference is the structure of the input data and how the filter (also called a feature detector or a convolution kernel) moves across the data. It is worth mentioning that the attained FHDNN model layers and their parameters in this network are one-dimensional (will be explained and shown according to the order of layers).

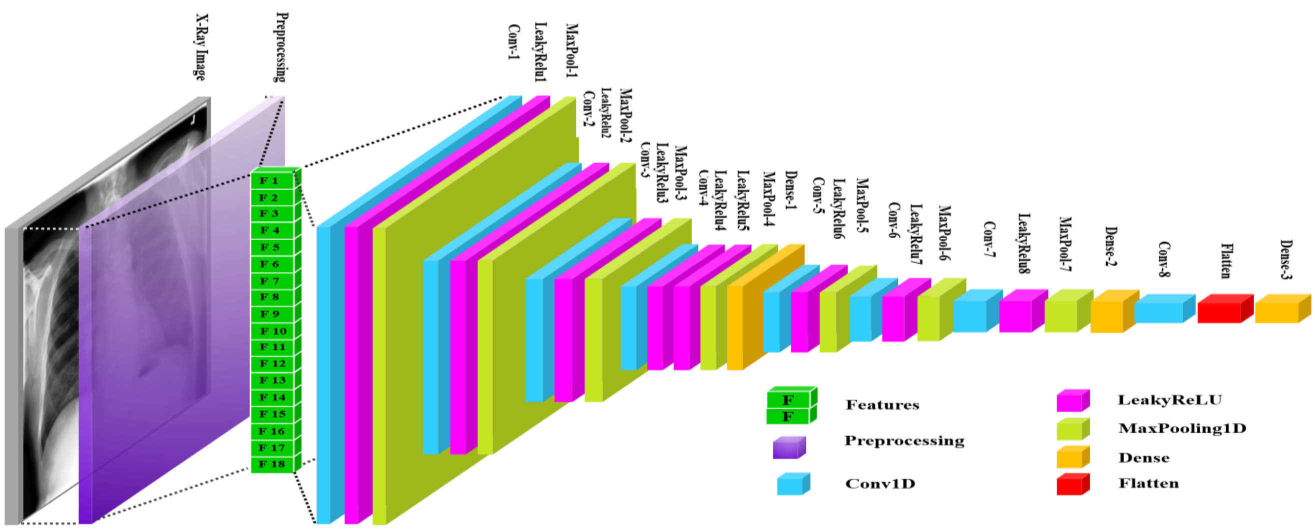


Fig. 6. FHDNN architecture.

TABLE I. INFORMATION OF FHDNN LAYERS

No	Layer	Filter	Strides	Padding
1.	Max pooling 1D-1	16	1	-
2.	LeakyReLU-1	-	-	-
3.	Convolution 1D -1	16	1	Valid
4.	Max pooling 1D-2	32	1	-
5.	LeakyReLU-2	-	-	-
6.	Convolution 1D-2	32	1	Valid
7.	Max pooling 1D-3	32	1	-
8.	LeakyReLU-3	-	-	-
9.	Convolution 1D-3	32	1	Valid
10.	Max pooling 1D-4	128	1	
11.	LeakyReLU-4	-	-	-
12.	Convolution 1D-4	128	1	Valid
13.	Dense 1D-1	128	-	-
14.	Max pooling 1D-5	256	1	-
15.	LeakyReLU-5	-	-	-
16.	Convolution 1D-5	256	1	Valid
17.	Max pooling 1D-6	512	1	-
18.	LeakyReLU-6	-	-	-
19.	Convolution 1D-6	512	1	Valid
20.	Dense 1D-2	1024	-	-
21.	Max pooling 1D-7	1024	1	-
22.	LeakyReLU-7	-	-	-
23.	Convolution 1D-7	1024	1	Valid
24.	LeakyReLU-8	-	-	-
25.	Convolution 1D-8	50	1	-
26.	Dense 1D-3	4	-	-
27.	Flatten 1D-1	50	-	-

The utilized layers are arranged as follows:

- Eight convolutional layers in the proposed lightweight CNN model have sequential filters (16, 32, 32, 128, 256, 512, 1024, and 50). This layer applies kernels that slide through the pattern to extract low- and high-level features [41]. The kernel is a (one × three) shaped matrix to be transformed with the input pattern matrix. The stride is (one × one) that parameter is the number of steps tuned for shifting over the input matrix. During the design process, the padding was omitted from the border elements. The padding is designated as 'VALID' in the remaining convolutional layers and does not consider the boundary components.
- Eight LeakyReLU layers are set to (alpha=0.3). The activation function LeakyReLU is used as an activation function after every convolutional layer. The LeakyReLU layers were used twice after the convolutional layer with filter size 128. Next, the third layer after the LeakyReLU layer is the pooling layer.
- The pooling layer is usually applied to the created feature maps. It aims to reduce the number of feature maps and network parameters by applying corresponding mathematical computation. A (one × one) max pool is used seven times after each convolutional block.
- As we propose a lightweight FHDNN model, three dense layers have been used. The dense layer performs a matrix-vector multiplication, and the values used in the matrix are parameters that can be trained and updated with the help of backpropagation. The output generated by the dense layer is an 'n-dimensional vector. The activation function (linear) uses the two dense layers by setting the vector sequential as follows (128, 1024). The activation function (softmax, 4) is the last dense layer to classify four classes.

- Lastly, to flatten the matrix into a vector, a flattening layer was used once. Additionally, the most and last crucial layer of FHDNN is fully connected. This layer functions like a multilayer perceptron. Rectified linear unit (LeakyReLU) activation function is commonly used on a fully connected layer. In contrast, the softmax activation function is used to predict output images in the last layer of a fully connected layer. This step will look at the softmax regression used for multi-class classification problems.

Now, we can explain the experimental parameters obtained from each layer at the proposed system procedures implementation on the CXR Images for diagnosis of COVID-19. As listed in Table II, the FHDNN model includes a mix of convolution neural network layers. Afterward, the parameters of every layer in the FHDNN are presented to show the success of the submitted diagnosis of COVID-19 by using CXR images. The total number of parameters is 3,303,494, and the batch size equals 64. Finally, Table II also summarizes the proper trainable parameters for each layer of the FHDNN model.

TABLE II. SUMMARY OF THE FHDNN MODEL

Layer (type)	Output Shape	Number of parameters
Conv1d (Conv1D)	(None, 16, 16)	64
Leaky_re_lu (LeakyReLU)	(None, 16, 16)	0
Max_pooling1d (MaxPooling1D)	(None, 16, 16)	0
Conv1d_1 (Conv1D)	(None, 14, 32)	1568
Leaky_re_lu_1 (LeakyReLU)	(None, 14, 32)	0
Max_pooling1d_1 (MaxPooling1D)	(None, 14, 32)	0
Conv1d_2 (Conv1D)	(None, 12, 32)	3104
Leaky_re_lu_2 (LeakyReLU)	(None, 12, 32)	0
Max_pooling1d_2 (MaxPooling1D)	(None, 12, 32)	0
Conv1d_3 (Conv1D)	(None, 10, 128)	12416
Leaky_re_lu_3 (LeakyReLU)	(None, 10, 128)	0
Leaky_re_lu_3 (LeakyReLU)	(None, 10, 128)	0
Max_pooling1d_3 (MaxPooling1D)	(None, 10, 128)	0
Dense (Dense)	(None, 10, 128)	16512
Conv1d_4 (Conv1D)	(None, 8, 256)	98560
Leaky_re_lu_5 (LeakyReLU)	(None, 8, 256)	0
Max_pooling1d_4 (MaxPooling1D)	(None, 8, 256)	0
Conv1d_5 (Conv1D)	(None, 6, 512)	393728
Leaky_re_lu_6 (LeakyReLU)	(None, 6, 512)	0
Max_pooling1d_5 (MaxPooling1D)	(None, 6, 512)	0
Conv1d_6 (Conv1D)	(None, 4, 1024)	1573888
Leaky_re_lu_7 (LeakyReLU)	(None, 4, 1024)	0
Max_pooling1d_6 (MaxPooling1D)	(None, 4, 1024)	0
Dense_1 (Dense)	(None, 4, 1024)	1049600
Conv1d_7 (Conv1D)	(None, 2, 50)	153650
Flatten (Flatten)	(None, 100)	0
Dense_2 (Dense)	(None, 4)	404

IX. PERFORMANCE EVALUATION

A. Evaluation Metrics

After the model is trained, we evaluate its performance using the images of the testing dataset. In this work, we adopted the following evaluation matrices: Accuracy, Precision, Recall, and F1-Score. Before delving into the results, we need to explain these metrics. After the model's movement, the FHDNN's performance on the testing dataset was evaluated. It is important to mention that this study did not use cross-validation. Therefore, the holdout validation technique has been used instead because the dataset has enough samples for testing and training.

A confusion matrix table is usually used to explain the classifier's performance (i.e., model classification) on a data set in which its actual cases are known. The confusion matrix comprises four parameters (TP, FP, TN, and FN). While TP and TN refer to the patient being classified correctly, FP and FN mean it misdiagnosed the case (i.e., classified incorrectly). This section briefly defines the evaluation metrics in AD classification. Following, we explain the evaluation metrics.

1) Accuracy: It is widely used to measure the model's accuracy validation (training) or classification. As mentioned above, via a confusion matrix, the number of TP, TN, FP, and FN are calculated, which further helps check the proposed model's efficacy. Accordingly, as formulated in Eq. (1), the accuracy is the number of correct classifications (i.e., TP and TN) divided by the total number of classifications.

$$\boldsymbol{Accuracy} \;= \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{10}$$

2) Precision: It calculates how many cases classified as positive (infection) by the model was supposed to be indicated as positive. It is defined as the ratio of classified TPs to the total number of classified positives (correctly (TP) and incorrectly (FP). Thus, it is formulated as follows.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{11}$$

3) Recall: The third measure of the confusion matrix is Recall (or sensitivity). The recall is the number of positive cases foretold adequately by the model is estimated by the recall, sensitivity, or "actual positive rate. Recall calculates how well the model has classified the positive examples. It is defined as the ratio of classified TPs to the real positives and is written as follows.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{12}$$

4) F1-score: It is the harmonic mean of precision and recall. Perfect precision and Recall can achieve the highest F-score (i.e., close to 1).

$$\boldsymbol{F1-score} \;= \boldsymbol{2} * \frac{(\boldsymbol{Recall} * \boldsymbol{Precision})}{\boldsymbol{Recall} + \boldsymbol{Precision}} \tag{13}$$

Preparing the dataset preprocessing is one of the prime aspects of training and testing deep learning for the desired results. The CXR images have passed through multi-steps in preprocessing before entering the FHDNN layers, providing an accurate effect in the classification procedure.

B. Model Performance

The training time required to train the FHDNN algorithm is important to look at the accuracy obtained from the classification algorithm. For instance, 100 epochs are recommended in FHDNN training. When using FHDNN, the training time is 14 sec, which differs from the attained feature extraction algorithms. The obtained error rate (α) from training the classifier is 0.0001. Table III lists the results of applying FHDNN with the four classes (COVID-19 class, normal, lung-opacity, and viral Pneumonia, based on the performance confusion matrix. As listed in Table III, the performance metrics of the proposed modes achieved the following results. Firstly, the accuracy of the model reached 99.9%. Secondly, the model's foretelling of positive samples is assessed by precision. Meanwhile, the models' level of precision showed that FHDNN has the most fantastic weighted average value of 99.9% for the four classes. Thirdly, the actual positive rate (sensitivity) measures the number of positive cases correctly foretold by the model. The Sensitivity of the model reached 99.9%. Lastly, the F1-score achieved an almost perfect value, 99.9%.

TABLE III. PERFORMANCE OF THE PROPOSED MODEL

Type	Accuracy	Precision	Recall	F-score
COVID-19	99.9%	99.9%	99.9%	99.9%
Normal	99.9%	99.9%	99.9%	99.9%
Lung- Opacity	99.9%	99.9%	99.9%	99.9%
Viral Pneumonia	99.9%	99.9%	99.9%	99.9%

C. FHDNN vs. Counterparts

Using CXR images, many DL models-based COVID-19 classification systems have been utilized widely. For example, AlexNet, ResNet, VGG16, VGG19, and GoogleNet. To demonstrate the superiority of our work, we compare it with the achievements of some CNN-based models in the literature. The comparisons are based on: classification classes, accuracy, precision, recall, F-score, and training time. Additionally, we compared our proposal with 12 studies that utilized CNN models.

Before comparing the models with ours, it is important to mention that the convolutional layers for MobileNetV2, DenseNet-121, ResNet50, VGG19, and ChestX-Ra6 are 53, 121, 50, 16, and 6, respectively. Their training times are 6457.20, 7571.84, 7018.10, 8215.94, and 6150.62 sec, respectively (see Fig. 7). It is important to mention that different computer configurations could provide different results. Although some studies calculated the training time, they consumed a very long training time. Specifically, the models completed the training in thousands of seconds (i.e., hours). On the other hand, our model needed only 13 sec. Some examples can be found in references [27] [42-52]. Except for the study [50], most models performed on 3 and 2-class classifications. However, our model is better than [50] in all evaluation metrics (accuracy, precision, recall, and F-score). Furthermore, our training time is significantly less than it. Similar to our work, the models [42] classified four classes; however, our model outperformed in performance evaluation metrics and training time. Accordingly, our model outperformed the counterparts.

When it comes to detecting COVID-19 using transfer learning techniques, one of the significant challenges is the amount of time it takes to apply these methods to large datasets. This process is particularly time-consuming when dealing with large amounts of data of CXR images, as the model needs to be fine-tuned to adjust its parameters. This requires retraining the model on the new data, which can take a significant amount of time, even with powerful computing resources. As a result, researchers and practitioners who use transfer learning for COVID-19 detection need to consider the trade-offs between accuracy and computational resources, as well as the time required to fine-tune the model on large datasets. It is important to carefully balance these factors to ensure that transfer learning can be applied effectively in the state of the art for COVID-19 detection. These trade-offs are not easy to ensure. To solve this problem, the proposed HFDNN aims to automate COVID-19 disease classification to maintain high accuracy and reduce execution time. The dataset was first preprocessed before being extended and the best features were retrieved using LDA. Next, in order to decrease training time and hardware requirements, we changed it from two dimensions to one dimension. This strategy will avoid fine-tuning existing models to large datasets, and considerably speed up the training process as shown in Fig. 7.

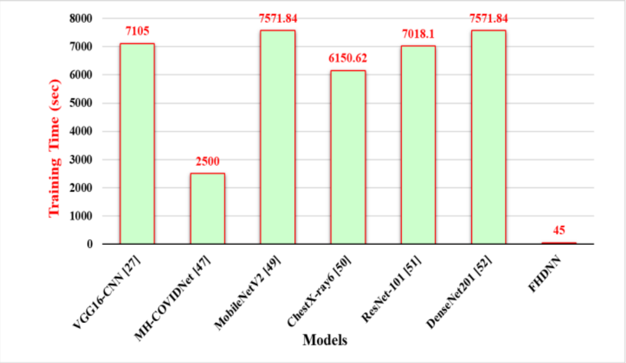


Fig. 7. Training time of each model (sec).

D. Performance matrix and comparison with related works

In this subsection, we explain the performance matrix and comparison with related works. Though the model is complex in architecture, it has twenty-seven convolutional layers. Still, the model classification performance is better than the other models. One of the main concerns of this study is designing a model that provides higher classification accuracy and reduces the training time with the lowest requirement hardware for large or small amounts of data. For that reason, a lightweight FHDNN model has been designed to measure whether it performed well for these criteria in the case of a multiclass environment.

1) Accuracy: Fig. 8 shows the tested obtained accuracy using some of the models of DL and our work. It is clear from the figure that our model achieved the highest classification accuracy, which has an accuracy of 99.9%.

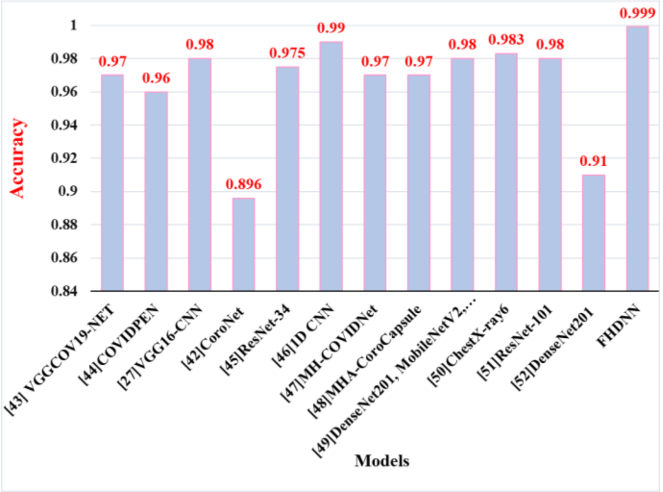


Fig. 8. The accuracy of the models.

2) *Precision:* The second is the precision that assesses the model precision in foretelling positive samples. Meanwhile, the six models' level of precision showed that FHDNN has the most excellent weighted average value (99.9%) for four classes than other models. Fig. 9 shows that the precision of FHDNN has the most excellent weighted average value (0.999) for four classes than other models of CNN.

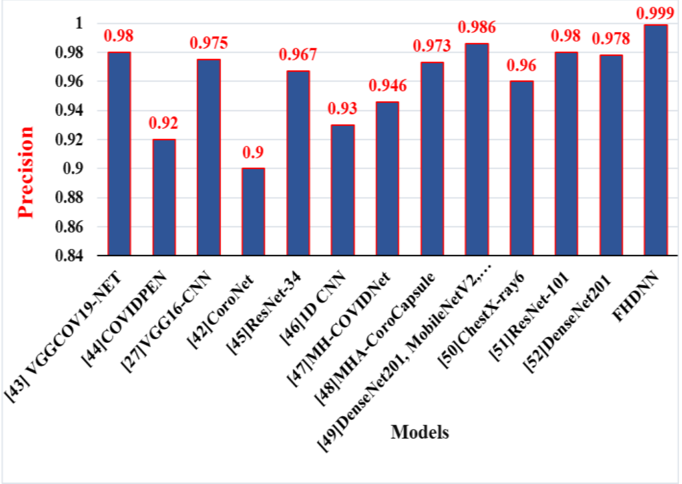


Fig. 9. The precision of the models.

3) *Recall:* As shown in Fig. 10, the results show FHDNN method outperforms the close models of CNN in a variety of weighted average sensitivities. The FHDNN determined weighted average has (0.999) of sensitivity, but the VGG19 has (0, 98), GoogleNet has (0, 92), DenseNet201 has (0.92), etc.

4) *F1-score:* As mentioned earlier, the F1-score is the harmonic mean of precision and recall. As shown in Fig. 11, the F1- score value of the different models also shows that the FHDNN has a value (of 99.9%), the highest compared to the related work comparison models. The results state that DL methods outperform the comparative models in classification F1-score (percentage).

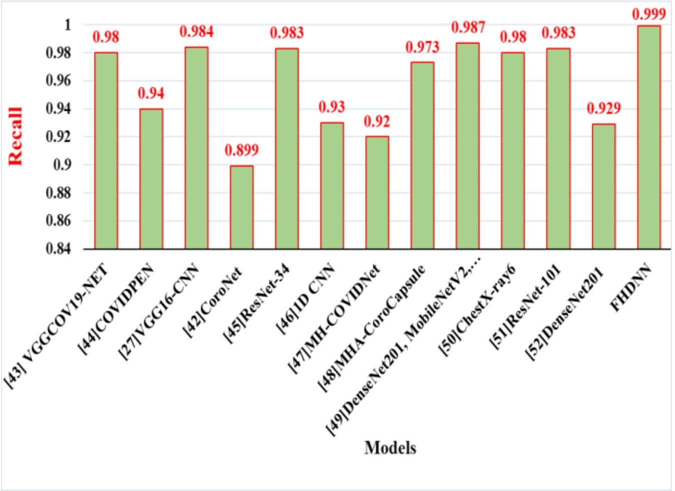


Fig. 10. Recall (sensitivity) of the models.

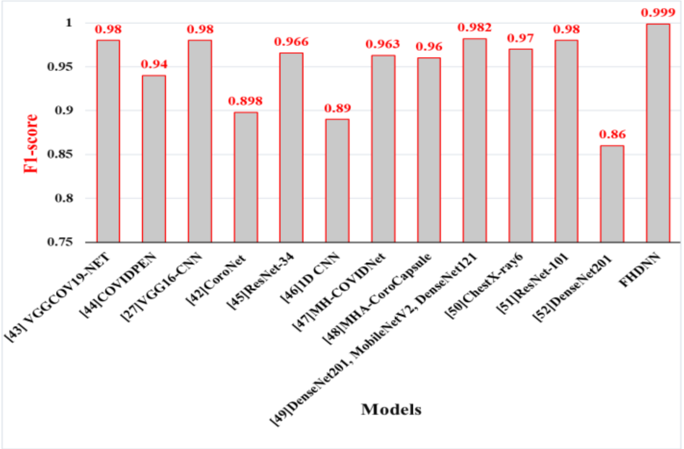


Fig. 11. F1-score of the models.

This article presented a DL method for fast detection and classification of COVID-19 disease. The preprocessing stage has been performed on the dataset to achieve better speed and accuracy. The comparison demonstrated that we achieved better results than some relevant previous works, knowing that it dealt with an extensive and unbalanced database. Furthermore, different DL models were applied to the same dataset for comparison, but they got lower results in all measures than in our proposal, FHDNN. The result states that FHDNN possesses the highest important parameters that measure each model's performance compared with other CNN models. Meanwhile, our model's accuracy, precision, and sensitivity show that FHDNN has 99.9% of the most outstanding value. The F1-score value of the DL models also states that the FHDNN has 99.9% and stays the most excellent weighted average compared to the CNN comparison models with less than FHDNN a value.

X. Conclusion

COVID-19 is one of the cases in which doctors and medical physicians face difficulties in correctly identifying these diseases. CXR imaging has several advantages over other imaging and detection techniques. If trained correctly, a machine can classify CXR images more accurately than a human. Numerous results have reported COVID-19 detection from a smaller or large set of CXR images. However, the literature did not say the effect of image enhancement and lung x-ray preprocessing of a large or small dataset of CXR images.

Therefore, this article proposed a novel lightweight new deep CNN (FHDNN) model to detect COVID-19 diseases using CXR images. To calculate our model's performance for detecting COVID-19 diseases, we trained our model using 21,165 CXR images after completing all preprocessing stages. These are classes unbalanced using various preprocessing techniques such as gray image level, resizing, and then enhancement by histogram equalization. Furthermore, we proposed extracting the best features by LDA, then the feature expansion approach and converting from two-Dimension to one-Dimension reduce time training. The proposed FHDNN model for multiple classifications (normal, pneumonia, lung-opacity, and COVID-19) achieved 99.9% in all performance metrics (accuracy, precision, recall, and F1-score). These results demonstrated the superiority of our model over the proposed CNN models (previous works). Our FHDNN model has achieved excellent performance in diagnosing COVID-19 by using CXR Images, which can help medical physicians diagnose these diseases correctly. In the future, we intend to apply the model and improve it for diagnosing other diseases by CXR images.

REFERENCES

[1] T. Islam, S. Absar, S. A. I. Nasif, and S. S. Mridul, "Deep Neural Network models for diagnosis of COVID-19 Respiratory diseases by analyzing CT-Scans and Explain-ability using trained models," in 2022 International Conference on Inventive Computation Technologies (ICICT), 2022: IEEE, pp. 16-23.

[2] C. A. Larabell and K. A. J. C. o. i. s. b. Nugent, "Imaging cellular architecture with X-rays," vol. 20, no. 5, pp. 623-631, 2010.

[3] W. Hariri and A. J. S. c. Narin, "Deep neural networks for COVID-19 detection and diagnosis using images and acoustic-based techniques: a recent review," vol. 25, no. 24, pp. 15345-15362, 2021.

[4] H. S. Alghamdi, G. Amoudi, S. Elhag, K. Saeedi, and J. J. I. A. Nasser, "Deep learning approaches for detecting COVID-19 from chest X-ray images: A survey," vol. 9, pp. 20235-20254, 2021.

[5] M. Agarwal et al., "A novel block imaging technique using nine artificial intelligence models for COVID-19 disease classification, characterization, and severity measurement in lung computed tomography scans on an Italian cohort," vol. 45, no. 3, pp. 1-30, 2021.

[6] A. M. Ayalew, A. O. Salau, B. T. Abeje, B. J. B. S. P. Enyew, and Control, "Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients," vol. 74, p. 103530, 2022.

[7] R. Sadik et al., "Covid-19 pandemic: a comparative prediction using machine learning," vol. 1, no. 1, pp. 1-16, 2020.

[8] M. S. Anggreainy, F. H. Kansil, and A. M. Illyasu, "Comparative Performance Analysis of Machine Learning Classifier for COVID-19 Detection using Chest X-Ray Images," in 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), 2022: IEEE, pp. 337-341.

[9] H. A. Ali, W. Hariri, N. S. Zghal, and D. B. Aissa, "A Comparison of Machine Learning Methods for best Accuracy COVID-19 Diagnosis Using Chest X-Ray Images," in 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2022: IEEE, pp. 349-355.

[10] S. Akbar, H. Tariq, M. Fahad, G. Ahmed, and H. J. J. E. Syed, "Contemporary Study on Deep Neural Networks to Diagnose COVID-19 Using Digital Posteroanterior X-ray Images," vol. 11, no. 19, p. 3113, 2022.

[11] M. Ahsan, M. A. Based, J. Haider, and M. J. S. Kowalski, "COVID-19 detection from chest X-ray images using feature fusion and deep learning," vol. 21, no. 4, p. 1480, 2021.

[12] K. Suzuki, "Overview of deep learning in medical imaging," Radiological physics and technology, vol. 10, no. 3, pp. 257-273, 2017.

[13] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," IEEE access, vol. 2, pp. 514-525, 2014.

[14] M. F. Aslan, K. Sabanci, A. Durdu, and M. F. Unlersen, "COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization," Computers in Biology and Medicine, p. 105244, 2022.

[15] T. B. Alakus and I. Turkoglu, "Comparison of deep learning approaches to predict COVID-19 infection," Chaos, Solitons & Fractals, vol. 140, p. 110120, 2020.

[16] G. S. George, P. R. Mishra, P. Sinha, and M. R. Prusty, "COVID-19 detection on chest X-ray images using Homomorphic Transformation and VGG inspired deep convolutional neural network," Biocybernetics and Biomedical Engineering, 2022.

[17] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. Rodrigues, "Identifying pneumonia in chest X-rays: A deep learning approach," Measurement, vol. 145, pp. 511-518, 2019.

[18] B. Antin, J. Kravitz, and E. Martayan, "Detecting pneumonia in chest X-Rays with supervised learning," Semanticscholar. org, 2017.

[19] N. N. Das, N. Kumar, M. Kaur, V. Kumar, and D. Singh, "Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays," Irbm, 2020.

[20] E. Ayan and H. M. Ünver, "Diagnosis of pneumonia from chest X-ray images using deep learning," in 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), 2019: Ieee, pp. 1-5.

[21] G. Gaál, B. Maga, and A. Lukács, "Attention u-net based adversarial architectures for chest x-ray lung segmentation," arXiv preprint arXiv:2003.10304, 2020.

[22] A. Sharma, K. Singh, D. J. B. S. P. Koundal, and Control, "A novel fusion based convolutional neural network approach for classification of COVID-19 from chest X-ray images," vol. 77, p. 103778, 2022.

[23] M. E. Chowdhury et al., "Can AI help in screening viral and COVID-19 pneumonia?," vol. 8, pp. 132665-132676, 2020.

[24] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[25] A. Narin, C. Kaya, Z. J. P. A. Pamuk, and Applications, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," vol. 24, no. 3, pp. 1207-1220, 2021.

[26] W. A. Hamwi and M. M. J. I. i. M. U. Almustafa, "Development and integration of VGG and dense transfer-learning systems supported with diverse lung images for discovery of the Coronavirus identity," p. 101004, 2022.

[27] A. A. Ramadhan and M. J. A. S. Baykara, "A Novel Approach to Detect COVID-19: Enhanced Deep Learning Models with Convolutional Neural Networks," vol. 12, no. 18, p. 9325, 2022.

[28] S. Bharati, P. Podder, M. Mondal, and V. Prasath, "Medical imaging with deep learning for COVID-19 diagnosis: a comprehensive review," arXiv preprint arXiv:2107.09602, 2021.

[29] K. Padmavathi, K. J. I. J. o. S. Thangadurai, and Technology, "Implementation of RGB and grayscale images in plant leaves disease detection–comparative study," vol. 9, no. 6, pp. 1-6, 2016.

[30] C. Nithyananda and A. Ramachandra, "Review on histogram equalization based image enhancement techniques," in 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016: IEEE, pp. 2512-2517.

[31] W. Siddique, I. V. Shevchuk, L. El-Gabry, N. B. Hushmandi, T. H. J. H. Fransson, and M. Transfer, "On flow structure, heat transfer and pressure drop in varying aspect ratio two-pass rectangular channel with ribs at 45," vol. 49, no. 5, pp. 679-694, 2013.

[32] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," IEEE Transactions on Neural Networks, vol. 22, no. 7, pp. 1119-1132, 2011.

[33] M. Anggo and L. Arapu, “Face recognition using fisherface method,” in Journal of Physics: Conference Series, 2018, vol. 1028, no. 1: IOP Publishing, p. 012119.

[34] T. V. Bandos, L. Bruzzone, G. J. I. T. o. G. Camps-Valls, and R. Sensing, “Classification of hyperspectral images with regularized linear discriminant analysis,” vol. 47, no. 3, pp. 862-873, 2009.

[35] L. J. T. A. S. Zhang, “Sample mean and sample variance: Their covariance and their (in) dependence,” vol. 61, no. 2, pp. 159-160, 2007.

[36] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in Traces and emergence of nonlinear programming: Springer, 2014, pp. 247-258.

[37] N. M. Kopelman, J. Mayzel, M. Jakobsson, N. A. Rosenberg, and I. J. M. e. r. Mayrose, “Clumpak: a program for identifying clustering modes and packaging population structure inferences across K,” vol. 15, no. 5, pp. 1179-1191, 2015.

[38] C. Leys, C. Ley, O. Klein, P. Bernard, and L. J. J. o. e. s. p. Licata, “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median,” vol. 49, no. 4, pp. 764-766, 2013.

[39] T. C. Urdan, Statistics in plain English. Routledge, 2011.

[40] Khamparia, Aditya, et al. "An internet of health things - driven deep learning framework for detection and classification of skin cancer using transfer learning." Transactions on Emerging Telecommunications Technologies 32.7 (2021): e3963.

[41] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436-444, 2015.

[42] A. I. Khan, J. L. Shah, and M. M. Bhat, “CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images,” Computer methods and programs in biomedicine, vol. 196, p. 105581, 2020.

[43] A. J. N. C. Karacı and Applications, “VGGCOV19-NET: automatic detection of COVID-19 cases from X-ray images using modified VGG19 CNN architecture and YOLO algorithm,” vol. 34, no. 10, pp. 8253-8274, 2022.

[44] A. K. Jaiswal, P. Tiwari, V. K. Rathi, J. Qian, H. M. Pandey, and V. H. C. J. M. Albuquerque, “Covidpen: A novel covid-19 detection model using chest x-rays and ct scans,” 2020.

[45] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, R. B. J. B. S. P. Pachori, and Control, “Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study,” vol. 64, p. 102365, 2021.

[46] E. A. Abbood and T. A. Al-Assadi, “GLCMs Based multi-inputs 1D CNN Deep Learning Neural Network for COVID-19 Texture Feature Extraction and Classification,” Karbala International Journal of Modern Science, vol. 8, no. 1, pp. 28-39, 2022.

[47] M. J. B. S. P. Canayaz and Control, “MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images,” vol. 64, p. 102257, 2021.

[48] F. Li, X. Lu, and J. J. I. T. o. M. I. Yuan, “MHA-CoroCapsule: Multi-Head Attention Routing-Based Capsule Network for COVID-19 Chest X-Ray Image Classification,” vol. 41, no. 5, pp. 1208-1218, 2021.

[49] V. C. Shinde and P. S. Kulkarni, “Automatic COVID-19 Detection from Chest X-Rays using Deep Learning Techniques,” in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022: IEEE, pp. 587-594.

[50] M. Nahiduzzaman, M. R. Islam, and R. J. E. S. w. A. Hassan, “ChestX-Ray6: Prediction of multiple diseases including COVID-19 from chest X-ray images using convolutional neural network,” vol. 211, p. 118576, 2023.

[51] A. Gupta, S. Gupta, and R. Katarya, “InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray,” Applied Soft Computing, vol. 99, p. 106859, 2021.

[52] G. Dhiman, V. Chang, K. Kant Singh, and A. Shankar, “Adopt: automatic deep learning and optimization-based approach for detection of novel coronavirus covid-19 disease using x-ray images,” Journal of biomolecular structure and dynamics, vol. 40, no. 13, pp. 5836-5847, 2022.

Digital Image Encryption using Composition of RaMSH-1 Map Transposition and Logistic Map Keystream Substitution

Rama Dian Syah[1], Sarifuddin Madenda[2], Ruddy J. Suhatril[3], Suryadi Harmanto[4]
Department of Computer Science, Universitas Gunadarma, Depok, Indonesia[1, 3]
Department of Information Technology, Universitas Gunadarma, Depok, Indonesia[2, 4]

***Abstract*—Digital communication of multimedia data (text, signal/audio, image, and video) through the internet network has an important role in the era of industrial revolution 4.0 and society 5.0. However, the easiness of exchanging personal and confidential digital information/data has a high risk of being hijacked by irresponsible people. The development of reliable and robust data encryption methods is a solution to this risk. This paper suggests combining modified Henon Map transposition encryption and Logistic Map keystream substitution encryption to create a novel data encryption method (called RaMSH-1). The proposed algorithm simultaneously transposes data positions and substitute data values randomly, as well as having encryption key combination or key space of 1.05 × 10^{670}. A few images with various sizes and variations of color features, object shapes, and textures have been tested. Based on the results of the analysis of randomness, key sensitivity, and visual, it is evident that the proposed encryption algorithm is resistant to differential attack, entropy attack and brute force attack.**

Keywords—Encryption; Decryption; Digital Image; RaMSH-1 Map Transposition; Logistic Map Substitution

I. Introduction

Data security is an important issue in technological advances. Private data requires data security to protect the data from hacker or cracker [1]. Cryptography is a science related to data security. Cryptography is used to transform plaintext into ciphertext to make the meaning of message difficult to read [2]. Data can be in the form of digital images that can have many meanings. Data security methods such as encryption can be implemented on digital image data.

Transposition and substitution methods are techniques of data encryption. The transposition method in data security is used to randomize data by permuting the position of data sequence from the original data [3]. In the digital image encryption, this method is used to randomize the pixel position coordinate. The new pixel position coordinate is an encrypted image. The substitution methods are used to replace original data with other data [4][5]. This method is used to replace the pixel color intensity value to a different value [6]. The new pixel color intensity value is an encrypted image. These two methods can be combined to strengthen the security of encrypted image.

Transposition and substitution techniques have been used in numerous studies on digital image security. Ping's research suggests applying Henon Map keystream substitution and transposition for encryption. Henon Map comes in two models which are transposition and generating keystream. Henon Map transposition encryption and decryption functions can only be used for square-size-image N×N (width = height) [7].

Research by Lone proposes encryption using Random Matrix Affine Cipher (RMAC) transposition, Henon Map transposition, Logistic Map keystream substitution. The encryption process is carried out sequentially, so it takes longer time in encryption process. Research by Lone has limitation which is that the encryption process cannot be implemented on images with size of M≠N (width ≠ height) [8].

Research by Ratna proposes encryption using Cat Map transposition and Henon Map keystream substitution. The encryption process on image with size M≠N is carried out by adding dummy pixels, so the image size is larger than the original size. The limitation in Ratna's research is that the Cat Map transposition function can only carried out in the image with size N×N [9].

The Henon Map and Cat Map transposition functions have the advantage of being able to randomize the position of pixel coordinates based on the value of key variable, but these functions have limitations which can only be implemented on image with size N×N. Padding the image with size M ≠ N (width ≠ height) into N×N (width = height) can be done but it will enlarge the size of original image, so the encryption process times takes longer.

The Logistic Map functions has the advantage of being able to generate a keystream using two sensitive key variable which can be used to substitute pixel color intensity values, but to strengthen the security of encryption method, the substitution technique can be combined with transposition technique.

This paper proposes a combination of transposition and substitution encryption method called RaMSH-1 (Rama, Madenda, Suhatril, Harmanto). In the transposition encryption section, a modified Henon Map function is proposed, so that it can be applied to all image sizes without padding. Furthermore, the combination of the proposed transposition encryption function with the Logistic Map keystream substitution function aims to increase the encryption key combination so that the probability of finding the key will be smaller.

II. Methods

This research proposes the RaMSH-1 Map transposition and Logistic Map keystream substitutions functions for encryption and decryption methods. The development of proposed method aims to encrypt and decrypt digital image with all size of image and has a high level of security. This study is focused on 8-bit RGB images.

The RaMSH-1 Map transposition is used to randomize the pixel position coordinates and the Logistic Map is used to generate a keystream that is used to replace the pixel color intensity values. The encrypted image is represented by new pixel position coordinates and color intensity values. To convert the encrypted image back to the original image, the pixel position coordinates, and color intensity values must be restored. The proposed methods for encryption and decryption are shown in Fig. 1.

A. RaMSH-1 Map

RaMSH-1 Map is modification of the Henon Map transposition function. Henon Map function has limitation which can only perform encryption and decryption on image with size N×N (width = height) because there are a crossing transposition in coordinates x́ ← y and ý ← x, so some pixels in the original image with different coordinates (x,y) are transposed to the same coordinate (x́, ý) [10]. This cause- the pixels cannot return to the original coordinates when decryption process is carried out. Eq. (1) and (2) are the encryption and decryption functions of Henon Map Transposition.

$$I(\acute{x},\acute{y}) \leftarrow I(x,y)\ where \begin{cases} \acute{x} = 1 - cx^2 + y \bmod (N) \\ \acute{y} = (x+d) \bmod (N) \end{cases} \qquad (1)$$

$$I(x,y) \leftarrow I(\acute{x},\acute{y})\ where \begin{cases} \acute{x} = 1 - cx^2 + y \bmod (N) \\ \acute{y} = (x+d) \bmod (N) \end{cases} \qquad (2)$$

Modification of the Henon Map to RaMSH-1 Map is carried out by transposing x́ ← x and ý ← y. The modulo process is also carried out according to the size of image at these coordinates. Eq. (3) and (4) are the encryption and decryption function of the RaMSH-1 Map transposition.

$$I(\acute{x},\acute{y}) \leftarrow I(x,y)\ where \begin{cases} \acute{x} = 1 - cy^2 + x \bmod (M) \\ \acute{y} = (y+d) \bmod (N) \end{cases} \qquad (3)$$

$$I(x,y) \leftarrow I(\acute{x},\acute{y})\ where \begin{cases} \acute{x} = 1 - cy^2 + x \bmod (M) \\ \acute{y} = (y+d) \bmod (N) \end{cases} \qquad (4)$$

The variable keys of RaMSH-1 Map function are c and d with positive integer values. The width of the image is represented by M. The height of the image is represented by N [11]. The pixel coordinates are represented by (x, y). I(x,y) represents the original image as its value. The value of I(x́,ý) represents an encrypted image whose pixel position coordinates have been randomly generated. RaMSH-1 Map function can be applied in all image size.

B. Logistic Map

The chaotic function known as the logistic map is highly sensitive to the generation of keystream [12]. The original image's pixel color intensity values are substituted with the keystream value, changing the color intensity value. The Logistic Map function to generate keystream is defined in Eq. (5).

$$g_{n+1} = z \times g_n \times (1 - g_n) \qquad (5)$$

The Logistic Map function has two key variables g_0 and z which are real number with range of value $0 < g_0 < 1$ and $3.5 < z \leq 4$. The keystream is calculated by Eq. (6). The keystream is represented as e_n with value from 0 to 255. The value of n is the numbers of pixels in the image where n = (0, 1, …., T-1), T = M×N. Keystream substitution is carried out by XOR process in Eq. (7).

$$e_n = round(|g_n \times 10000|) \bmod 256 \qquad (6)$$

$$\acute{I}(\acute{x},\acute{y}) = e_n \oplus I(\acute{x},\acute{y})_n \qquad (7)$$

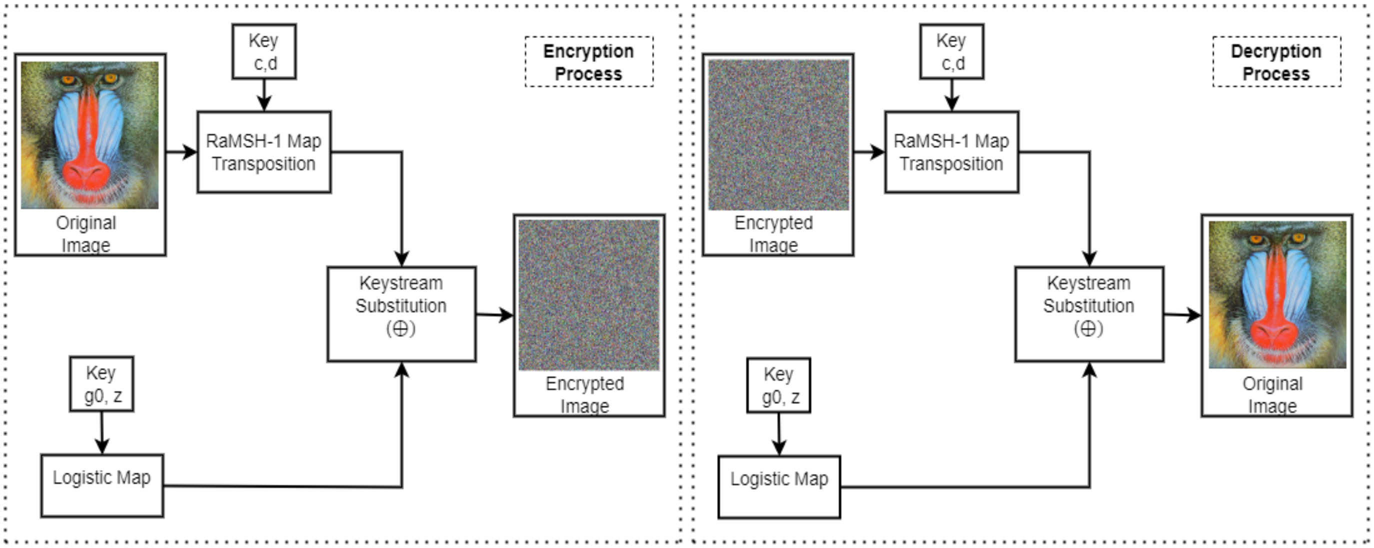


Fig. 1. Proposed methods for encryption and decryption.

C. Proposed Encryption Algorithm

RaMSH-1 Map transposition is used in Eq. (3) to randomize the pixel position coordinates during the encryption process. The encryption algorithm that is proposed is Algorithm 1. Algorithm 2 is used to modify the pixel color intensity values by keystream substitution.

Algorithm 1: Process for Proposed Encryption

```
Input: Variable c, d, g_0, z, Original Image $I(x,y)$
Output: Cipher Image $I'(x',y')$
$M \leftarrow width\ of\ image$
$N \leftarrow height\ of\ image$
$e_n \leftarrow Logistic\,(N, g_0, z)$
$n \leftarrow 0$
    for $x \leftarrow 0$ to $M-1$ do
        for $y \leftarrow 0$ to $N-1$ do
            $x' \leftarrow (1 - cy^2 + x)\ \mathrm{mod}(M)$
            $y' \leftarrow (y + d)\ \mathrm{mod}(N)$
            $I'(x',y') \leftarrow e_n \oplus I(x,y)$
            $n \leftarrow n+1$
        end for
    end for
```

Algorithm 1 step $e_n \leftarrow Logistic\,(N, g_0, z)$ is to call the Logistic Map function for generating keystream. Steps $x' \leftarrow (1 - cy^2 + x)\ \mathrm{mod}(M)$ and $y' \leftarrow (y + d)\ \mathrm{mod}(N)$ are randomization of pixel position coordinates by RaMSH-1 Map transposition. Step $I'(x',y') \leftarrow e_n \oplus I(x,y)$ is generating encrypted image using a keystream replacement to change the value of the pixel's color intensity. The encryption technique uses a single loop to carry out the transposition and substitution operations.

D. Logistic Map Keystream Algorithm

The Logistic Map function in Eq. (5) and (6) are applied in the keystream generation process. The Logistic Map keystream algorithm is used in Algorithm 2.

Algorithm 2: Keystream Generation of Logistic Map

```
    function $Logistic(M, N, g_0, z)$
    $T \leftarrow M \times N$
        for $n \leftarrow 0$ to $T-1$ do
            $g_{n+1} \leftarrow z \times g_n \times (1 - g_n)$
            $e_n \leftarrow round\left(\left|g_{n+1} \times 10000\right|\right)\mathrm{mod}(256)$
        end for
    end function
```

Algorithm 2 step $T \leftarrow M \times N$ is a calculation of pixels number in the image. Step g_{n+1} is the Logistic Map function. Step e_n is a function to generate keystreams Logistic Map. Keystreams were generated in an amount equal to the number of pixels in the image.

E. Proposed Decryption Algorithm

The decryption method uses the same procedure as the encryption process, but instead returns values for pixel position coordinates and color intensity. The proposed decryption algorithm is Algorithm 3.

Algorithm 3: Process for Proposed Decryption

```
Input: Variable c, d, g_0, z, Cipher Image $I'(x',y')$
Output: Original Image $I(x,y)$
$M \leftarrow width\ of\ image$
$N \leftarrow height\ of\ image$
$e_n \leftarrow Logistic\,(N, g_0, z)$
$n \leftarrow 0$
    for $x \leftarrow 0$ to $M-1$ do
        for $y \leftarrow 0$ to $N-1$ do
            $x' \leftarrow (1 - cy^2 + x)\ \mathrm{mod}(M)$
            $y' \leftarrow (y + d)\ \mathrm{mod}(N)$
            $I(x,y) \leftarrow e_n \oplus I'(x',y')$
            $n \leftarrow n+1$
        end for
    end for
```

The variable key used in the encryption and decryption processes is same. The distinction between the encryption and decryption processes is that the decryption process uses encrypted images as input, whereas step $I(x,y) \leftarrow e_n \oplus I'(x',y')$ of algorithm 3 restores encrypted images to their original forms.

III. Result and Discussion

The research experiment of proposed algorithm used image data with variety of colors, shape, and texture. The size of image is M = N: 512×512 pixel, M > N: 1500×1000 pixel, and M < N: 2850×3200 pixel. Matlab R2020a, which runs on a computer system with an Intel (R) Core (TM) i7-4790 CPU, a processor clocked at 3.60 GHz, 16 GB of RAM, and the Microsoft Windows 10 operating system, is the software utilized in this study. The analyses used in the experimental are histogram, correlation, NPCR, UACI, decrypted image quality, key sensitivity, and key space.

A. Histogram Analysis

The distribution of pixel color intensity in an image is displayed using a histogram [13][14]. In this study, the pixel color intensity distribution was represented by a histogram, which can demonstrate the effectiveness of pixel randomization in the encrypted image produced by the proposed technique. If the histogram displays a particular pattern, the information in the image can be interpreted easily [15]. In column 3 of Table I, are encrypted images that visually differ from the original image. Columns 4 and 5 show how the histogram of every original image follows a certain pattern, while the histogram of every encrypted image follows a uniform pattern. This shows how the proposed algorithm can change the color intensity value and randomize all pixel coordinates within the encrypted image. It becomes challenging to interpret the information in an encrypted image [16].

B. Correlation Analysis

The correlation parameter is used to determine how similar two neighboring pixels are to one another in an image's horizontal (I(x,y) and I(x+1,y)), vertical (I(x,y) and I(x,y+1)), and diagonal directions (I(x,y) and I(x+1,y+1)). When the correlation value is near to 1 or -1, two neighboring pixels are either closely associated or have comparable colors [17]. In contrast, if the correlation value is close to 0, either both adjacent pixels have a high degree of randomness or there is low color correlation between them [18]. Eq. (8) functions as the correlation formula, where A and B are similar images that represent two nearby pixels on all three directions. The average value of all the pixels in images A and B is represented by variables $\bar{A}$ and $\bar{B}$, respectively.

$$r = \frac{\sum_x^M \sum_y^N (A_{xy}-\bar{A})(B_{xy}-\bar{B})}{\sqrt{(\sum_x^M \sum_y^N (A_{xy}-\bar{A})^2)(\sum_x^M \sum_y^N (B_{xy}-\bar{B})^2)}} \qquad (8)$$

Columns 5, 6, and 7 of Table II display that the correlation absolute value of the encrypted image ranges from 0.00024 to 0.10269. In columns 2, 3, and 4, the encrypted image has a lower correlation value than the original image. This shows that the proposed algorithm can randomize each bit of information in the encrypted image, making it impossible to crack the data. The level of randomization in pixel color intensities is displayed in Table II. The curve lines of the original images can be seen moving diagonally in columns 2, 3, and 4, which indicates that the two neighboring pixels in the original image have a strong correlation. However, columns 5, 6, and 7 display the random movement of the curve lines of the encrypted image, indicating that the color correlation between two neighboring pixels in the encrypted image is low. The proposed image encryption algorithm can randomize the position and color intensity of every pixel so that the messages contained in the encrypted image are difficult to read, according to the correlation value and correlation curve in Table II.

C. Decrypted Image Quality Analysis

Peak Signal to Noise Ratio (PSNR) and MSE (Mean Square Error) are used to evaluate the quality of encrypted images. Eq. (9) functions as the PSNR formula, and Eq. (10) is used to calculate the MSE value between the original image and the decrypted image [9]. The quality of the decrypted images is evaluated using an RGB image with a resolution of 2850×3200 pixels.

$$PSNR = 20\ Log_{10} \frac{255}{MSE} \qquad (9)$$

$$MSE = \frac{1}{M \times N} \sum_x^M \sum_y^N (I(x,y) - \acute{I}(x,y)^2) \qquad (10)$$

The encryption algorithm used by the previous researcher is listed in Table III column 1. Visually, the original image, the encrypted image, and the decrypted image are shown in columns 2, 3, and 4. MSE value and PSNR value are displayed in columns 5 and 6, respectively. There are numerous encrypted image pixels that differ from the original image in rows 2 and 3. It implies that the Henon Map and Cat Map transposition cannot perform the encryption and decryption on an image with a size of M ≠ N (width ≠ height), resulting in noise in the decrypted image, as seen by values of PNSR ≠ ∞ and MSE ≠ 0. The decrypted image in row 4 is identical to the original image with values of MSE = 0 and PSNR = ∞ [19]. It means that the proposed algorithm can be used to process images of all image size.

TABLE I. ORIGINAL AND ENCRYPTED IMAGE HISTOGRAM

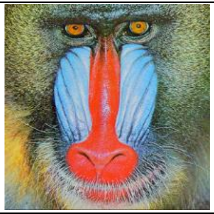
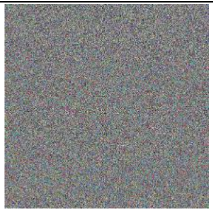
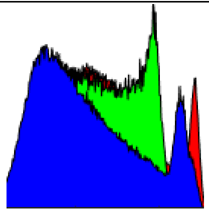
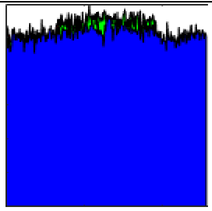
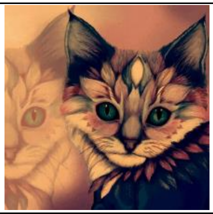
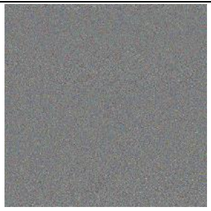
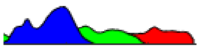
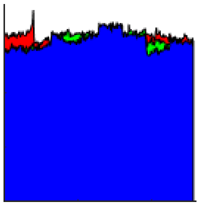
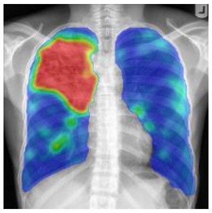
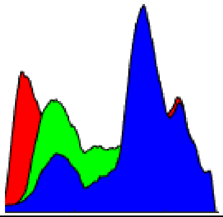
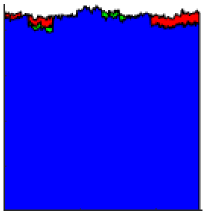
Name	Image		Histogram	
	Original	***Encrypted***	***Original***	***Encrypted***
Baboon (512×512)				
Cat (1500×100)				
Lung (2850×3200)				

TABLE II. ORIGINAL AND ENCRYPTED IMAGE CORRELATION

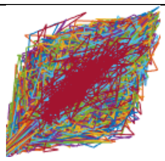
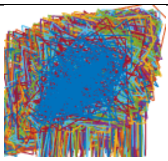
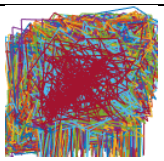
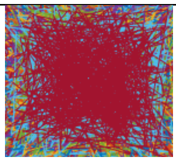
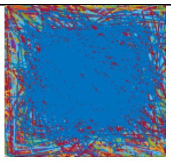
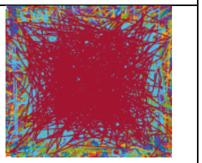
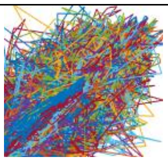
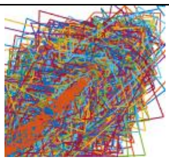
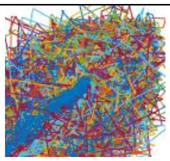
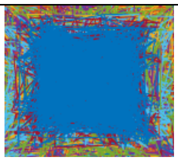
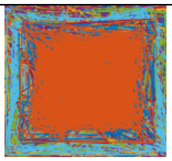
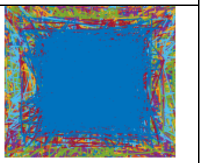
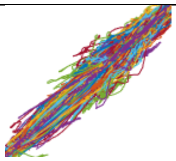
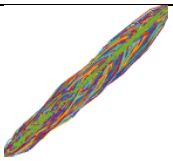
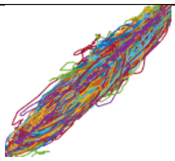
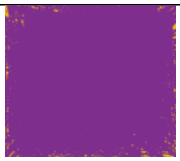
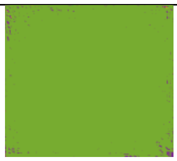
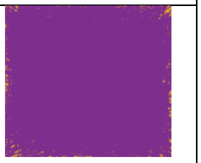
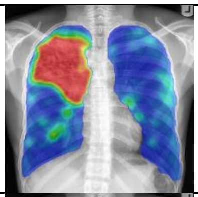
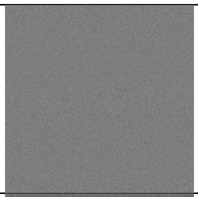
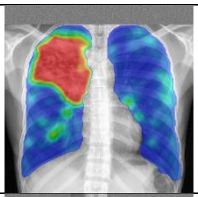
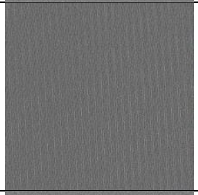
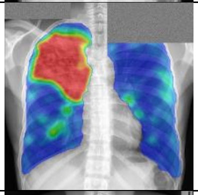
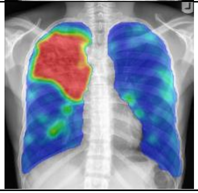
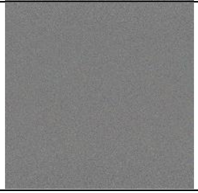
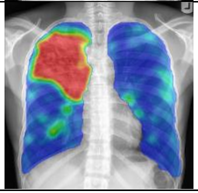
Name	Correlation of Original Image			Correlation of Encrypted Image		
	Horizontal	*Vertical*	*Diagonal*	*Horizontal*	*Vertical*	*Diagonal*
Baboon (512×512)	 0.86657	 0.75875	 0.72620	 -0.07237	 -0.00229	 -0.00100
Cat (1500×100)	 0.98543	 0.98256	 0.97936	 -0.09986	 -0.00142	 0.00187
Lung (2850×3200)	 0.99897	 0.99959	 0.99871	 -0.10269	 -0.00024	 -0.00090

TABLE III. DECRYPTED IMAGE ANALYSIS

Encryption Algorithm	Original Image	Encrypted Image	Decrypted Image	MSE	PSNR
Henon Map Transposition & Henon Map Keystream Substitution (Ping, 2018 [7])				1079.84	17.80 dB
Cat Map Transposition & Henon Map Keystream Substitution (Ratna, 2021 [9])				1322.47	16.92 dB
RaMSH-1 Map Transposition & Logistic Map Keystream Substitution (Proposed Algorithm)				0	∞

D. NPCR and UACI Analysis

The NPCR (Number of Pixel Change Rate) parameter measures how differently colored pixels in the original image affect the outcomes of the encrypted image [20][21]. The UACI parameter assesses the average pixel change percentage in the encrypted image. The security level of an algorithm's resistance to differential attack is shown by the NPCR and UACI value [7][22].

Eq. (11) and (12) are the formulas of NPCR and UACI where C1 and C2 are two encrypted images. The D_{xy} coefficient is the similarity value of pixel in same coordinate of both C1 and C2 images. The $C1_{xy}$ and $C2_{xy}$ are color intensity values of the C1 and C2 at the (x,y) position.

$$NPCR = \frac{1}{M\times N}\sum_{x}^{M}\sum_{y}^{N} D_{xy} \times 100\% \quad (11)$$

$$UACI = \frac{1}{M\times N}\sum_{x}^{M}\sum_{y}^{N}\frac{|C1_{xy}-C2_{xy}|}{255}\times 100\% \quad (12)$$

The largest UACI value is 33.37521%, while the largest NPCR value is 99.61014%, according to Table IV. The proposed algorithm is resistant to differential attack [8] and has sensitivity of pixel changes in original image effects to encrypted image results since both values are above the standard 99.6% of NPCR value and 33.3% of UACI value.

E. Entropy Analysis

The entropy parameter calculates the minimal average number of bits required to decode a symbol from a sequence of bits. The entropy value displays the effectiveness of the algorithm's security level and the degree of unpredictability in the encrypted image [23]. Entropy is calculated using Eq. (13), where Pi is the chance that a pixel with the value of i will appear. Entropy can have a maximum value of 8 [8][24].

$$H = \sum_{i=0}^{i=255} P_i Log_2\left(\frac{1}{P_i}\right) \qquad (13)$$

The highest entropy value, according to Table IV, is 7.99980. It shows that the proposed algorithm can randomize the pixel coordinates and change the pixel color intensity value in the encrypted image and is resistant to entropy attacks [14].

TABLE IV. NPCR, UACI, ENTROPY OF ENCRYPTED IMAGE

Name	NPCR (%)	UACI (%)	Entropy
Baboon	99.61014	33.02773	7.99822
Cat	99.60851	33.29189	7.99964
Lung	99.60999	33.37521	7.99980

F. Key space Analysis

The proposed algorithm is composition of Logistic Map substitution and RaMSH-1 Map transposition. The key variables are c, d, g_0, and z. The range of values c, d ∈ $\mathbb{Z}^+$. The range of values for z and g_0 are $3.5699 < r \leq 4$ and $0 < g_0 < 1$, respectively, where g_0, r ∈ $\mathbb{R}^+$ [8]. In Matlab, the largest integer number value is $2^{64} \approx 1.8 \times 10^{19}$ and the largest real number value is 1.8×10^{308}. The default mantissa value in floating-point is 10^{15} [23]. Table V shows the proposed algorithm's key space.

TABLE V. KEY SPACE

Function	Variable Key	Key Space
RaMSH-1 Map	c, d ∈ $\mathbb{Z}^+$	3.24×10^{38}
Logistic Map	g0 ∈ [0, 1]; z ∈ [3.5699, 4]	3.24×10^{631}
Proposed Algorithm	c, d ∈ $\mathbb{Z}^+$ g0 ∈ [0, 1]; z ∈ [3.5699, 4]	1.05×10^{670}

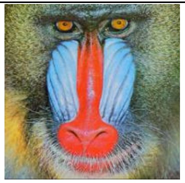
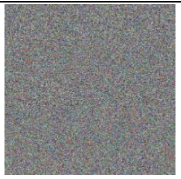
The proposed algorithm's key space is $(1.8 \times 10^{19}) \times (1.8 \times 10^{19}) \times (1.8 \times 10^{308}) \times (1.8 \times 10^{308}) \times (10^{15}) \approx 1.05 \times 10^{670}$. A computer that can perform 10^{24} computations in a second is used to simulate testing every key combination [22]. It would take 3.33×10^{639} years to complete the vast array of computations through the computer in a single year $(1.05 \times 10^{670}) \div (10^{24}) \times 365$ (days) × 24 (hour) × 60 (min) × 60(s) = 3.33×10^{639}. This amount of time for attempting every combination is sufficient to fend off a brute force attack [23][25].

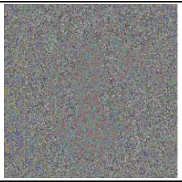
G. Key Sensitivity Analysis

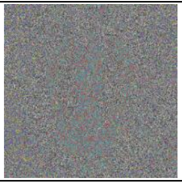
The proposed algorithm using four variable keys. Key of c and d are positive integer numbers. Key of g_0 and z are positive real numbers. The decrypted key value with small changes on decrypted process will affect the decrypted image. The key sensitivity can be seen in Table VI.

Table VI row 2 shows the decrypted keys for decryption process are c = 20, d = 30, g_0 = 0.1, and z = 3.6. Each key is changed with a very small value change in key values. Rows 3, 4, 5, and 6 show the decrypted image still unreadable. In row 5 shows the key sensitivity of g_0 reached 10^{-16} which indicate the proposed algorithm is robust against brute force attack.

TABLE VI. KEY SENSITIVITY ANALYSIS

Key Sensitivity	Key Value	Decrypted Image
Decryption Key	c = 20; d = 30 g_0 = 0.1; z = 3.6	
Sensitivity of Decryption Key c	c = 21; d = 30 g_0 = 0.1; z = 3.6	
Sensitivity of Decryption Key d	c = 20; d = 31 g_0 = 0.1; z = 3.6	

Sensitivity of Decryption Key g_0	c = 20; d = 30
	$g_0 = 0.1 + 10^{-16}$;
	z = 3.6
		

Sensitivity of Decryption Key z	c = 20; d = 30 $z = 3.6 + 10^{-15}$; g_0 = 0.1	

IV. CONCLUSIONS

The Henon Map transposition function is modified in the RaMSH-1 Map transposition. The proposed algorithm can be applied for all image size without padding. The key variable of proposed algorithm generated a key space of 1.05×10^{670}. The entropy values reached 7.99980 out of 8 as perfect entropy value. The correlation value for all encrypted images is very close to 0, indicating that there is no correlation between the color intensity of neighboring pixels. Entropy and correlation analyses show that the encrypted image's information is all scrambled, which makes it hard to decipher. The results of the key sensitivity reached 10^{-16} which indicate the proposed algorithm is very sensitive. The UACI value of 33.37521 and the NPCR value of 99.61014 respectively show that the proposed algorithm is resistant to differential attack.

V. FUTURE WORK

The proposed algorithm's weakness is that it only uses four variable keys. The algorithm's key space can be expanded. The data security can be improved by combining RaMSH-1 Map transposition, Logistic Map substitution, and Cat Map transposition.

REFERENCES

[1] M. K. Hasan et al., “Lightweight Cryptographic Algorithms for Guessing Attack Protection in Complex Internet of Things Applications,” Complexity, vol. 2021, pp. 1–9, 2021, doi: 10.1155/2021/5540296.

[2] B. K. Yakti, S. A. Sudiro, S. Madenda, and S. Harmanto, “Hardware Implementation Method of Secret Data Security on Fpga Based on Zig-Zag Map Encryption and Stegano Algorithms,” J. Theor. Appl. Inf. Technol., vol. 100, no. 17, pp. 5572–5582, 2022.

[3] K. Renuka and G. N. Harshini, “Analysis and Comparison of Substitution and Transposition Cipher,” Int. J. Res. Anal. Rev., vol. 6, no. 2, pp. 549–555, 2019.

[4] P. Poonia and P. Kantha, “Comparative Study of Various Substitution and Transposition Encryption Techniques,” Int. J. Comput. Appl., vol. 145, no. 10, pp. 24–27, 2016, doi: 10.5120/ijca2016910783.

[5] R. S. Navale, A. N. Jalgeri, and B. B. Jagadale, “Survey on various substitution techniques for Cryptography,” Int. J. Res. Dev. Technol., vol. 7, no. 4, pp. 613–616, 2017.

[6] Y. P. K. Nkandeu and A. Tiedeu, “An image encryption algorithm based on substitution technique and chaos mixing,” Multimed. Tools Appl., vol. 78, no. 8, pp. 10013–10034, 2019, doi: 10.1007/s11042-018-6612-2.

[7] P. Ping, F. Xu, Y. Mao, and Z. Wang, “Designing permutation substitution image encryption networks with Henon map,” Neurocomputing, vol. 283, pp. 53–63, 2018, doi: 10.1016/j.neucom.2017.12.048.

[8] P. N. Lone, D. Singh, and U. H. Mir, “A novel image encryption using random matrix affine cipher and the chaotic maps,” J. Mod. Opt., vol. 68, no. 10, pp. 507–521, 2021, doi: 10.1080/09500340.2021.1924885.

[9] A. A. P. Ratna et al., “Chaos-based image encryption using Arnold’s cat map confusion and Henon map diffusion,” Adv. Sci. Technol. Eng. Syst., vol. 6, no. 1, pp. 316–326, 2021, doi: 10.25046/aj060136.

[10] R. D. Syah, S. Madenda, R. J. Suhatril, and S. Harmanto, “Hybrid Digital Image Cryptography Using Composition of Henon Map Transposition and Logistic Map Substitution,” in 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), 2022, pp. 1–6, doi: 10.1109/ICOSNIKOM56551.2022.10034926.

[11] V. Tyagi, Understanding Digital Image Processing. CRC Press, 2018.

[12] A. T. Ruslan, Marwan, and Q. Aini, “Behavior of logistic map and some of its conjugate maps,” AIP Conf. Proc., vol. 2641, no. 1, p. 20002, 2022, doi: 10.1063/5.0115103.

[13] H. Kaur and N. Sohi, “A Study for Applications of Histogram in Image Enhancement,” Int. J. Eng. Sci., vol. 06, no. 06, pp. 59–63, 2017, doi: 10.9790/1813-0606015963.

[14] A. Benlashram, M. Al-ghamdi, R. Altalhi, and P. Kaouther, “A novel approach of image encryption using pixel shuffling and 3D chaotic map A novel approach of image encryption using pixel shuffling and 3D chaotic map,” in Journal of Physics: Conference Series, 2020, vol. 1447, doi: 10.1088/1742-6596/1447/1/012009.

[15] N. Munir, M. Khan, A. Al Karim Haj Ismail, and I. Hussain, “Cryptanalysis and Improvement of Novel Image Encryption Technique Using Hybrid Method of Discrete Dynamical Chaotic Maps and Brownian Motion,” Multimed. Tools Appl., vol. 81, no. 5, pp. 6571–6584, 2022, doi: 10.1007/s11042-021-11810-2.

[16] H. Gao and W. Zeng, “Image compression and encryption based on wavelet transform and chaos,” Comput. Opt., vol. 43, no. 2, pp. 258–263, 2019, doi: 10.18287/2412-6179-2019-43-2-258-263.

[17] R. D. Syah and R. J. Suhatril, “Digital Image Cryptography Using Combination of Arnold’s Cat Map and Bernoulli Map Based on Chaos Theory,” Int. Res. J. Adv. Eng. Sci., vol. 4, no. 2, pp. 258–262, 2019, doi: 10.5281/zenodo.3153337.

[18] S. Sabir and V. Guleria, “Multilayer color image encryption using random matrix affine cipher, RP2DFrHT and 2D Arnold map,” Multimed. Tools Appl., vol. 80, pp. 27829–27853, 2021, doi: 10.1007/s11042-021-11003-x.

[19] M. T. Suryadi, M. Y. T. Irsan, and Y. Satria, “New modified map for digital image encryption and its performance,” J. Phys. Conf. Ser., vol. 893, no. 1, 2017, doi: 10.1088/1742-6596/893/1/012050.

[20] Y. Liu and Y. C. Ko, “Image Processing Method Based on Chaotic Encryption and Wavelet Transform for Planar Design,” Adv. Math. Phys., vol. 2021, pp. 1–12, 2021, doi: 10.1155/2021/7511245.

[21] Y. Chen, S. Xie, and J. Zhang, “A Hybrid Domain Image Encryption Algorithm Based on Improved Henon Map,” Entropy, vol. 24, no. 2, pp. 1–28, 2022, doi: 10.3390/e24020287.

[22] S. Kanwal et al., “An Effective Color Image Encryption Based on Henon Map, Tent Chaotic Map, and Orthogonal Matrices,” Sensors, vol. 22, no. 12, p. 4359, 2022, doi: 10.3390/s22124359.

[23] L. Zhang, L. Zhang, and L. Zhang, “Application Research of Digital Media Image Processing Technology Based on Wavelet Transform,” J Image Video Proc, vol. 138, no. 2018, 2018, doi: 10.1186/s13640-018-0383-6.

[24] Y. A. Hamza, “Highly Secure Image Steganography Approach Using Arnold’s Cat Map and Maximum Image Entropy,” in Proceedings of the International Conference on Information and Communication Technology, 2019, pp. 134–138, doi: 10.1145/3321289.3321323.

[25] M. C. Alipour, B. D. Gerardo, and R. P. Medina, “A secure image encryption architecture based on pseudorandom number generator and chaotic logistic map,” ACM Int. Conf. Proceeding Ser., pp. 154–159, 2019, doi: 10.1145/3352411.3352436.

State-of-the-Art of the Swarm Ship Technology for Alga Bloom Rapid Monitoring

Denny Darlis[1], Indra Jaya[2], Karlisa Priandana[3], Yopi Novita[4], Ayi Rachmat[5]
Diploma of Telecommunication Technology, Telkom University, Bandung, Indonesia[1]
Marine Science and Technology Department, IPB University, Bogor, Indonesia[2, 4, 5]
Computer Science Department, IPB University, Bogor, Indonesia[3]

***Abstract*—The swarm intelligence has become an interesting topic for employing of multi-agent robotics with specific purpose. The capability of multi-coordination, scalability and goals-oriented control in spatial and temporal environment are already concerned and proven for several applications such as in military patrol, and drones leader-follower coordination. In marine-based environment, swarm intelligence adopted by ASV or ROV has been used for water quality and environment monitoring with sufficient optimized results making it convenient for rapid assessments. In this paper, the arrangement for building a trusted cyber-physical systems for algal bloom rapid assessment using swarm ship technology were explained in state-of-the-art perspectives. The minimum requirements for sensing, vehicle controlling, and communication of this system with others were explored as well as algorithm chosen for the best known configuration to monitor algal bloom events before spreading so fast to larger area. Some models were explained to show the robustness of autonomous unmanned ship control. From this point of view, we concluded that swarm ship technology has become an important potential implementation for near real time in situ monitoring compared to other decision making method such as laboratory examination or remote sensing-based results. The results of this review open the opportunity to realization of swarm ship technology in cyber physical system for monitoring algal bloom in specific area near real time efficiently.**

***Keywords*—Swarm ship; algal bloom; cyber physical system; rapid monitoring**

I. Introduction

Dissolved oxygen in the water is produced through the process of photosynthesis by phytoplankton. However, the degree of respiration will be greater than photosynthesis when there is no sunlight. This doesn't just happen at night. In the surface layer, the oxygen level will be higher, due to the diffusion process between water and free air and the process of photosynthesis. With increasing depth there will be a decrease in the level of dissolved oxygen, since the process of photosynthesis is decreasing and the existing oxygen levels are widely used for breathing and oxidation of organic and inorganic materials. Dissolved oxygen concentration is an important parameter in determining the quality of waters. Oxygen concentration is influenced by the balance between oxygen production and consumption in the ecosystem. Oxygen is produced by the autotrophic community through the process of photosynthesis and consumed by all organisms through breathing. The solubility of oxygen in the waters is affected by temperature and salinity. The higher the temperature and salinity of the waters, the smaller is the solubility of oxygen. When phytoplankton bloom, the sunlight is blocked from entering the water column. So that dissolved oxygen is decreasing [1]. Phytoplankton is a major component of the food chain and the main production of the marine environment. However, a high abundance of phytoplankton can have a harmful effect on aquatic ecosystems. In addition, they can produce toxic substances that will accumulate in their consumers. This accumulation can be dangerous for humans or animals [2].

The phenomenon of algal bloom occurs quite often in several waters in Indonesia and the world. Research conducted by [3] in Jakarta Bay mentioned the occurrence of eutrophication since 1986 which caused the death of several types of fish and other organisms due to several types of phytoplankton such as Skeletonema which reached 534 x 106 cells / m3 in September 1984 and in April 1985 reached 2,316 x 106 cells / m3. Chaetoceros reached 5.6 x 106 cells/m3 in April 1985. Noctiluca reached 0.4 x 106 cells/m3 in September 1984. Prorocentrum was found to be very dominant in June 1988. The phenomenon of algal bloom on Ringgung beach, Lampung Bay, also resulted in the death of fish farmed in the Floating Net Cage [1]. Other events are also triggered by several types of algae such as Cyanobacteria, Dinoflagellates, Cyanobacteria, and Diatoms in the Ambon Bay and Jakarta Bay areas [4]. Research on the analysis of phytoplankton abundance causing HAB (Harmful Algal Bloom) in Lampung Bay Waters in the western season and eastern season was also carried out to identify algae that have the potential to cause HAB in Lampung Bay and see patterns of phytoplankton abundance relationships that have the potential to cause HAB with nutrients contained in the waters of Lampung Bay, Pesawaran Regency, Lampung Province. In the eastern season, the abundance of phytoplankton that causes HAB in the waters of Lampung Bay is dominated by the species Ceratium sp. with an average of 1,802 ind/L while in the western season the abundance of phytoplankton in Lampung Bay is dominated by Nitczchia sp., with an average abundance of 161,207 ind/L [5]. The incident that occurred in the Thousand Islands, Jakarta Bay on October 15, 2020 was caused by the phytoplankton type Trichodesmium. Real time sampling of phytoplankton that undergoes blooming has been carried out and the parameters analyzed are the composition of phytoplankton types and water quality. This is closely related to seasonal changes and an increase in nutrient content in waters, especially nitrates and orthophosphates, which support the rapid growth of phytoplankton [6].

Factors causing algae abundance carried out in tropical lakes, Telaga Menjer Wonosobo Indonesia mentioned that observations and sampling carried out three times with an interval of one month at seven locations in the photic zone resulted in concentrations of macronutrients and micronutrients in Telaga Menjer spatially even in all locations and temporal concentrations increased slightly the same during the measurement period. The macronutrients Cu, NO_2, Si, and Na are the determining factors for algae bloom in Lake Menjer Wonosobo. The effect of Cu, NO_2 and Si concentrations is inversely proportional to the abundance of algae, while the concentrations of Ca, Na and Mo are proportional to the abundance of algae [7]. Meanwhile [8] links the causes of algal bloom events with climate change in the process of assessing and mitigating risks in mariculture and marine fisheries.

II. Problem Statement

In monitoring the condition of the aquatic environment, some important parameters that need to be considered are the concentration of dissolved oxygen, turbidity conditions (turbidity), water temperature, chlorophyll-a concentration and the type of algae that develops. By using the necessary sensors, and installed directly in a fixed or moving condition, monitoring can be carried out in real-time or periodically.

The use of water vehicles that can move simultaneously opens up the potential for the application of swarming methods, which are inspired by swarming intelligence in animals or mammals in nature. A group robotics system is the application of a multi-agent system that has intelligence when in groups such as multi-robots that apply ant colony optimization algorithms [9]. This system was introduced by [10]. The application of other Swarm Robotic systems can be found in studies conducted by [11] [12] [13] and [14].

To provide some proof-of-concept to this swarmship technology, we need to build several surface vessel integrated with sensors, control and communication system and make some performance measurements.

III. Research Framework

In this study, an algal bloom monitoring system used several autonomous vessels that could directly sense environmental conditions and water quality. At the initial stage the autonomous ship is designed starting from its mechanical part, then equipped with a propulsion system, a control system, These ships will communicate with each other at a certain distance in order to exchange information related to the parameters of acidity, dissolved oxygen, algae concentration and weather in a certain water area. If in the region it is informed of an anomaly related to the parameters measured by one or several ships simultaneously, decision-making for monitoring together will be carried out and the handling is quickly carried out both automatically and by humans. The system to be designed and implemented is 3 – 5 autonomous vessels that have group intelligence in ensuring water conditions related to potentially harmful algae concentrations. Once it is confirmed that the anomalies in the parameters of the waters being monitored have the potential to produce algae abundance, then the next treatment will be carried out automatically or handed over to humans as decision makers. The real-time data retrieved will also be disseminated through the selected communication network so that it can be used for other purposes.

After reviewing several libraries related to Harmful Algal Bloom (HAB) and its monitoring methods, one of the in-situ measurement methods using several autonomous ships that can coordinate with each other using wireless communication by running an intelligent swarm-based optimization algorithm was chosen. In its application, these ships will send spatial and temporal data from the aquatic environment and be monitored through physical cyber systems located on land so that human in-the loop system can be made.

At this time several preliminary studies have been carried out related to the construction of prototypes of Automatic Surface Vehicle (ASV) ships that have sensors, control systems, navigation and standard communications. In the year of this research, several additional ASV ships will be built and developed as a physical cyber system that can communicate with each other in exchange for sensor information and their respective positions. In this section, an intelligent computing model is also developed by ships in order to be able to analyze water quality containing potential toxic algae, exchange of information with other vessels, actions that need to be carried out automatically and feedback systems for these information. Next, by testing several intelligence algorithms that have been analyzed and simulated before, the joint decision making interpreted by the navigation system and the movement of these autonomous ships can be optimized. It is hoped that from the results of some of these trials, several characteristics can be mapped in determining the most effective time in monitoring algal bloom in certain waters. This research also opens up opportunities for further research in terms of mitigating algal bloom conditions quickly and precisely.

IV. Literature Review

Early detection of Algal Bloom is an important topic to do, so that earlier and more accurate treatment can be carried out before it spreads more widely to the surrounding waters. Various direct (in-situ) studies in identifying algae that cause it or indirectly in the laboratory have also been carried out by several researchers in the field of environmental ecology. Several technologies have also been used, such as placing special buoys that are used to monitor their abundance, as well as the factors that cause them. But until now, this incident has not been well predicted.

In monitoring, detecting and handling the abundance of harmful algae (Harmful Algal Bloom), several studies have been carried out in several ways. Generally, the method used is to take samples directly in the water environment to be observed, and then the characteristics will be determined in the laboratory. Sampling is carried out several times in several places during a certain time and season. The sample is then examined by cyst extraction, sediment culture experiments, cyst culture experiments, DNA extraction and molecular identification. The parameters measured in general are water quality, oxygen concentration, macronutrients and micronutrients such as nitrates and orthophosphates along with concentrations [15] of phytoplankton and or chlorophyll-A in the sampling area. Another approach taken is to extract information from imagery captured by remote sensing satellites. Research conducted on the [16] [17] algal bloom phenomenon that occurred in the waters of Lampung Bay verified SPOT 4 satellite images using the Red Tide algorithm quantitatively and qualitatively. This remote sensing approach is sharpened by the use of several machine learning algorithms based on CNN with spatial and temporal analysis methods, LSTM alongside random forests, and SVM classification methods as performed by the HABNet application architecture. The results showed that their method could detect the incidence of algal bloom with a maximum accuracy of 91% and a Kappa coefficient of 0.81 for Karenia brevis (K. brevis) type algae on the Florida coast (as many as 2850 events from 2003 to 2018 [18]. The use of biosensors to monitor the incidence of algal bloom is also an opportunity to accelerate the handling of these events such as several studies reviewed where the biosensor has the potential to be installed far enough but can provide measurement information quickly and continuously on algae measurements and biotoxin levels as needed and at a low cost using [19] an environment sample processor (ESP) that can analyze DNA quickly and continuously in-situ.

During the years when research began to consider mobile robotics, the behavior of animal groups appeared in several key observations. The V formation arises from the basic local behavior of each bird. A proposed behavior-based formation control for multi-robotic teams also began as related research such as the coordination of groups of mobile robots using the rules of the nearest neighbors. Another articles focus on algorithms and theories regarding its flocking multi-agent dynamic systems. Some of the reasons to consider are instability and dynamic systems as its problems with oscillations. The chain of formation requires fine tuning of local control. Control engineers should consider 3D scenarios when dealing with a group of AUVs or UUVs [20].

Swarm intelligence is a decentralized collective behavior, a self-organized system (natural or artificial) that can maneuver quickly in a coordinated manner. In nature, this closed loop, that is, collaborative behavior is unique in each species. Nature has proven that when individual beings collaboratively work and think together as a unified system towards a common goal, they are more likely to achieve that goal faster and more accurately than if they tried it individually. In other words, they are smarter together than themselves. Ants lay pheromones that direct each other to the resource, the bees use vibrations, the fish feel vibrations in the water and the birds detect movements that spread through the swarm.

Swarm intelligence in the robotics domain has a wide range of applications and benefits. The main benefits of herd intelligence include:

1) Flexibility: The swarm system responds to internal disturbances and external challenges.

2) Resiliency: The task completes regardless of whether some agents fail.

3) Self-organization: Roles are undefined - they arise.

4) Adaptation: The herd can adapt to predetermined and new stimuli.

5) Decentralization: There is no central control, allowing for fast local collaboration.

The most suitable applications for robot swarms require smaller fleets of robots to perform tasks, such as mapping and foraging in places that are difficult for humans to reach. For example, swarms of search and rescue robots of various sizes can be sent to places that rescue workers cannot safely reach. In general, swarms of robots can be utilized for tasks as mentioned in Fig. 1 [21] [22].

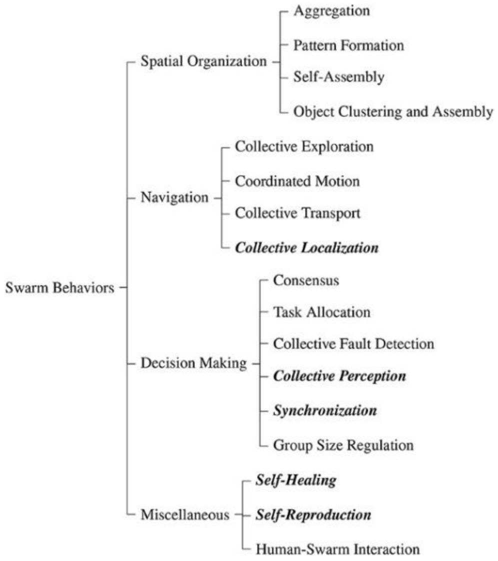


Fig. 1. Taxonomy of the application of swarm behavior to robotics.

V. Challenge and Discussion

Swarmship techology is proposed as a novel solution to the global threat posed by harmful algal blooms. This work is being done in parallel with the development of physical USVs that can strain and skim algae from the water [23]. The rate of algae collection need to be studied relative to: the diffusion of the algae in the water (due to turbulence), the composition of the swarm as either few large USVs or many small USVs, and the quantity of constant-size USV collected by the swarm formation. The results should be shown as plots of uncollected biomass vs. time, and as maps of the algae distribution after the swarmship have begun collection. Both a partitioning and non-partitioning controls approach are taken, which impose different hardware and communication requirements on the swarmship.

Meanwhile we developed several ships which has water quality monitoring device as shown on Fig. 2.

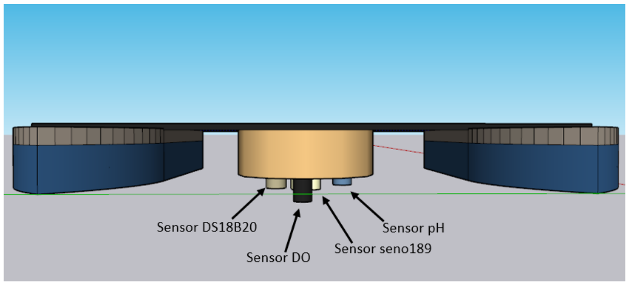


Fig. 2. Ships model for algal monitoring.

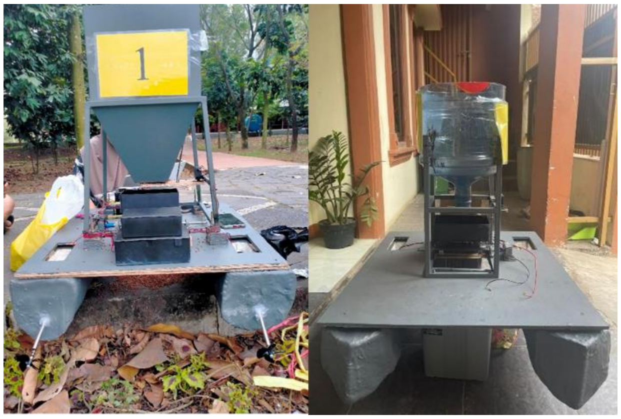


Fig. 3. Two models of swarmship implemented.

To aid in the development of this swarmship, several sets of scenarios are needed to inform the swarm designer about (i) the ability of the swarmship to collect a diffusive substance, i.e. the algae, (ii) whether how many USVs should be used, and (iii) what scaling effects are present as more and more USVs are need to be added to the swarm formation. Collecting a diffusive substance is an interesting problem, although somewhat specific to this application, but the second and third points are generally applicable to any swarm technology. Fig. 3 shows two kind of USVs used.

Several sets of simulations were needed to be performed, with a list of the most important parameters. These simulations address and are referred to as: Diffusive Collection, USV Size vs. Quantity, and USV Density. As its scenarios, swarmships move through water at a speed of one unit per second and collect 90% of the algae that they pass over.

As previously mentioned, this simulation environment is a very complex system. These simulations only included the effect of algae diffusion, due to turbulence in the water. This work also started with a homogeneous algae distribution. In reality, algae is sometimes distributed homogeneously The inclusion of collision tracking, algae advection, and the possibility of a heterogeneous initial algae distribution are three areas targeted for the swarmships system for future work as shown on Fig. 4.

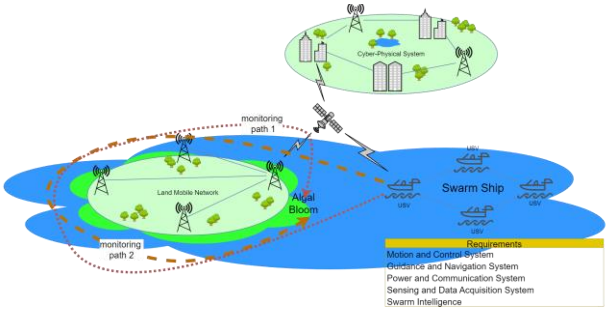


Fig. 4. Illustration of swarmship technology for harmful algae monitoring.

VI. Conclusion

From this point of view, we concluded that swarm ship technology has become an important potential implementation for near real time in situ monitoring compared to other decision making method such as laboratory examination or remote sensing-based results. The results of this review open the opportunity to realization of swarm ship technology in cyber physical system for monitoring algal bloom in specific area near real time efficiently.

Acknowledgment

This research was supported by Penelitian Disertasi Doktor (PDD), Contract No. 3793 /IT3. L1/PT.01.03/P/B/2022. The authors also would like to thank Telkom University for providing the laboratory for in-situ pre-experiments.

References

[1] A. Irawan, Qadar Hasani and H. Yuliyanto, "Fenomena Harmful Algal Blooms (HABs) di Pantai Ringgung Teluk Lampung, Pengaruhnya dengan Tingkat Kematian Ikan yang Dibudidayakan pada Karamba Jaring Apung," Jurnal Penelitian Pertanian Terapan, pp. 48-53, 2014.

[2] R. Aryawati, D. G. Bengen, T. Prartono and H. Zulkifli, "Harmful Algal in Banyuasin Coastal Waters, South Sumatera," Biosaintifika Journal of Biology & Biology Education, pp. 231-239, 2016.

[3] Q. Adnan, "Algal blooms in Jakarta Bay, Indonesia," Marine Coastal Eutrophication, pp. 809-818, 1992.

[4] H. Thoha, "Recent Harmful Algal blooms (HABs) Events in Indonesia," in WESTPAC Workshop on the Development of a Research Strategy for Harmful Algal Blooms, Nha Trang, Vietnam, 2016.

[5] G. R. Barokah, A. K. Putri and Gunawan, "KELIMPAHAN FITOPLANKTON PENYEBAB HAB (HARMFUL ALGAL BLOOM) DI PERAIRAN TELUK LAMPUNG PADA MUSIM BARAT DAN TIMUR," Jurnal Pascapanen dan Bioteknologi Kelautan dan Perikanan, pp. 115-126, 2017.

[6] Mursalin, R. Zulmi, M. D. Putra, L. D. W. Handayani and I. A. Nur, "Blooming fitoplankton di perairan Kepulauan Seribu," Jurnal Pengelolaan Lingkungan Berkelanjutan, pp. 652-667, 2021.

[7] A. S. Piranti, D. N. Wibowo and D. R. Rahayu, "Nutrient Determinant Factor of Causing Algal Bloom in Tropical Lake (Case Study in Telaga Menjer Wonosobo Indonesia)," Journal of Ecological Engineering, p. 156–165, 2021.

[8] A. R. Brown, M. Lilley, J. Shutler, C. Lowe, Y. Artioli, R. Torres, E. Berdalet and C. R. Tyler, "Assessing risks and mitigating impacts of harmful algal blooms on mariculture and marine fisheries," Reviews in Aquaculture, p. 1–26, 2019.

[9] C. A. Nugroho, A. Vidura, M. R. Rahman, M. Iqbal, M. G. A. Satria and I. Jaya, "Implementation of swarm intelligence algorithm on Autonomous Surface Vehicle (ASV)," in IOP Conference Series: Earth and Environmental Science, 2020.

[10] M. Dorigo, G. Theraulaz and V. Trianni, "Swarm robotics: Past, present, and future," in Proceeding of IEEE, 2021.

[11] A. R. Cheraghi, S. Shahzad and K. Graffi, "Past, Present, and Future of Swarm Robotics," Lecture Notes in Networks and Systems, pp. 190-233, 2022.

[12] P. G., F. Dias, M. C. Silva, G. P., R. Filho, P. A. Vargas, L. P. Cota and G. Pessin, "Swarm Robotics: A Perspective on the Latest Reviewed Concepts and Applications," Sensors, pp. 1-30, 2021.

[13] P. Bannur, P. Gujarathi, K. Jain and A. J. Kulkarni, "Application of Swarm Robotic System in a Dynamic Environment using Cohort Intelligence," Soft Computing Letters, 2020.

[14] S. Melanie, U. Martina, S. Micha and E. Wilfried, "Swarm Robotic Behaviors and Current Applications," Frontiers in Robotics and AI, p. 36, 2020.

[15] A. Toldrà, C. K. O'Sullivan and M. Campàs, "Detecting Harmful Algal Blooms with Isothermal Molecular Strategies," Trends in Biotechnology, 2019.

[16] H. J. van der Woerd, R. Pasterkamp, R. Blauw and L. Peperzak, "A harmful algal bloom warning system for the North Sea: a combination of remote sensing and computer models for algal growth and bloom

transport," in Proceedings of the Ocean Optics Conference, halifax, Canada, 2006.

[17] Emiyati, E. Parwati and S. Budhiman, "HARMFUL ALGAL BLOOM 2012 EVENT VERIFICATION IN LAMPUNG BAY USING RED TIDE DETECTION ON SPOT 4 IMAGE," International Journal of Remote Sensing and Earth Sciences, p. 1 – 8, 2017.

[18] P. R. Hill, A. Kumar, M. Temimi and D. R. Bull, "HABNet: Machine Learning, Remote Sensing-Based Detection of Harmful Algal Blooms," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, pp. 3229-3239, 2020.

[19] D. A. McPartlin, J. H. Loftus, A. S. Crawley, J. Silke, C. S. Murphy and R. J. O’Kennedy, "Biosensors for the monitoring of harmful algal blooms," Current Opinion in Biotechnology, p. 164–169, 2017.

[20] J. M. Giron-Sierra, A. Gheorghita, J. M. Riola and J. J. Díaz, "Swarms of Unmanned Vehicles for Area-Scan: Conceptual and Practical Control Aspects," in STO NATO Meeting Proceeding: Swarm-Centric Solution for Intelligent Sensor Networks, 2016.

[21] T. McClean, "Forbes," 2021. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2021/05/13/the-collective-power-of-swarm-intelligence-in-ai-and-robotics/?sh=7be81f14252f. [Accessed Sunday 11/14 2021].

[22] M. Brambilla, E. Ferrante, M. Birattari and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," Swarm Intelligence, vol. 7, p. 1–41, 2013.

[23] Schroeder, Adam M.. "Mitigating Harmful Algal Blooms using a Robot Swarm.", The University of Toledo, 2020.

Modeling of Organic Waste Classification as Raw Materials for Briquettes using Machine Learning Approach

Norbertus Tri Suswanto Saptadi[1], Ansar Suyuti[2], Amil Ahmad Ilham[3]*, Ingrid Nurtanio[4]
Department of Electrical Engineering, Universitas Hasanuddin, Gowa, Indonesia[1, 2]
Department of Informatics, Universitas Hasanuddin, Gowa, Indonesia[3, 4]
Department of Informatics, University of Atma Jaya Makassar, Makassar, Indonesia[1]

***Abstract*—The existence of organic waste must be utilized by the community so that it does not only end up in landfills but can also be processed into something constructive so that it is useful and has high economic value. Organic waste can be converted into raw materials to manufacture of biomass briquettes. Machine learning techniques were developed for technological applications, object detection, and categorization. Methods with artificial reasoning networks that use a number of algorithms, such as the Naive Bayes Classifier, will work together in determining and identifying certain characteristics in a digital data set. The manufacturing method goes through several processes with a waste classification model as a source of learning data. The image data is based on five types: coconut shells, sawdust, corn cobs, rice husks, and plant leaves. The research aims to identify and classify types of waste both organically and non-organically so that it will make it easier to sort waste. The results of testing the organic waste application from digital images have an accuracy rate of 97%. The model design carried out in training data is useful for producing a data model.**

Keywords—Classification; organic waste; raw material; machine learning

I. Introduction

Waste that has not been treated optimally can damage the environment and make the water not clear, thus disrupting life and health around human habitation [1]. Household waste has dominated in generating waste in the city of Makassar, with an average of 900 tonnes per day sent to the Antang Final Disposal Site [2]. The Head of the Makassar City Environment Service said that high public consumption during the COVID-19 period, both before and during the implementation of the Large Scale Social Restrictions (LSSR) to Enforcement of Restrictions on Community Activities (ERCA), did not significantly decrease in the presence of waste ranging from 850-950 tons. The community must make use of the organic waste that already exists so that it doesn't just wind up in landfills but is also transformed into something productive with significant economic value [3].

Utilizing waste produced by the community and businesses, urban development aims to create urban areas with greater productivity, capacity, efficiency, creativity, competitiveness, innovation, and use of information technology [4]. The application of technology, object detection, and classification was developed using machine learning methods. Methods with artificial reasoning networks that use a number of algorithms will work together in determining and identifying certain characteristics in a digital data set [5]. Machine and deep learning are programmed with capabilities to learn, digest, and classify data [6] and have good skills in computer vision by classifying objects in digital images [7][8].

A method with a specialized approach that uses particular electronic systems or devices to detect an object is needed because a concentration of waste in one location makes it difficult to understand the potential for new renewable energy in detail [9]. The increase in population, ways of consumption, and lifestyle of urban residents increase waste production and problems in big cities. Problems identified include the amount of waste generation, types of waste, and various waste characteristics [10]. To make waste classification easier, the research intends to detect and categorize different waste types, both organic and non-organic.

II. Related Work

One source of energy that could be produced is organic waste, which the government needs to handle properly [11] [12]. On the other hand, the demand for energy from communities and development organizations is rising [13], particularly for new and renewable sources of energy, which are becoming more and more limited in Indonesia [14]. With information on the different types of waste, landfill locations, transportation routes, and cleaning personnel, the geographic information system has been able to offer waste management services in every sub-district of the city of Makassar [15].

Governments and communities face a problem because they have not fairly and effectively utilised the potential of organic waste, especially in Makassar, Indonesia. To meet the energy needs of residents in a city or region, it can be converted into raw materials for briquettes [16][17]. This project's objective is to establish a prediction system data model for converting organic waste into a fuel that meets societal and industrial goals. An important scientific contribution is the use of detection methods on organic waste objects. Fourteen districts: Ujung Pandang, Bontoala, Makassar, Tamalanrea, Panakukkang, Wajo, Tallo, Mamajang, Manggala, Mariso, Rappocini, Biringkanaya, Ujung Tanah, and Tamalate, already exist inside the city of Makassar. This

*Corresponding Author.

case study is used to develop a system model based on the city's geographic characteristics [18].

III. Material and Method

A. Approach and Type of Research

The research approach uses quantitative methods. This type of research is based on experimental studies (results of literature studies), interviews, and direct observations of city environmental service officers, city government, and communities in the city of Makassar, South Sulawesi province, Indonesia [19].

Object detection is carried out on various sources of organic waste. After obtaining information about the source of the waste, it is then analyzed through digital image detection to determine the classification of the type of waste [20]. Classification is useful for determining raw materials in sorting and grouping for the purposes of effectively and efficiently making biomass briquettes [21][22].

B. Data Sources and Data Collection Techniques

To meet energy needs, it is essential to design a system that can provide data on the market and fuel availability. The amount of energy potential found in organic waste source materials is the anticipated piece of information. The design information in Fig. 1 provides an overview of the anticipated procedures for obtaining data on population numbers and needs [18][19].

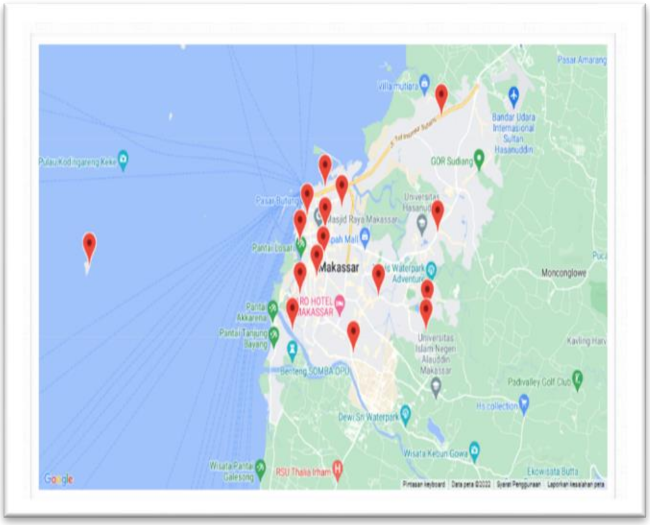
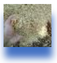


Fig. 1. Model for a waste location mapping system.

Organic waste is used to determine primary data. The Makassar City Environmental Service provided information on the need for secondary data by observing and speaking with staff members and locals on the origin of the organic waste collected.

The primary study data are the raw materials and specifications of organic waste utilized to make biomass briquettes, as compared to 5 (five) different types of organic waste sources. Organic waste can consist of coconut shells [23], sawdust [24], corncob [25], rice husks [26], and plant leaves [27]. The collected garbage data is made from a collection of digital image data in Table I. The predetermined image parameters are color, size (average), texture shape, reference mass, and manual mass (in the photo).

TABLE I. Dataset Organic Waste

No	Type	Image	Color	Size (average)	Shape and Texture	Mass Reference (Average)	Manual Mass (100%)
1	coconut shells		chocolate green	width: 3mm diameter: 95mm	circle variation	140g /unit	75g / half
2	sawdust		white yellow chocolate	/variation	/variation	/variation	40gr (8 oz)
3	corncob		yellow chocolate white gray	width: 21cm diameter: 47mm	tube beam	260g/unit (with corn)	260g/unit (with corn)
4	rice husks		yellow chocolate black	long: 9mm diameter 2.4mm	oval	/variation	50gr (8 oz)
5	plant leaves		green yellow chocolate	/ variation	leaf	/variation	40gr (16 oz) (vegetable leaves)

C. Research Mechanism

The object and model detection mechanism is based on the Machine Learning method approach by conducting a classification consisting of fully connected output with a Convolutional Neural Network (CNN) architecture [28][29].

D. Naive Bayes Classifier

One of the supervised learning algorithms in Fig. 2 that is used in machine learning for binary and multiclass classification processes is the Naive Bayes Classifier Algorithm in Fig. 3 [30][31].

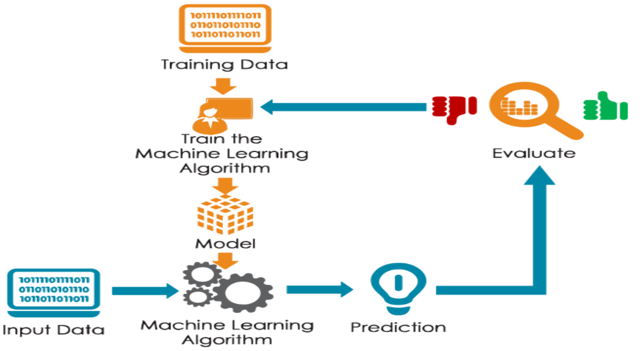

Training Data
Train the Machine Learning Algorithm
Evaluate
Model
Input Data
Machine Learning Algorithm
Prediction

Fig. 2. How machine learning works.

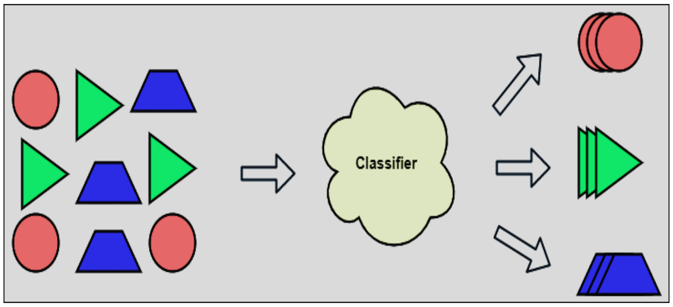


Fig. 3. Naive Bayes classifier method.

E. Formulation

Based on the Bayes principle, the Naive Bayes Classifier is a classification technique as shown in Fig. 4. It is generally agreed that the Naive Bayes Classifier outperforms a variety of other classification techniques. First, a very strong (naive) assumption of independence from each condition or event is the key feature of Naive Bayes. Second, it provides a straightforward, straightforward paradigm. Finally, large data sets can be used with the model [31].

$$P_{ro}(C_l|X_a) = \frac{P_{ro}(C_l)P_{ro}(X_a|C_l)}{P_{ro}(X_a)} \qquad (1)$$

Where:

X_a : attributes

C_s : class

$P_{ro}(C_s|X_a)$: probability of even C_s given X_a has occurred

$P_{ro}(X_a|C_s)$: probability of even X_a given C_s has occurred

$P_{ro}(C_s)$: probability of event C_s

$P_{ro}(X_a)$: probability of event X_a

X_a can be written as follow:

$$X_a = X_1, X_2, X_3, \dots, X_n \qquad (2)$$

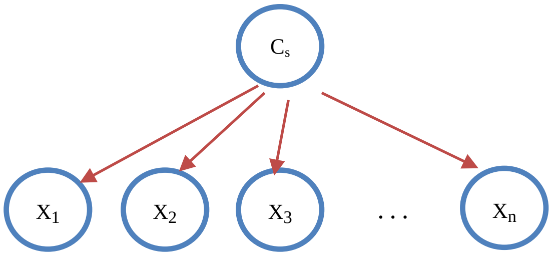


Fig. 4. Naïve Bayes.

F. Data Collection Techniques

The five types of organic waste were taken on a bright and plain background so that the various waste objects could still be seen [32]. Image data creation is carried out in detail by taking pictures of each type of waste [33]. Taking pictures of several combinations of one object with another object [34]. Images are photographed manually using smartphones and digital cameras specifically to get the best image and resolution.

G. Hardware Requirement

Specifications for some of the equipment used in the study include laptop Intel® CoreTM i5-5200U CPU @ 2.20GHz, Windows 10, Memory 8 GB, and AMD RadeonTM Graphics R5 M330. With the OPPO A5S smartphone, which features dual cameras with a configuration of 13 MP (wide), f/2.2, and 2 MP (depth), as well as an AF feature, LED flash, panoramic, HDR, and video 1080p@30fps, you may take images of five different forms of organic waste [35][36].

H. Preprocessing Data

Preprocessing data is a step that includes data collecting, data labeling, and data augmentation operations.

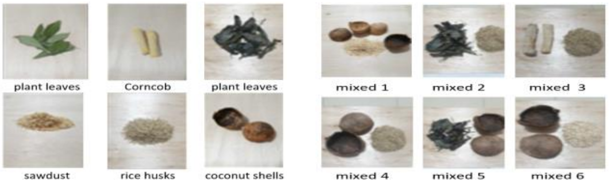
1) Data collection: Three thousand seventy-four photos totaled the information that was gathered and created. Organic waste can consist of coconut shells (cs) [23], sawdust (sd) [24], corncob (cc) [25], rice husk (rh) [26] and plant leaves (pl) [15]. The amount of information contained on the types of organic waste is shown in Table II.

TABLE II. DATASET ORGANIC WASTE

No	Waste Type	Digital Image
1	Coconut shells	686
2	Sawdust	717
3	Corncob	449
4	Rice husk	513
5	Plant leaves	709
Total		3.074

After data collection then proceed with the data preprocessing process. This process is carried out by dividing the pixel image into sizes with low, medium, and high criteria. Digital images will be used as training data to make them more effective in identifying types of organic waste [37].

Images of the five types of organic waste were taken against a bright and plain background to ensure these various types of waste objects could still be seen. Image data is collected by taking pictures of each kind of waste with a certain combination model as shown in Fig. 5 [38].



(a) (b)

Fig. 5. Preprocessing data and type of waste (a) One type of waste (b) More than one type of waste.

2) Labeling: The data labeling process aims in Fig. 6 to give the discarded image data that was taken a name or label so that it can be quickly identified. A train folder and a test folder are the components of creating a folder. At the same time, the test folder is used for running tests and validating data in the training process, and the train folder stores data that is processed throughout the learning process. It is separated into five different categories of organic waste for data labeling.

3) Augmented data: The technical stage in data augmentation is useful for augmenting existing training data using the TensorFlow library. This process is carried out by naming the image data taken to be neat, structured, and orderly.

The data augmentation process seeks to modify image data using the Adobe Photoshop application. Modifications are made based on the type of organic waste by combining many image data into one image data. Efforts were made by combining ten organic waste image data into one digital image data. For optimization purposes, resize the organic waste image data from the original size of 5184 x 3456 pixels to 224 x 224 pixels.

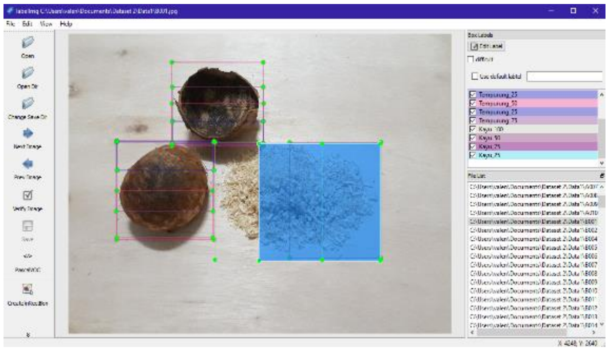


Fig. 6. Object size and mass predictive labeling.

I. Data Classification

There are 2473 training data and 293 digital photo test data as shown in Table III for five types of organic waste.

The combination of waste images in Table IV collected in combination A for five types in one image, combination B for three types in one image, and combination C for two types of waste are all in a folder. The other folders provide classification information for each of the five different forms of waste. The categorization that follows is based on the folder that was made.

TABLE III. ORGANIC WASTE

No	Waste Type	Train	Test
1	coconut shells (cs)	548	76
2	sawdust (sd)	589	55
3	corncob (cc)	364	45
4	rice husk (rh)	414	47
5	plant leaves (pl)	558	70
Total		2.473	293

TABLE IV. DATA COMBINATION

No	Combination of Organic Waste	Number of Combinations
1	cs, sd, cc, rh, pl (A)	5 pcs
2	cs, sd, rh (B)	3 pcs
3	cs, sd (C)	2 pcs
4	cs, rh (C)	2 pcs
5	cs, pl (C)	2 pcs
6	sd, pl (C)	2 pcs
7	rh, pl (C)	2 pcs
8	rh, cc (C)	2 pcs
9	rh, sd (C)	2 pcs
10	cs (D)	1 pcs
11	sd (D)	1 pcs
12	cc (D)	1 pcs
13	rh (D)	1 pcs
14	pl (D)	1 pcs
Total		23 pcs

Undoubtedly, a wide range of related factors will affect where and how organic waste is obtained. Table V lists the influential elements that must be calculated and considered.

TABLE V. INFLUENCE FACTORS

No	Factors	Influence
1	Photo Capture Distance	size
2	Object Location (x, y on photo)	accuracy
3	Weather/Temperature	texture
4	Lighting	coloring
5	Place/Environment	area/background

J. Data Modeling

The model is created using a number of procedures with the waste type classification model serving as a source of learning data. Five categories are utilized to organize the image data. A data model can be created by doing model design using training data. An illustration of the proposed data modeling flowchart is shown in Fig. 7.

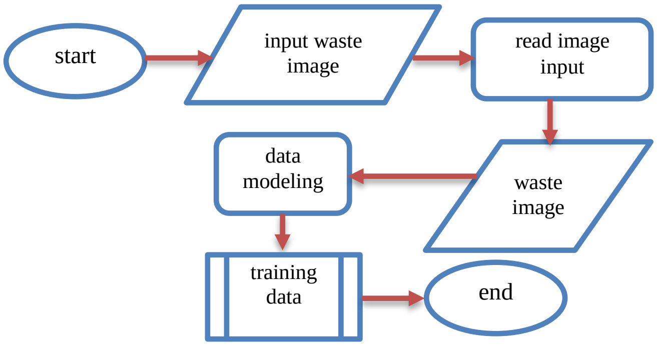


Fig. 7. Modeling flowchart.

K. Evaluation Method

A confusion Matrix is an evaluation method that can be used to calculate the classification process's performance or level of correctness. Create a table containing four unique combinations of the expected and actual values. In the Confusion Matrix, the classification process outcomes are denoted by four terms: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The equation that follows is created.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$\text{F1} = \frac{2\times Recall \times Precision}{Recall + Precision} \quad (6)$$

IV. RESULTS AND DISCUSSION

To produce a good model requires the support of methods, algorithms and organic waste datasets that are relevant and sufficient in number.

A. Modeling CNN

They use two CNN models from scratch and CNN with ResNet transfer learning. The design of the CNN model is carried out with training to produce a model (classifier rules) that classifies waste sources as raw materials for making biomass briquettes. The proposed CNN model is in the form of parameters connected to a vector (flatten) to enter the fully connected layer. After that, enter the dense layer, which reduces the size to 1024 outputs and then reduces them to 256 outputs.

The subsequent stage moves into the last layer of classification, which employs the softmax function to generate the number of classes corresponding to the category in the data. It then moves on to the stage of the image prediction process. The waste detection procedure is as follows.

```
data = np.ndarray(shape=1, 224, 224, 3), dtype=np.float32)
def entry(input_name_img, input_id):
image = Image.open(r"c:\potential\image/"+input_name_img)
size = (224, 224)
…
labels = ['coconut shell', 'sawdust', 'corn cobs', 'rice husk', 'plant leaves']
prediction = model.predict(data)
# print prediction
detection = str(labels[np.argmax(prediction)])
sql = "UPDATE image SET desc = %s WHERE id_image = %s"
val = (detektion, input_id)
cursor.execute(sql, val)
conn.commit()
input_nama_img = str(query_dict['img'])[2:-2]
input_id = str(query_dict['id_c'])[2:-2]
print(content-type: text/html\n\n")
entry(input name img, input id)
```

B. UML Interface Design

There are use cases that are useful for knowing the activity of the relationship between the user and the system in Fig. 8.

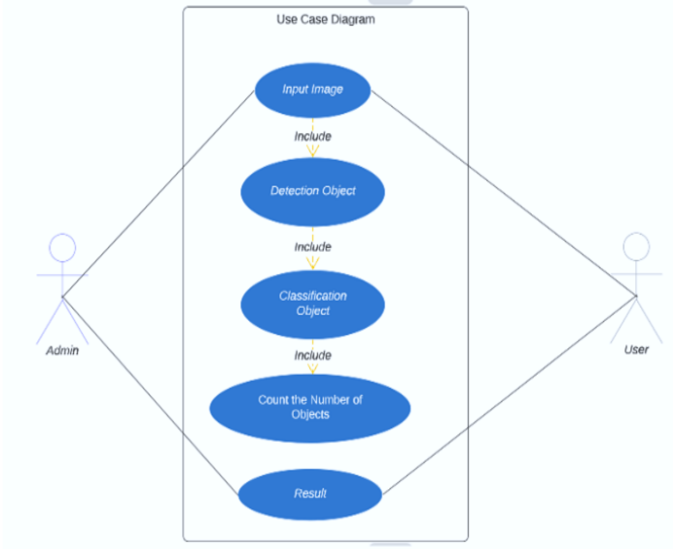


Fig. 8. Use case diagram.

Class diagram Fig. 9 illustrates the structure of class formation in the design of organic waste detection applications.

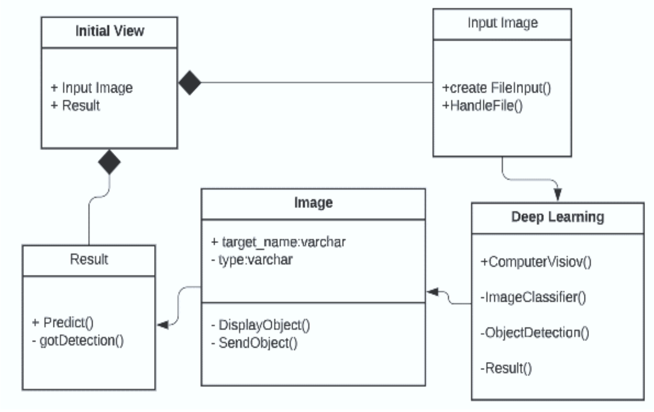


Fig. 9. Data training class diagrams.

Fig. 10 training data according to predefined batch values for all trained data. The results of the weight values are in the form of a JSON file that is used to help classify organic waste.

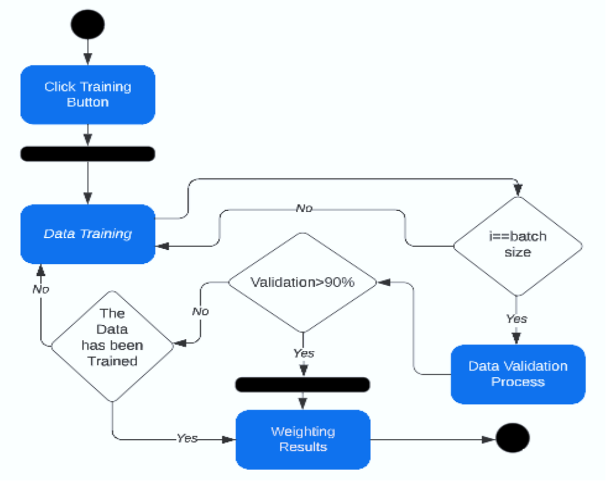
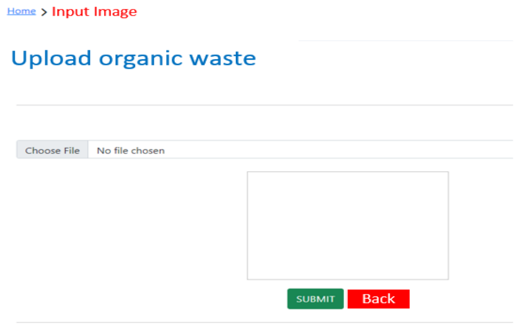


Fig. 10. Data training activity diagrams.

C. Application Interface Design

Input objects are needed by taking pictures in the system to carry out the process of predicting waste images. The data will be executed according to the program that can be detected. Fig. 11 displays the image that will be displayed after pressing the submit button in Fig. 12.


Home > Input Image

Upload organic waste

Fig. 11. Image input display on the application.

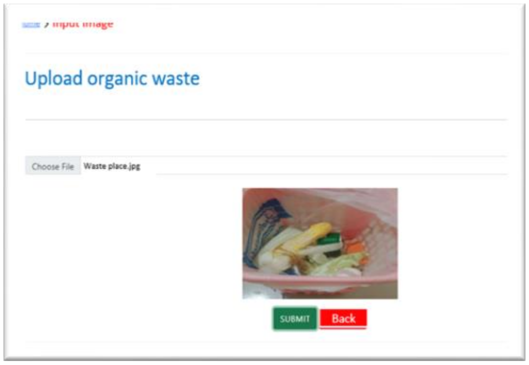


Fig. 12. Image upload view.

D. Training

The stage begins with collecting data used as training data. The image dataset for training purposes consists of 800 coconut shell waste, 750 sawdust, 750 corn cobs, 700 rice husks, and 700 plant leaves. The modeling phase divides the initial dataset into training and test data. The following are the stages of data modeling carried out in the study.

Modeling Script 1: Test data and validation make up each of the two sections of the test data with

```
test_size=0.20, random_state=300)
test_size=0.5, random_state=100)
```

Modeling Script 2: Merging data into each data frame.

```
train size 3359
val size 420
test size 420
```

Modeling Script 3: Organizing test, train, and validation data using the shut-in module.

```
datasource_path = “waste/”
dataset_path = “dataset2/”
```

Modeling Script 4: The process determines how many epochs to use.

```
#Define Input Parameters
Dim = (224, 224)
# dim = (456, 456)
```

Modeling Script 5: The data transformation process used in the image is transformed into image augmentation.

```
rescale=1. / 255, shear_range=0,2, zoom_range=0,2, horizontal_flip=True)
val_datagen
test_datagen
```

Modeling Script 6: Defines the origin directory by source files for the training phase.

```
train_generator = from_directory(‘dataset1/train/’ …,)
val_generator = from_directory(‘dataset1/validation/’ …,)
test_generator = test_datagen.flow_from_directory(‘dataset1/test/’,
num_class = tes_generator.num_classes
labels = train_generator.class_indices.key()
```

Modeling Script 7: Create tf.data to provide high compatibility for TensorFlow.

```
train_data = tf_data_generator(train_generator, input_shape)
test_data = tf_data_generator(test_generator, input_shape)
val_data = tf_data_generator(val_generator, input_shape)
```

Model Script 8: Sequential process for activation.

```
model_Sequential()
model.add(Conv2D(128, (3, 3), padding=’same;, input_shape=input_shape))
model.add(Activation(‘relu’))
model.add(conv2D(32, (3, 3)))
model.add(Activation(‘relu’))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
…
model.add(Flatten())
model.add(Dense(512))
model.add(Activation(‘relu’))
model.add(Dropout(0.5))
model.add(Dense(num_class))
model.add(Activation(‘softmax’))
# Compile the model
print(‘Compiling Model…….’)
model.compile(Optimizer=’adam’, loss=’categorical_crossentropy’,
metrics=[‘accuracy’])
```

Modeling Script 9: Adding a top layer defined set false to the base model.

```
predictions = layer.Dense(num_class, activation=”softmax”)(x)
model = Model(inputs=base_model.input, outputs=predictions)
```

Modeling Script 10. Preparing the model to be ready for the training process.

```
# Compile the model
…
metrics=[‘Accuracy’])
from efficientnet.tfkeras import EfficientNetB1
# get base models
…
```

Modeling Script 11. Creating a machine learning process model in considering class patterns.

```
EPOCH = 2
…
Epoch 1/2
210/210 [=======] – 528s 2s/step – loss: 0.2357 – accuracy: 0,9357 –
val_loss: 0.0439 – val_accuracy: 0.9929
Epoch 2/2
210/210 [=======] – 466s 2s/step – loss: 0.1083 – accuracy: 0,9687 –
val_loss: 0.0244 – val_accuracy: 0.9952
```

Modeling Script 12. The training results get a loss of 0.2 and 0.1 and an accuracy of 0.9.

```
history.history[‘loss’]
[0.2356509417295456, 0,10834924131631851]
history.history[‘accuracy’]
[0.9356759786605835, 0,96873140335508301]
```

model.save (save_model_path, include_optimizer=False) is useful so that TensorFlow does not save the state of the optimizer at the last time it was saved but to save storage media and simplify the deployment process in Fig. 13.

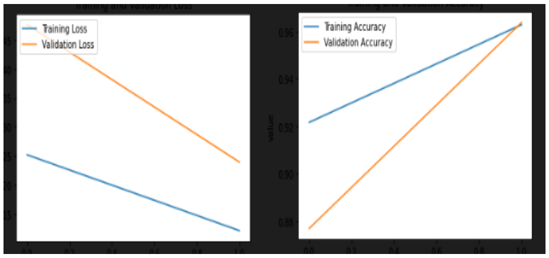


Fig. 13. Training data.

After training is done, then be plotted with extra training. Then the model results will be obtained, which can be saved to detect objects.

```
MODEL_BASE_PATH = “image_model”
PROJECT_NAME = “waste organic”
SAVE_MODEL_NAME = “image_model.h5”
```

E. Testing

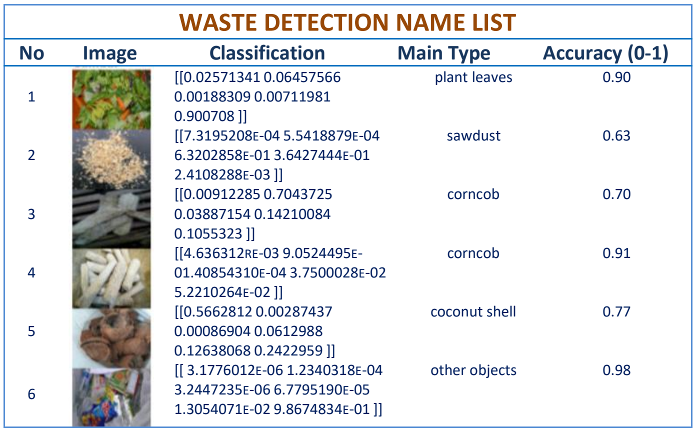
After determining the training data, test data is needed which consists of 120 coconut shells, 113 sawdust, 113 corncobs, 105 rice husks, and 105 plant leaves. Before loading the model, it is necessary to define parameters and label the five types of waste. After that, a function is created for pre-processing.

```
model = load_model(‘\medium_project/keras_model.h5’)
data = np.ndarray(shape=1, 224, 224, 3), dtype=np.float32)
…
prediction = model.predict(data)
print(prediction)
```

The script is an image prediction process. Prediction results are displayed in an array based on index 0,1,2,3,4,5. The process of detecting according to five characteristics of waste such as coconut shell, sawdust, corncobs, rice husks, and plant leaves.

```
1/1 [========================] – 0s 415ms/step
[[1.7525636e-02  3.1258736e-03  4.1368250e-03  5.6559354e-04  9.7464603e-01]]
```

Based on the picture in the script for detecting organic waste, which is the result of waste detection, the results of the prediction of waste produce a value of 9.7 or 97% accuracy in detecting the type of waste. Fig. 14 below is the result of testing the image input on object detection.



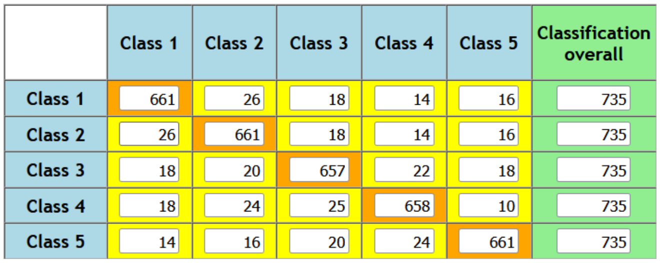
WASTE DETECTION NAME LIST

No	Image	Classification	Main Type	Accuracy (0-1)
1		[[0.02571341 0.06457566 0.00188309 0.00711981 0.900708]]	plant leaves	0.90
2		[[7.3195208E-04 5.5418879E-04 6.3202858E-01 3.6427444E-01 2.4108288E-03]]	sawdust	0.63
3		[[0.00912285 0.7043725 0.03887154 0.14210084 0.1055323]]	corncob	0.70
4		[[4.636312RE-03 9.0524495E-01.40854310E-04 3.7500028E-02 5.2210264E-02]]	corncob	0.91
5		[[0.5662812 0.00287437 0.00086904 0.0612988 0.12638068 0.2422959]]	coconut shell	0.77
6		[[3.1776012E-06 1.2340318E-04 3.2447235E-06 6.7795190E-05 1.3054071E-02 9.8674834E-01]]	other objects	0.98

Fig. 14. Waste detection name list.

F. Evaluation

Class 1 coconut shell, Class 2 sawdust, Class 3 corncobs, Class 4 rice husks, and Class 5 plant leaves are among the 735 picture data used for each class. The column shows the actual type class, and the row shows the type being tested. The first column of the first row displays the results that are in Class 1 coconut shell. The coconut shell image tested reads as TP (True Positive) of 661, so the number of images is read exactly according to the type of organic waste class.Likewise for other classes as shown in Fig. 15.



	Class 1	Class 2	Class 3	Class 4	Class 5	Classification overall
Class 1	661	26	18	14	16	735
Class 2	26	661	18	14	16	735
Class 3	18	20	657	22	18	735
Class 4	18	24	25	658	10	735
Class 5	14	16	20	24	661	735

Fig. 15. Confusion matrix.

Testing accuracy, precision, and recall by the confusion matrix using test data that is not sourced from datasets show an accuracy value of up to 73%. This research has produced an application that is able to detect objects properly. Through object detection, it will be known that the image is a potential organic waste which is a raw material for processing biomass briquettes. Even though the results in detecting objects still have a value with a percentage that has not reached a value of 70%, this is a relatively good approach to be used as an alternative model in selecting objects that will be the raw material for making briquettes.

Detection of objects in the form of powder is still relatively difficult to distinguish from similar objects considering that objects consist of small and separate parts. Comparison with large and unified objects will certainly produce a better accuracy value. To overcome this, further research is needed on object detection with the segmentation method.

V. Conclusion

Research on organic waste has produced object detection, namely: the results of identifying the type of organic waste by detecting digital image objects can predict the type of waste contained in the inserted image. The results of testing the organic waste application from digital images have an accuracy rate of 97%.

Acknowledgment

We would want to express our gratitude to the Makassar City Environmental Department as well as to the general public and business owners.

References

[1] A. C. Malina, Suhasman, A. Muchtar, and Sulfahri, “Environmental Assessment of Waste Sorting Sites in Makassar City,” Journal of Innovation and Public Service, vol. 1, no. 1, pp. 14–27, 2017.

[2] S. Asiri and M. Manaf, “The Effect of the Existence of the Tamangapa Landfill on Changes in the Use of the Surrounding Space,” Plano Madani, vol. 8, no. 2, pp. 138–146, 2019.

[3] I. M. Harjanti and P. Anggraini, “Waste Management at the Jatibarang Final Disposal Site, Semarang City,” Jurnal Planologi, vol. 17, no. 2, p. 185, 2020, doi: 10.30659/jpsa.v17i2.9943.

[4] M. Huda and E. B. Santoso, “Development of Regency/City Regional Competitiveness in East Java Province based on Regional Potential,” Journal of Engineering POMITS, vol. 3, no. 2, pp. 81–86, 2014.

[5] S. Cheon, H. Lee, C. O. Kim, and S. H. Lee, “Convolutional Neural Network for Wafer Surface Defect Classification and the Detection of Unknown Defect Class,” IEEE Transactions on Semiconductor Manufacturing, vol. 32, no. 2, pp. 163–170, 2019, doi: 10.1109/TSM.2019.2902657.

[6] F. Elmaz, B. Büyükçakır, Ö. Yücel, and A. Y. Mutlu, “Classification of solid fuels with machine learning,” Fuel, vol. 266, no. July 2019, 2020, doi: 10.1016/j.fuel.2020.117066.

[7] V. Wiley and T. Lucas, “Computer Vision and Image Processing: A Paper Review,” International Journal of Artificial Intelligence Research, vol. 2, no. 1, p. 22, 2018, doi: 10.29099/ijair.v2i1.42.

[8] H. Kim, J. Kim, and H. Jung, “Convolutional neural network based image processing system,” Journal of Information and Communication Convergence Engineering, vol. 16, no. 3, pp. 160–165, 2018, doi: 10.6109/jicce.2018.16.3.160.

[9] Y. Han, T. Jiang, Y. Ma, and C. Xu, “Pretraining Convolutional Neural Networks for Image-Based Vehicle Classification,” Advances in Multimedia, vol. 2018, 2018, doi: 10.1155/2018/3138278.

[10] A. Kahfi, “Overview of Waste Management,” Jurisprudentie: Department of Law, Faculty of Sharia and Law, vol. 4, no. 1, p. 12, 2017, doi: 10.24252/jurisprudentie.v4i1.3661.

[11] S. Soeprijanto, A. R. Fatullah, S. Agustina, D. F. Amalia, and A. A. Kaisar, “Biogas Production from Vegetables and Fruit Wastes Using Anaerobic Floating Bioreactor,” Eksergi, vol. 17, no. 2, p. 99, 2020, doi: 10.31315/e.v17i2.3733.

[12] S. Ma’arif and W. Wardoyo, “Potential of Electric Energy from Waste in Kaliurang Tourism Area, Sleman, Special Region of Yogyakarta,” Conserve: Journal of Energy and Environmental Studies, vol. 4, no. 1, pp. 1–8, 2020, [Online]. Available: https://ejournal.up45.ac.id/index.php/cjees/article/view/718.

[13] S. I. Faizah and U. A. Husaeni, “Development of consumption and supplying energy in Indonesia’s economy,” International Journal of Energy Economics and Policy, vol. 8, no. 6, pp. 313–321, 2018, doi: 10.32479/ijeep.6926.

[14] S. Subagyo, J. P. Moh. Yanuar, P. N. Bambang, and A. Saleh, “Substitution of energy needs with renewable energy sources,” IOP Conf Ser Earth Environ Sci, vol. 927, no. 1, p. 012032, 2021, doi: 10.1088/1755-1315/927/1/012032.

[15] N. Saptadi, P. Chyan, and A. C. Pratama, “Geographic Information System for Waste Management for the Development of Smart City Governance Geographic Information System for Waste Management for the Development of Smart City Governance,” 2020, doi: 10.1088/1757-899X/854/1/012040.

[16] T. Chen, H. Jia, S. Zhang, X. Sun, Y. Song, and H. Yuan, “Optimization of cold pressing process parameters of chopped corn straws for fuel,” Energies (Basel), 2020, [Online]. Available: https://www.mdpi.com/1996-1073/13/3/652.

[17] P. Thomas, N. Soren, N. P. Rumjit, and J. G. James, “Biomass resources and potential of anaerobic digestion in Indian scenario,” Renewable and Sustainable Energy Reviews, 2017, doi: 10.1016/j.rser.2017.04.053.

[18] N. T. S. Saptadi, P. Chyan, and A. C. Pratama, “Analysis and Design of Waste Management System Using the Spiral Model Towards Smart Cities,” Sisforma, vol. 6, no. 2, p. 41, 2020, doi: 10.24167/sisforma.v6i2.2313.

[19] N. Saptadi, P. Chyan, and A. C. Pratama, “Geographic Information System for Waste Management for the Development of Smart City Governance,” IOP Conf Ser Mater Sci Eng, vol. 854, no. 1, 2020, doi: 10.1088/1757-899X/854/1/012040.

[20] A. Chailes, A. Hermawan, and D. Kurnaedi, “Application of the Data Mining Method to Determine Purchasing Patterns Using the Apriori and Fp-Growth Algorithms at Mukara Stores,” Journal of Algor, vol. 1, no. 2, pp. 1–8, 2020.

[21] P. M. Dass, I. I. Nkafamiya, B. A. Thliza, and J. I. Joseph, “Analysis of Biomass and Cotton Stalk Charcoal Briquettes Produced from Biu, Nganzai, and Zabarmari in Borno State using a Locally Fabricated Briquetting Machine,” International Quarterly Scientific Journal, 2019.

[22] J. Zhang, D. Zheng, K. Wu, and X. Zhang, “The optimum conditions for preparing briquette made from millet bran using Generalized Distance Function,” Renewable Energy, 2019.

[23] R. P. Dewi and M. Kholik, “The effect of adhesive concentration variation on the characteristics of briquettes,” Journal of Physics: Conference Series, vol. 1517, no. 1, 2020, doi: 10.1088/1742-6596/1517/1/012007.

[24] M. Thabuot, T. Pagketanang, K. Panyacharoen, P. Mongkut, and P. Wongwicha, “Effect of Applied Pressure and Binder Proportion on the Fuel Properties of Holey Bio-Briquettes,” Energy Procedia, vol. 79, pp. 890–895, 2015, doi: 10.1016/j.egypro.2015.11.583.

[25] L. Sulistyaningkarti and B. Utami, “Making Charcoal Briquettes from Corncobs Organic Waste Using Variation of Type and Percentage of Adhesives,” Journal of Chemistry and Chemistry Education, vol. 2, no. 1, p. 43, 2017, doi: 10.20961/jkpk.v2i1.8518.

[26] P. M. Dass, I. I. Nkafamiya, B. A. Thliza, and J. I. Joseph, “Analysis of Biomass and Cotton Stalk Charcoal Briquettes Produced from Biu, Nganzai, and Zabarmari in Borno State using a Locally Fabricated Briquetting Machine,” International Journal of Progressive Sciences and Technologies (IJPSAT), vol. Vol.16, no. 2, p. PP. 83-91, 2019.

[27] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, “Cucumber leaf disease identification with global pooling dilated convolutional neural network,” Comput Electron Agric, vol. 162, no. February, pp. 422–430, 2019, doi: 10.1016/j.compag.2019.03.012.

[28] E. N. Arrofiqoh and H. Harintaka, “Implementation of Convolutional Neural Network Method for Plant Classification in High Resolution Image,” Geomatika, vol. 24, no. 2, p. 61, 2018, doi: 10.24895/jig.2018.24-2.810.

[29] V. Venkatesh, N. Yallappa, S. U. Hegde, and S. R. Stalin, “Fine-Tuned MobileNet Classifier for Classification of Strawberry and Cherry Fruit Types,” Journal of Computer Science, vol. 17, no. 1, pp. 44–54, 2021, doi: 10.3844/jcssp.2021.44.54.

[30] R. Adrian, M. A. J. S. Perdana, A. Asroni, and S. Riyadi, “Applying the Naive Bayes Algorithm to Predict the Student Final Grade,” Emerging Information Science and Technology, vol. 1, no. 2, pp. 49–57, 2020, doi: 10.18196/eist.127.

[31] N. Salmi and Z. Rustam, “Naïve Bayes Classifier Models for Predicting the Colon Cancer,” IOP Conference Series: Materials Science and Engineering, vol. 546, no. 5, 2019, doi: 10.1088/1757-899X/546/5/052068.

[32] N. Tri, S. Saptadi, A. Suyuti, A. Ahmad, and I. Nurtanio, “Prediction System Data Model In Obtaining Energy Potential of Biomass Briquette Compared to Other,” Journal of Southwest Jiaotong University, vol. 57, no. 5, 2022, doi: https://doi.org/10.35741/issn.0258-2724.57.5.38.

[33] P. Hwangdee, C. Jansiri, and S. Sudajan, “Physical characteristics and energy content of biomass charcoal powder,” Journal of Renewable Energy and Environment, 2021.

[34] H. C. Law, L. M. Gan, and H. L. Gan, “Experimental study on the mechanical properties of biomass briquettes from different agricultural residues combination,” MATEC Web of Conferences, 2018.

[35] N. T. S. Saptadi, P. Chyan, and V. M. Widjaja, “Model Design of Organic Waste Classification into Raw Materials for Biomass Briquettes Using Deep Learning Methods,” JIKO (Journal Informatics dan Computer), vol. 6, no. 2, p. 160, 2022, doi: 10.26798/jiko.v6i2.559.

[36] R. Razuan, K. N. Finney, Q. Chen, and V. N. Sharifi, “Pelletised fuel production from palm kernel cake,” Fuel Processing Technology, 2011.

[37] W. You, C. Shen, X. Guo, X. Jiang, J. Shi, and Z. Zhu, “A hybrid technique based on convolutional neural network and support vector regression for intelligent diagnosis of rotating machinery,” Advances in Mechanical Engineering, vol. 9, no. 6, pp. 1–17, 2017, doi: 10.1177/1687814017704146.

[38] A. M. Torkashvand, A. Ahmadi, and N. L. Nikravesh, “Prediction of kiwifruit firmness using fruit mineral nutrient concentration by artificial neural network (ANN) and multiple linear regressions (MLR),” J Integr Agric, vol. 16, no. 7, pp. 1634–1644, 2017, doi: 10.1016/S2095-3119(16)61546-0.

AUTHORS' PROFILE

Norbertus Tri Suswanto Saptadi was born on June 07 1975 in Cirebon, West Java, Indonesia. He is currently a doctoral student in Electrical Engineering (Informatics), Hasanuddin University, Indonesia. He accomplished Bachelor Degree of Informatics, University of Digital Technology Indonesia in 1998. Master of Management, Hasanuddin University in 2004. Master of Information Technology, Gadjah Mada University in 2007. Engineer Professional Education, Hasanuddin University in 2020. His research interests include artificial intelligence, data science, software engineering, and IT governance. He is member of the Indonesian National Resilience Institute (Lemhannas RI).

Ansar Suyuti was born on December 31 1967 in Sidrap, South Sulawesi, Indonesia. He received a bachelor's degree in 1991, a master's degree in 2002, and a doctorate in 2013 from Universitas Hasanuddin in Makassar, Indonesia. His primary topic of study is Electrical Power Engineering. He is a professor at Universitas Hasanuddin's Department of Electrical Engineering. He is the current head of the Power System Distribution Laboratory. His research interests include distributed generation planning, sensor networks, and environmental engineering, and he has produced numerous articles in electrical power engineering. Prof. Ir. Ansar Suyuti is a member of the Technical Committee for IPU.ASEAN. Eng is also a member of the Association of Electrical Power Engineering Experts and is an assessor for the National Accreditation Bureau for Higher Education. He is the Rector of Institut Teknologi BJ Habibie, Indonesia.

Amil Ahmad Ilham was born on October 10 1973 in Bulu, South Sulawesi, Indonesia. He accomplished his Bachelor Degree of Electrical Engineering, Hasanuddin University, Indonesia in 1997, Master of Information Technology, The University of Newcastle, Australia in 2003 and Doctor of Informatics from Kyushu University in 2011. He is currently a lecturer at Hasanuddin University, Department of Informatics, since 1998. His research interests are Information Systems and Data Analytics. He is the Vice Dean for Academic and Student Affairs of Faculty of Engineering, Hasanuddin University, Indonesia.

Ingrid Nurtanio was born on August 13 1961 in Makassar, South Sulawesi, Indonesia. Her accomplished Bachelor Degree of Electrical Engineering, Hasanuddin University in 1986, Master of Electrical Engineering, Hasanuddin University in 2002, Doctor from Institut Teknologi Sepuluh Nopember in 2013. Her is currently a lecturer at the Hasanuddin University of Informatics, Indonesia since 1988. Her research interests are Artificial Intelligence.

Predicting Hypertension using Machine Learning: A Case Study at Petra University

Yasmin Sakka[1], Dina Qarashai[2], Ahmad Altarawneh[3]
Faculty of Administrative and Financial Sciences-Management Information Systems Department,
University of Petra, Amman, Jordan[1, 3]
Healthcare Center, University of Petra, Amman, Jordan[2]

***Abstract*—Hypertension is a key cardiovascular disease risk factor (CVD). Identifying these high-risk individuals is crucial since it would save time and money before using any sophisticated, invasive, or costly diagnostic technique. This endeavour may be accomplished in part with the use of modern machine learning techniques. Specifically, a prediction model may be created based on several easily-obtained, non-invasive, and inexpensive indicator characteristics of high-risk individuals. This research is an effort to forecast hypertension risks based on Petra University's population. This case study was done between 2019 and 2020 at Petra University. Using hospital-visited patients' medical records, the gathered data was used to develop a model. The research comprised a comprehensive dataset of 31500 patients, comprising 12658 hypertension cases and 18842 non-hypertensive cases. SMOTE was used as a dataset for the categorization of hypertension. The SMOTE-k-nearest neighbour prediction model performs exceptionally well, as evidenced by its excellent performance (83.9% classification accuracy, 85.1% specificity, 83.3% sensitivity, and 89.6% AUC) when compared to other classifiers using 10-fold cross-validation with full features and no oversampling on the hypertension dataset. The data extracted from Petra University Health Center is considered to be very helpful for ML and is availed to produce a decision tree to identify the data related to hypertension.**

Keywords—Hypertension; machine learning; medical records; sensitivity; specificity

I. Introduction

There are 8.5 million fatalities worldwide attributed to hypertension, making it the leading cause of cardiovascular disease. Among these, 88% occur in low- and middle-income nations [1]. Access to unhealthy foods, sedentary lifestyles, and rural-urban migration are major contributors to the rising rate of hypertension in South Asia [2, 3]. The identification, treatment, and management of hypertension are similarly lowest in South Asia, and there has been little progress in these areas over the previous three decades [1]. Due to low levels of screening awareness, many cases of hypertension in South Asia go unreported [4]. Heart disease, stroke, renal failure, and premature death may all result from hypertension, although they are often avoidable with early diagnosis and treatment.

Physical inactivity, low levels of knowledge, smoking, an unhealthy diet, a lack of access to healthcare, and the high cost of drugs all play roles in explaining why hypertension is more common in South Asia [4-6]. However, most of the studies utilized unreliable methods to assess risk, had insufficient sample sizes and failed to adequately reflect the general population. The Framingham Risk Score for predicting coronary heart disease [7] and the American College of Cardiology/American Heart Association (ACC/AHA) Pooled Cohort Equations Risk Calculator are just two examples of risk prediction models that have been successfully used to identify and stratify patients according to risk factors and initiate preventative therapies [8].

When it comes to the creation of risk stratification tools for the diagnosis of cardiovascular illnesses, machine learning (ML) methods have recently been demonstrated to outperform classic statistical approaches [9,10,11]. The field of computer science known as "machine learning" allows computers to "learn" independently of explicit programming and handle massive datasets with complicated relationships. In contrast to more conventional statistical approaches, ML algorithms are not dependent on causal inference; nonetheless, this does not make them any less important for assessing causal effects in observational research. When compared to more conventional statistical methods, ML eliminates bias, the automated handling of missing variables with little changes to the original data, the mitigation of confounding factors, and the balancing of data [10]. More importantly, traditional statistical methods often fail when used in "big data" situations, whereas ML techniques succeed. Therefore, machine learning techniques may be used to create automated systems for illness prediction, decision support, and estimating hypertension prevalence in a community [10].

A dearth of research integrating socio-demographic and clinical data with signal processing, which might improve model performance, was observed in a recent review of ML approaches in hypertension identification [12]. ML algorithms were employed in prior work to automatically classify individuals with hypertension based on their unique phenotypes, but this analysis lacked socio-demographic information [13]. In a separate study, researchers in India used information gathered by community health workers from 2,278 individuals to create ML risk classification algorithms for diabetes and hypertension [14]. ML was employed in two different studies in China to analyze EHRs for signs of hypertension [15].

In the previous studies related to the machine learning models, the success lies in the complex pattern which can predict the data which is not observed in any other model. The concept of machine learning develops the power of interpretability by introducing the strategies of the description, predictive relevant outcome and the predictive description of relevant data. These are all presented in the form framework providing an interpretation of the discussion. The data obtained by machine learning provide the accuracy and the relevancy of the demand of human audience [16]. The study proposed by Alaa et al. [17] stated that people who are at a higher risk of cardiovascular attack need more preventative measures for the protection of their hearts. The clinical guideline has presented the model for risk prediction to identify the optimal performance of the patient groups. The data is collected from the machining process to improve the complex learning outcome and the complex interactions. Automatic machine learning (Autoprogress) is one of the techniques used to improve the traditional approaches and the risk prediction of CVD. This increases the accuracy of the data which is obtained from the large population.

However, no research has yet employed ML models to predict hypertension at the population level and verified the models using big datasets in South Asian nations, despite the progress in ML models for individual risk prediction for various illnesses. The present study aimed to determine the model's predictive abilities for hypertension and sought to employ ML techniques to discover characteristics related to hypertension diagnosis.

Therefore the remaining paper is based on an academic article structure which is as follows: in the first part, all the studies and models related to the study have been outlined. Then in methodology, the SMOTE model theory with its implication has been described; how the study has been processed. All the findings and the parameters have been mentioned in detail. Finally, for future research, the direction has been provided with the study limitation, conclusion and proposed ideas based on the study findings.

II. Related Work

Artificial intelligence has a great impact on the field of medicine, diagnosis of the disease or the treatment of diseases. Scientific development has introduced several strategies with the help of advancements in medical technologies and the proposed models. In this study, the use of machine learning and its algorithms with the induced models have displayed the data of the study. This method of study has embedded its pipelines in the data mining procedure. The rules of decision-making and the learning pattern of the algorithms can extract the data. The study uses the predictive models of the SMOTE model comprising of logistic regression, K-nearest neighbour, Naïve Bayes method, REP tree, Random forest, Artificial neural network, and Repeated incremental pruning to produce error reduction (RIPPER). Previous studies have been used to state the classical strategies of statistics. In the study of Dagliti et al. [2018]. The predictive constructive model which has been used is the model validation and the predictive model construction. The missing data have been handled by the Random Forest (RF) to correct the imbalances in the logistic regression for suitable strategies. In this study, the parameters of the feature selection of retinopathy, neuropathy, and neuropathy in different scenarios have been identified in diabetic hospitals. The considered variables are hypertension and hemoglobin, gender, age, and smoking habit. The compilation of this analysis has led to different models to translate easy clinical practices [18]. The major risk of cardiovascular disease is hypertension. The classification of hypertension in traditional Chinese medicine has too much effective methodology according to the syndromes of the patients. The data mining has the multi-learning model of labelling like BrSmote SVM which was developed to deal with the diagnosis and the class unbalancing of the data set. The experiment represents that it has diversified the evaluation of the average precision, one error, and the coverage with the loss ranking [19].

The classification of the multi-labelling is one of the problem classifications which is used for the data mining of the patient and the learning of the strategies of machine learning. This practice has developed the fast accumulation of clinic data [20]. The technique motivated the task of the medical diagnosis and the text categorization. These are divided into two main categories which are the transformation of the problem method and the second is the adaptation of the algorithm method. The problem transformation has more than one regression and hence they depended on the learning algorithm's methods where the algorithm method learning depends directly on the research data for the evaluation [21].

III. Material and Methods

A. Study Design

This case study was done at Petra University from 2019 to 2020. Using medical records of people who went to the hospital, the data was collected to build a model. The study began after the ethics committee gave its approval, and it was done according to the rules of the Declaration of Helsinki. There were a total of 31500 complete cases in the study. Of these, 12658 had high blood pressure and 18842 did not. All of the people who signed up gave their medical record number and baseline and anthropometric information. During the study, patient information was kept secret at all times. The ML-based model which has been used is the SMOTE model consisting algorithmic method which is used to assemble all the predictive models by machine learning pipelines. These consist of feature processing, comprising data imputation, calibration and the classification of the algorithmic predictive models [17].

The basic and familiar risk factors are their age (in years), gender, blood pressure (in mmHg), blood glucose (in mmol/lit), urea (in mmol/lit), and creatinine (in umol/l) which were all used as parameters. We removed data where 30% of the values were missing. After the initial data screening, all of the records went through the labelling process so that the dataset could be cleaned up. This gave us the labelled standard records. This will be used to build a machine-learning model that will use the patient's blood biochemical tests to diagnose hypertension. The model's features are chosen based on how easy they are to use, to collect data, and they must be statistically significant for univariate logistic regression ($p < 0.05$). All the predictive parameters were assessed by using the area under the receiver operating curve of characteristic (AUC-ROC). The overall SMOTE model improved all the risk predictions by 10-fold cross-validation classification performance evaluation of different classifiers on the hypertension dataset on full features without oversampling and 10-fold cross-validation classification performance evaluation using ten classifiers on the hypertension dataset on full features using oversampling. In this experiment, the dataset was run on ten machine learning classifiers using 10-fold cross-validation. 90% of the data was utilized to train the classifiers in the 10-fold cross-validation, whereas only 10% was used to test them.

B. *Synthetic Minority Oversampling Approach (SMOTE)*

The proposed disease prediction model, which was called the "Synthetic Minority Oversampling Approach (SMOTE)," was used to classify hypertension by applying it to a dataset. This technique is the most specific and powerful method used for sampling which follows the algorithm technique to calculate the distance of the space features in minority examples. This helps to develop the synthetic data within the premises of the minority example and required help from the neighbouring data. This technique is also known as borderline SMOTE [22]. The idea is generated from this method to produce the synthetic sample from near boundaries. These algorithms are more effective towards the binary classes having more than two features. This method is predicted by the over-sampling method produced by Chawla et al. [23]. This method has been used to increase the number of the sampling by interpolating the clustered samples which are in minority. The selection of accurate parameters has been mentioned to function correctly for the interpretation of the SMOTE algorithms. This model has three parameters which are: (k) for the neighbour which is very close, (perc. over) used to determine the extra cases of the minority classes, and accurate selection of the parameters. In this model, the method of optimization is one of the best findings for solving the evolutionary biological process [24]. The flow diagram of the over-sampling method (Fig. 1) and the improved hybrid model (Fig. 2) is presented.

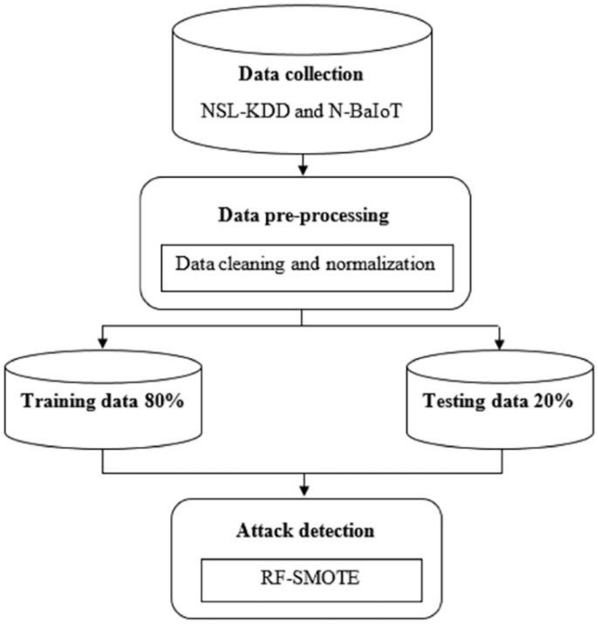


Fig. 1. Workflow radiofrequency SMOTE model. [25].

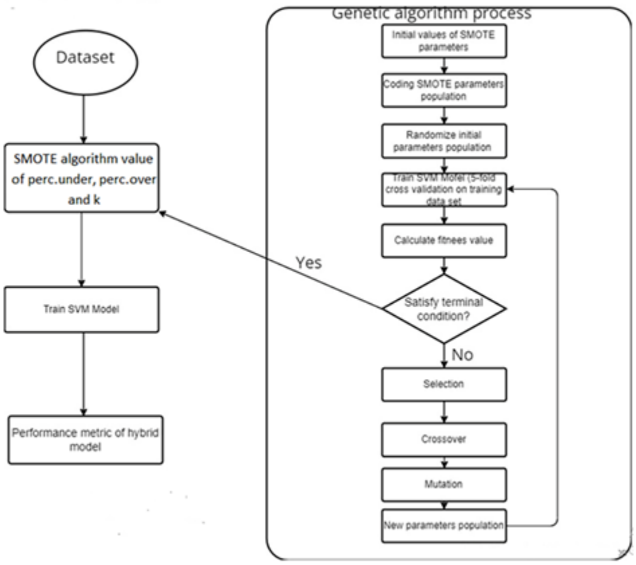

Dataset
SMOTE algorithm value of perc.under, perc.over and k
Train SVM Model
Performance metric of hybrid model
Genetic algorithm process
Initial values of SMOTE parameters
Coding SMOTE parameters population
Randomize initial parameters population
Train SVM Model (5-fold cross validation on training data set
Calculate fitnees value
Satisfy terminal condition?
Yes
No
Selection
Crossover
Mutation
New parameters population

Fig. 2. Improved SMOTE model. [24].

The model's features are chosen based on how easy they are to use to collect data, and they must be statistically significant for univariate logistic regression (p <0.05). The inputs were the number of nearest neighbours, the number of SMOTEs, and the number of minority class samples (MCS). The outputs were synthetic MCS to solve the problem of imbalance in classification [26].

IV. RESULTS

The suggested model uses the Synthetic Minority Oversampling Approach (SMOTE), which is based on the oversampling technique and generates synthetic samples for the minority class. This method assists in addressing the overfitting problem brought on by random oversampling. By overlaying several good examples, it focuses on the dataset to generate new cases [27, 28]. As a result, while evaluating the suggested model, the AUC value is taken into account. Table I displays the recommendations for evaluating any classifier using AUC [29].

In this experiment, the dataset was run on ten machine learning classifiers using 10-fold cross-validation. 90% of the data was utilized to train the classifiers in the 10-fold cross-validation, whereas only 10% was used to test them. The results of ten classifiers' 10-fold cross-validation are shown in Tables II and III.

TABLE I. AUC VALUES DESCRIPTION [3]

AUC Range	Description
AUC = 0.50	Bad classification (no discrimination)
0.50 < AUC < 0.70	Poor classification (poor discrimination)
0.70 ≤ AUC < 0.80	Acceptable classification (acceptable discrimination)
0.80 ≤ AUC < 0.90	Excellent classification (excellent discrimination)
AUC ≥ 0.90	Outstanding classification (outstanding discrimination)

TABLE II. 10-FOLD CROSS-VALIDATION CLASSIFICATION PERFORMANCE EVALUATION OF DIFFERENT CLASSIFIERS ON THE HYPERTENSION DATASET ON FULL FEATURES WITHOUT OVERSAMPLING

Predictive model	**Classifiers' performance evaluation metrics**			
	Accuracy (%)	***Precision (%)***	***Recall (%)***	***AUC (%)***
Logistic regression	76.5	74.4	76.5	79
K-nearest neighbour k=3	78	76.7	78	81.5
PART (rule-induction algorithms)	77.5	78	77.5	71.2
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)	77.5	77.3	77.5	65
NaiveBayes	79	77.8	79	81.7
Decision Tree (J48)	75.5	74.1	75.5	74
REPtree	76	75.1	76	72.2
Support Vector Machine (SVM)	76.5	74.1	76.5	62.4
Random forest	73	70.8	73	79
Artificial Neural Networks (ANN)	74	74.4	74	75.1

Therefore, Table II presents ten classifiers for the hypertension dataset with complete features and no oversampling. Accuracy, precision, recall, and AUC were some of the assessment performance metrics utilized to rate the suggested hypertension prediction model. It became clear that most classifiers can discover trustworthy results using these metrics. The greatest AUC value, equivalent to 81.7, was obtained experimentally for hypertension prediction (high blood pressure detection) using the Naïve Bayes classifier without oversampling.

The minority class of the family history of high blood pressure dataset is oversampled using the SMOTE method. Investigated is the class imbalance caused by the oversampling of the family history of high blood pressure dataset.

The hypertension dataset's entire characteristics were used in the tests, which made use of all classifiers, as shown in Table III. The findings show that all classifiers that used accuracy, precision, recall, and AUC values generated positive outcomes. Based on oversampling AUC values of 89.6 for the k-nearest neighbor, outstanding results were obtained. So, without oversampling, the suggested SMOTE-k-nearest neighbor prediction model outperformed the classifiers. As a result, the SMOTE-k-nearest neighbor prediction model performs exceptionally well, as evidenced by its excellent performance in Table III (83.9% classification accuracy, 85.1% specificity, 83.9% sensitivity, and 89.6% AUC) when compared to other classifiers using 10-fold cross-validation on the hypertension dataset with full features and no oversampling.

The AUC curve, which is a visual depiction of a classification model's true-positive and false-positive rates, is shown in Fig. 3. AUC around one indicates a superior classification with a good class separability metric. AUC which is around 0 indicates a poor model with no distinguishing class. AUC of 0.5 means that the model is unable to distinguish between classes. As shown in Table III, the majority of classifiers provided good classifications, with the k-nearest neighbor getting the greatest AUC rate, which is equivalent to 89.6 and denoting excellent classification, as shown.

TABLE III. 10-FOLD CROSS-VALIDATION CLASSIFICATION PERFORMANCE EVALUATION USING TEN CLASSIFIERS ON HYPERTENSION DATASET ON FULL FEATURES USING OVERSAMPLING

Predictive model	**Classifiers' performance evaluation metrics**			
	Accuracy (%)	***Precision (%)***	***Recall (%)***	***AUC (%)***
Logistic regression	77.9	78.0	77.9	84.1
K-nearest neighbor k=4	83.9	85.1	83.9	89.6
PART (rule-induction algorithms)	80.72	80.9	80.7	79.8
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)	80.7	81.8	80.7	80.5
NaiveBayes	78.7	78.5	78.7	85.6
Decision Tree (J48)	82.7	83	82.7	79.6
REPtree	80.3	80.2	80.3	80.2
Support Vector Machine (SVM)	77.51	77.6	77.5	76.6
Random Forest	81.5	81.9	81.5	86.5
Artificial Neural Networks (ANN)	78.8	79.5	78.7	83.4

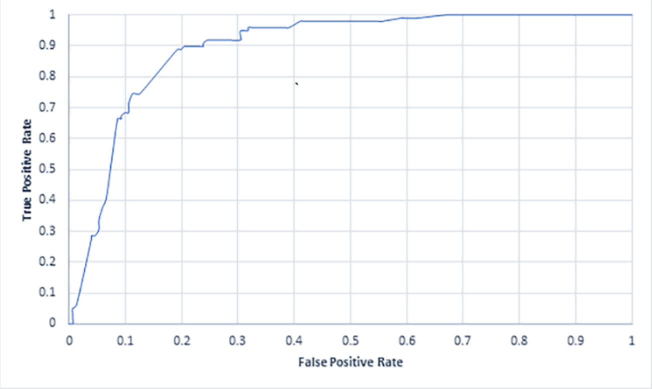


Fig. 3. AUC curves for the family history of high blood pressure dataset.

V. DISCUSSION AND CONCLUSION

The data which has been extracted from Petra University Health Center is considered to be very helpful for ML. It has been availed to produce a decision tree to identify the data related to hypertension. There are many implications to modifying the solutions which can perform preventive measures. These solutions have been identified by using projecting models. According to the previous studies the SMOTE model is proven to be a successful model with the combination of others like BrSmote model for the analysis of the classification of the accuracy into five multi-label classifiers and the ten multi-label classifiers. The average and loss ranking is more sensitive and accurate in SMOTE model as compared to BrSmote model. The diseases like hypertension and other syndromes can be classified to improve the diagnosis of the patients [19]. These models can be very predictive for those people who can develop hypertension at very high risk. [16]These predictive models provide better risk communication. They can guide people who are concerned about their health decision and diagnosis of disease. This awareness in the community develops positive impact.[25] Furthermore, the lowest classification of the number is based on the class interest and the parameter which are required for the diagnosis. In the real world, the classification varies with the nature of the risk management, detection of fraud and the diagnosis of the disease in the medical history [30]. This positive mediation helps to decide the significant level of understanding. In the future, this research can improve the reliability of the predictive models. It can minimize the count of unhealthy effects on a large population by using accurate algorithms. There are many assessing tools for predicting different algorithms. These predictors are boosting gradient machines, supporting vector machines, classifiers of naive Bayes, and neural artificial networks.

ACKNOWLEDGEMENT

The author is thankful to all the associated personnel who contributed to this study by any means.

REFERENCES

[1] B. Zhou, P. Perel, G.A. Mensah and M. Ezzati, “Global epidemiology, health burden and effective interventions for elevated blood pressure and hypertension,” Nat. Rev. Cardiol. vol. 18, pp. 785-802, November 2021. https://doi.org/10.1038/s41569-021-00559-8.

[2] W.H.O, “Global status report on noncommunicable diseases 2010,” World Health Organization, 2011.

[3] S.M.S. Islam, T.D. Purnat, N.T.A. Phuong, U. Mwingira, K. Schacht and G. Fröschl, “Non-Communicable Diseases (NCDs) in developing countries: a symposium report,” Glob. Health. vol. 10, pp. 1-8, July 2014. https://doi.org/10.1186/s12992-014-0081-9.

[4] K.T. Mills, A.Stefanescu and J. He, “The global epidemiology of hypertension,” Nat. Rev. Nephrol. vol. 16, pp. 223-237, September 2020. https://doi.org/10.1038/s41581-019-0244-2.

[5] S. Basu and C. Millett, “Social epidemiology of hypertension in middle-income countries: determinants of prevalence, diagnosis, treatment, and control in the WHO SAGE study,” Hypertension, Standford CA, vol. 62, pp. 18-26, November 2013. https://doi.org/10.1161/hypertensionaha.113.01374.

[6] Krishnan and R. Garg, “Hypertension in the South-East Asia region: an overview,” In Reg. Health Forum. India, vol. 17, pp. 7-14, September 2013.

[7] R.B. D’Agostino Sr, R.S. Vasan, M.J. Pencina, P.A. Wolf, M. Cobain, J.M. Massaro and W.B. Kannel, “General cardiovascular risk profile for use in primary care: the Framingham Heart Study,” Circulation, vol. 117, pp. 743-53, April 2008. https://doi.org/10.1161/CIRCULATIONAHA.107.699579.

[8] D. C. Goff, Jr, D. M. Lloyd-Jones, G. Bennett, S. Coady, R. B. D'Agostino, R. Gibbons, P. Greenland, D. T. Lackland, D. Levy, C. J. O'Donnell, J. G. Robinson, J. S. Schwartz, S. T. Shero, S. C. Jr, Smith, P. Sorlie, N. J. Stone, P. W. Wilson, H. S. Jordan, L. Nevo and J. Wnek, “2013 ACC/AHA guideline on the assessment of cardiovascular risk: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines,” Circulation, vol. 129, pp. S49-73, November 2014. https://doi.org/10.1161/01.cir.0000437741.48606.98.

[9] J.B. Echouffo-Tcheugui, G.D. Batty, M. Kivimäki and A.P. Kengne, “Risk models to predict hypertension: a systematic review,” PloS one. California, vol. 8, pp. e67370, July 2013. https://doi.org/10.1371/journal.pone.0067370.

[10] Q. Bi, K.E. Goodman, J. Kaminsky and J. Lessler, “What is machine learning? A primer for the epidemiologist,” Am. J. Epidemiol. Baltimore, vol. 188, pp. 2222-2239, December 2019. https://doi.org/10.1093/aje/kwz189.

[11] J.J. Beunza, E. Puertas, E. García-Ovejero, G. Villalba, E. Condes, G. Koleva, C. Hurtado and M.F. Landecho, “Comparison of machine learning algorithms for clinical event prediction (risk of coronary heart disease),” J. Biomed. Inform. Spain, vol. 97, pp. 103257. July 2019. https://doi.org/10.1016/j.jbi.2019.103257.

[12] E.A. Martinez-Ríos, M.R. Bustamante-Bello and L.A. Arce-Sáenz, “A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data,” Biomed. Signal Process Control. Mexico, vol. 68, pp. 102813, July 2021https://doi.org/10.1016/j.bspc.2021.102813.

[13] M. Nour and K. Polat, “Automatic classification of hypertension types based on personal features by machine learning algorithms,” Math. Probl. Eng. Saudia Arabia, pp. 1-13, January 2020. https://doi.org/10.1155/2020/2742781.

[14] J.J. Boutilier, T.C. Chan, M. Ranjan and S. Deo, “Risk stratification for early detection of diabetes and hypertension in resource-limited settings: machine learning analysis,” JMIR. Hyderabad, India, vol. 23, pp. e20123, January 2021. https://doi.org/10.2196/20123

[15] X. Diao, Y. Huo, Z. Yan, H. Wang, J. Yuan, Y. Wang, J. Cai and W. Zhao, “An application of machine learning to the etiological diagnosis of secondary hypertension: retrospective study using electronic medical records,” JMIR. Med. Inform. Beijing, China, vol. 9, pp. e19739, January 2021. https://doi.org/10.2196/19739.

[16] Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences. 2019 Oct 29;116(44):22071-80. https://doi.org/10.1073/pnas.1900654116.

[17] AM. Alaa, T. Bolton, E. Di Angelantonio, JH. Rudd, M. Van der Schaar, “Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK Biobank participants,” PloS one. vol. 14, pp. e0213653.May 2019. https://doi.org/10.1371/journal.pone.0213653.

[18] Dagliati, S. Marini, L. Sacchi, G. Cogni, M. Teliti, V. Tibollo, P. De Cata, L. Chiovato, R. Bellazzi, “Machine learning methods to predict diabetes complications,” J. Diab. Sci. Tech. vol. 12. pp. 295-302., March 2019 https://doi.org/10.1177/1932296817706375.

[19] GZ. Li, Z. He, FF. Shao, AH. Ou, XZ. Lin, “Patient classification of hypertension in Traditional Chinese Medicine using multi-label learning techniques,” BMC Med. Gen. vol. 8. pp.1-6, December 2015. https://doi.org/10.1186/1755-8794-8-S3-S4.

[20] GP. Liu, GZ. Li, YL. Wang, YQ. Wang, “Modelling of inquiry diagnosis for coronary heart disease in traditional Chinese medicine by using multi-label learning,” BMC Med. vol. 10. pp. 1-2. December 2010 https://doi.org/10.1186/1472-6882-10-37.

[21] YM. Huang, CM. Hung, HC. Jiau, “Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem,” Nonlinear Analy. Wor. App. vol. 7. pp.720-47, September 2006. https://doi.org/10.1016/j.nonrwa.2005.04.006.

[22] H. Han, Wang WY, Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1 2005 (pp. 878-887). Springer Berlin Heidelberg.).

[23] NV. Chawla, KW. Bowyer, LO. Hall, WP. Kegelmeyer. “SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research,” vol.1.pp. 321-57, June 2002.

[24] P. Akın, “A new hybrid approach based on genetic algorithm and support vector machine methods for hyperparameter optimization in synthetic minority over-sampling technique (SMOTE),” AIMS Mathematics. vol. 8. pp.9400-9415, 2023.

[25] MG. Karthik, MM. Krishnan, “Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks.” J.A.I.H.C. pp.1-1, March 2021.

[26] Azad, B. Bhushan, R. Sharma, A. Shankar, K.K. Singh and A. Khamparia, “Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus,”

Multimed. Syst. Switzerland, vol. 28, pp. 1289-1307, June 2022. https://doi.org/10.1007/s00530-021-00817-2.

[27] J. Zhai, M. Wang and S. Zhang, “Binary imbalanced big data classification based on fuzzy data reduction and classifier fusion,” Soft. Comput. Switzerland, vol. 26, pp. 2781-2792, March 2022. https://doi.org/10.1007/s00500-021-06654-9.

[28] Data, Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu and Y. Zhou, “A novel ensemble method for classifying imbalanced data,” Pattern. Recognit. China, vol. 48, pp. 1623-1637, November 2015. https://doi.org/10.1016/j.patcog.2014.11.014.

[29] Jr, D.W. Hosmer, S. Lemeshow and R.X. Sturdivant, “Applied logistic regression.” John Wiley & Sons, vol. 398, 2013.

[30] Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural networks. 2008 Mar 1;21(2-3):427-36.

Medical Name Entity Recognition Based on Lexical Enhancement and Global Pointer

Pu Zhang, Wentao Liang
School of Computer Science and Technology, Chongqing University of Posts and Telecommunications,
Chongqing, P. R. China

***Abstract*—Named entity recognition (NER) in biological sources, also called medical named entity recognition (MNER), attempts to identify and categorize medical terminology in electronic records. Deep neural networks have recently demonstrated substantial effectiveness in MNER. However, Chinese MNER has issues that cannot use lexical information and involve nested entities. To address these problems, we propose a model which can handle both nested and non-nested entities. The model uses a simple lexical enhancement method for merging lexical information into each character's vector representation, and then uses the Global Pointer approach for entity recognition. Furthermore, we retrain a pre-trained model with a Chinese medical corpus to incorporate medical knowledge, resulting in F1 score of 68.13% on the nested dataset CMeEE, 95.56% on the non-nested dataset CCKS2017, 85.89% on CCKS2019, and 92.08% on CCKS2020. These data demonstrate the efficacy of our proposed model.**

Keywords—MNER; nested NER; Global Pointer; lexical enhancement

I. Introduction

Electronic medical records are a digital repository of a patient's comprehensive medical and health information that can be used for medical and healthcare services. These are vital repositories of medical knowledge and therapeutic experience, containing detailed content about patients' diagnoses and treatment histories. These enable deeper analysis to extract and consolidate valuable medical knowledge, transform implicit knowledge into explicit knowledge, and build a medical knowledge system with a clear hierarchy, well-defined concepts, rich connotations, and significant practical implications. However, due to the high percentage of unstructured data in electronic medical records, direct exploitation is difficult. MNER is a vital step in the extraction of medical information and is essential for biomedical text mining.

MNER has received a lot of attention in recent years, which has resulted in a lot of assessment contests. The China Conference on Knowledge Graph and Semantic Computing (CCKS) held NER evaluation challenges for medical texts from 2017 to 2021. Meanwhile, AliCloud and the Chinese Information Processing Society of China (CIPSC) have released the Chinese Medical Information Extraction (CMeIE) subtask of the Chinese Biomedical Language Understanding Evaluation (CBLUE).

The previous studies in Chinese MNER have mainly used methods based on English MNER to improve the model performance [1]. Although these methods produce good results, they rely on word-level annotation models. However, unlike English, Chinese is not naturally tokenized and does not segment words by spaces in sentences, leading to additional ambiguities when using word segmentation as an additional step in Chinese MNER, which could result in inaccurate entity boundary detection and class prediction due to improper word segmentation errors [2]. As a solution to this problem, character-based Chinese MNER techniques that are better at avoiding segmentation errors have been proposed. Considering that character-based approaches cannot fully exploit lexical information, Zhang et al. [3] introduced the Lattice-LSTM model, which combines lexical information into a character-based recognition model. However, the complex structure of Lattice-LSTM made it difficult to combine with other models.

On the other hand, in the Chinese MNER, the challenge of identifying nested entities has remained unsolved. Several earlier models are based on sequence models, however not all entities in electronic medical records are self-contained, and there may be nested structures between them. As shown in Fig. 1, in the sentence, where the entity "肺 (lung)" is nested within the other entity "肺内病变 (lung lesions)". Because of the complexity of the nested entity structure and the irregularity in its granularity and number of nested levels, it is difficult to rapidly and accurately gather nested entity information for semantic comprehension, a critical component of improving Chinese MNER.

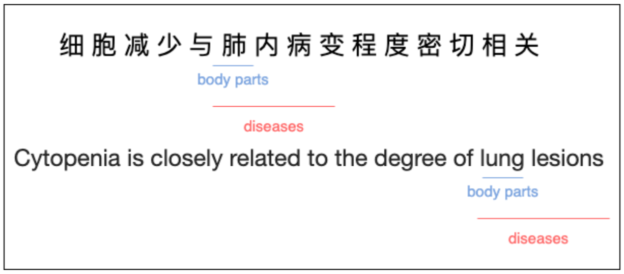

细胞减少与肺内病变程度密切相关
body parts
diseases
Cytopenia is closely related to the degree of lung lesions
body parts
diseases

Fig. 1. Example of nested entities.

To address the aforementioned challenges, our research provides a Chinese MNER model based on lexical enhancement and Global Pointer. The model first adds lexical information to each character in the electronic medical record by using the lexical enhancement method, and then uses the Global Pointer method to score the beginning and end of each character to identify the medical entity. The following are the primary contributions of our work:

1) We incorporate the word lexicons into the character representations by introducing a lexical enhancement approach. The approach matches each character with a dictionary built from a corpus to get word sets. Then the word set is compressed and combined with character representation. The experimental results demonstrate that the model with lexical enhancement can improve performance.

2) The Global Pointer is used as the entity recognition module. Extensive experiments are carried out on the nested dataset CMeEE and the non-nested datasets including CCKS2017, CCKS2018, and CCKS2020. The experimental results show that the Global Pointer model can use a unified method to deal with both nested and non-nested entity recognition problems and has better performance than some recent models.

3) The pre-trained model is retrained with Chinese medical corpus for experimentation. It has better results than the basic pre-trained model.

The final experimental results show that the proposed model can be universally applied to nested and non-nested Chinese MNER tasks, and both yield the best results on the four datasets.

The remaining sections are organized as follows: Section II presents the related work on MNER. Section III introduces our proposed model. Section IV presents experiments and results analysis on our model. Section V concludes this paper.

II. Related Work

The purpose of NER is to discover entities in a text and classify them into specified categories such as person, organization, place, and so on. NER is a critical component of information extraction that allows structured information to be extracted from unstructured text input. Effective NER models are required for a variety of downstream tasks, including entity linking, relation extraction, and event extraction. MNER focuses on recognizing and categorizing clinical terminologies such as symptoms, medicines, and therapies in medical data. The MNER task is often treated as a sequential labelling issue, with the objective of assigning a category to each word or character in the text.

The neural network models are now the recommended strategy for English NER. The Bi-LSTM-CRF [4] model is the most typical, incorporating Bi-directional Long Short-Term Memory (Bi-LSTM) for feature extraction and Conditional Random Fields (CRF) for decoding. Unlike English text, Chinese text does not possess clear boundary information. Hence, Chinese NER methods can be broadly categorized into two groups: word-based methods and character-based methods. In the word-based method, word segmentation is performed first, followed by entity recognition [5]. However, this approach may result in the propagation of errors from inaccurate word segmentation, thus leading to incorrect NER. The character-based method, on the other hand, operates on each word individually and does not have the issue of error propagation, but cannot utilize lexical information. Consequently, researchers in word-based models are striving to improve the use of word information [6]. The majority of current studies demonstrate that character-based methods often surpass word-based methods in Chinese NER, due to the issue of error propagation in word segmentation. So, we build a character-based model for Chinese MNER that incorporates lexical information to address the limitations of traditional character-based methods.

MNER in the Chinese language is primarily researched using deep learning-based approaches. Bi-LSTM-CRF model has been proposed for predicting the sequence labels in a posterior conditional random field. Gridach et al. [7] were the pioneers in using a Bi-LSTM-CRF model for NER in the biomedical domain. Dang et al. [8] fine-tuned the word vectors using linguistic information based on the Bi-LSTM-CRF model. Liu et al. [9] combined a multi-channel convolutional neural network with the Bi-LSTM-CRF model and used lexical and morphological features of words as information for entity recognition.

In addition to the Bi-LSTM-CRF model, there have been several studies that apply other deep learning models to the medical field for MNER. For instance, Qiu et al. [10] utilized a residual dilated convolution model for efficient and quick NER in the medical field. Zhang et al. [11] proposed a hybrid model of Dilated Convolutional Neural Network (DCNN) and Bi-LSTM for hierarchical encoding, taking advantage of DCNN to gain global information with fast computational speed. Du et al. [12] proposed a multi-task learning approach with multi-strategies based on MRC. NER in the medical field can be approached as a sequence labeling task or a span boundary detection task.

However, the methods mentioned above are based on sequence labeling and cannot be directly used to solve the identification problem of nested named entities, because the same lexical entry in a nested named entity may have two or more different labels at the same time.

Solving the identification problem of nested entities will also improve the accuracy of the model in extracting entities. Previously, a combination of rule-based and machine-learning-based approaches was often used to deal with nested named entities. First, the inner non-nested named entities are identified using the Hidden Markov Model (HMM). Then, the other named entities are identified using rule-based post-processing. Alex et al. [13] proposed several CRF-based models for nested named NER on the GENIA dataset. These methods apply CRFs to entity types in a specific order, so that each CRF can use the output of the previous CRFs. This cascading approach can achieve the best results for nested named NER. In 2009, Finkel and Manning [14] implemented the task of nested NER from a parsing perspective. They constructed a selection tree to map all named entities to a node in the tree. Rule based and machine learning based methods have high accuracy and can make rules according to a specific domain to extract nested named entities. However, there are some problems such as difficulty in recognizing the same type of nested named entities, high time complexity, and difficulty in scaling to large datasets with long sentences.

Recently, the research of nested NER has become a hot topic in the field of information extraction. Span-based methods have become increasingly popular due to their high performance. For example, Xu et al. [15] used a local detection approach, where each possible entity span is classified independently. Sohrab and Miwa [16] introduced a simple deep neural model that enumerates all possible spans and classifies them using LSTM. Wang et al. [17] proposed a transition-based method that builds nested entities incrementally by performing a series of actions designed for this purpose. Tan et al. [18] extended the span-based approach by including a boundary detection task that predicts entity boundaries in addition to classifying spans. Quoc et al. [19] proposed a two-stage entity recognition method to address the limitations of span-based models. Our method is also a span-based approach, but unlike previous studies, our model predicts each character as the beginning or end of each span without enumerating each span. As a result, it is highly efficient.

III. Proposed Model

To address the problem that traditional lexical enhancement methods are complicated and cannot be easily transferred to different deep learning neural network architectures, a simpler approach is used to quickly merge lexical information into each character's vector representation. Meanwhile, the entity recognition module uses a span-based entity recognition approach as the entity recognition module. The Fig. 2 depicts the model's general architecture:

A. *Lexical Enhancement*

The input sentence $s = \{c_1, c_2, \cdots, c_n\} \in V_c$ is processed as a sequence of characters by the Chinese character-based NER model, where $\mathcal{V}_c$ denotes the character vocabulary. Each character c_i is represented by a word vector:

$$x_i^c = e^c(c_i) \tag{1}$$

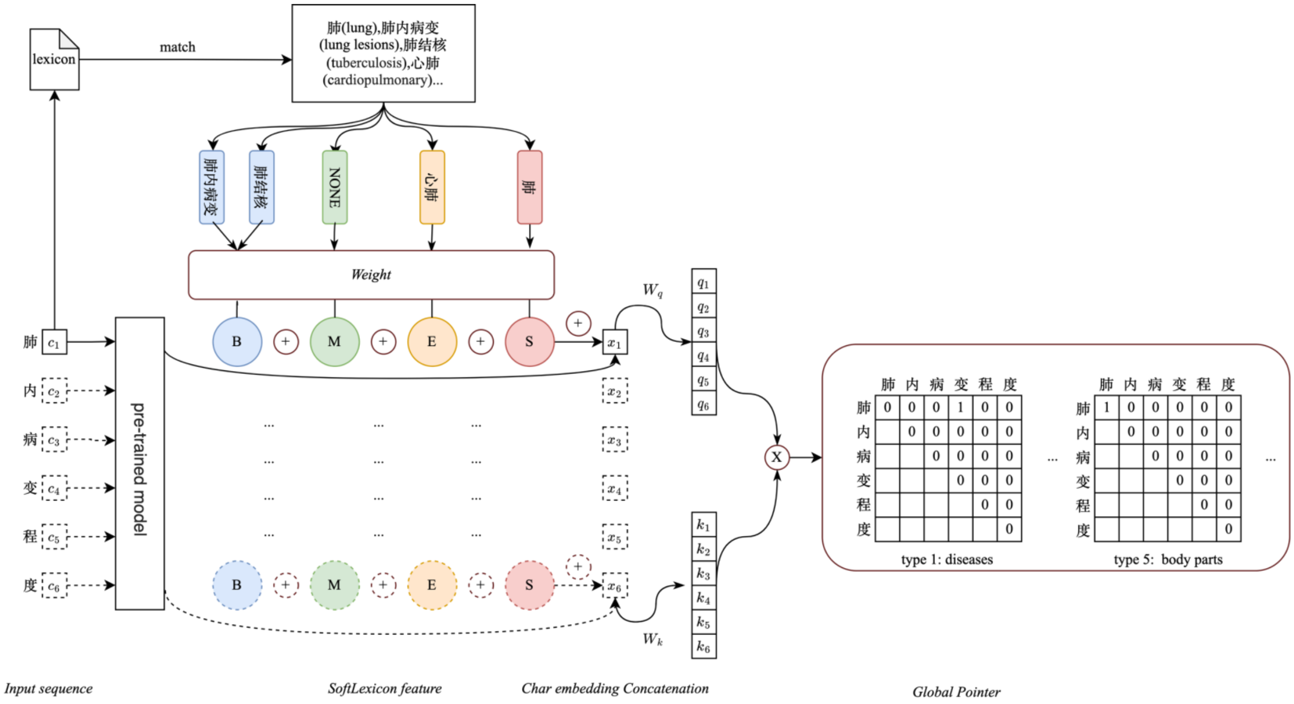
where e^c denotes the character embedding lookup table.

Character-based NER techniques have the disadvantage of not fully utilizing the information contained in words. To address this, we use SoftLexicon, a lightweight dictionary matching approach for entity recognition in Chinese electronic medical records. SoftLexicon incorporates lexical information into character representations, addressing the issue of character-based NER models' inability to use word information while simplifying the lexical enhancement process [20]. The lexical enhancement features are constructed in three steps:

First, in order to preserve the lexical information of characters, all matching words for each character C_i are divided into four sets of "BMES," which are constructed as follows:

$$\begin{aligned}
&\mathrm{B(c_i)} = \{\mathrm{w_{i,k}}, \forall \mathrm{w_{i,k}} \in \mathrm{L}, \mathrm{i} < \mathrm{k} \leq \mathrm{n}\} \\
&\mathrm{M(c_i)} = \{\mathrm{w_{j,k}}, \forall \mathrm{w_{j,k}} \in \mathrm{L}, 1 \leq \mathrm{j} < \mathrm{i} < \mathrm{k} \leq \mathrm{n}\} \\
&\mathrm{E(c_i)} = \{\mathrm{w_{j,i}}, \forall \mathrm{w_{j,i}} \in \mathrm{L}, 1 \leq \mathrm{j} < \mathrm{i}\} \\
&\mathrm{S(c_i)} = \{\mathrm{c_i}, \exists \mathrm{c_i} \in \mathrm{L}\}
\end{aligned} \tag{2}$$

Where $\boldsymbol{w}_{i,j}$ denotes the subsequence $\{c_i, c_{i+1}, \cdots, c_j\}$ and L is the lexicon we built. We constructed it from the already labelled texts of the train and dev sets and counted the number of their occurrences for the second step. The set $\mathrm{B(c_i)}$ represents the word set with the character $\mathrm{c_i}$ at the beginning and the length is greater than 1, the set $\mathrm{M(c_i)}$ represents the word set with the character $\mathrm{c_i}$ in the middle and the length is greater than 1, and the set $\mathrm{E(c_i)}$ represents the character $\mathrm{c_i}$ at the end and the length is greater than 1, the set $\mathrm{S(c_i)}$ represents a single character $\mathrm{c_i}$. If the word set corresponding to the current character is empty, it is set to the special word "NONE". Fig. 3 is an example of the SoftLexicon vocabulary extension. In this way, word embedding can be introduced and there is no loss of information because the tag matching result can be accurately restored by the four word sets.



	肺	内	病	变	程	度
肺	0	0	0	1	0	0
内		0	0	0	0	0
病			0	0	0	0
变				0	0	0
程					0	0
度						0

type 1: diseases

...

	肺	内	病	变	程	度
肺	1	0	0	0	0	0
内		0	0	0	0	0
病			0	0	0	0
变				0	0	0
程					0	0
度						0

type 5: body parts

...

Fig. 2. Architecture of the model.

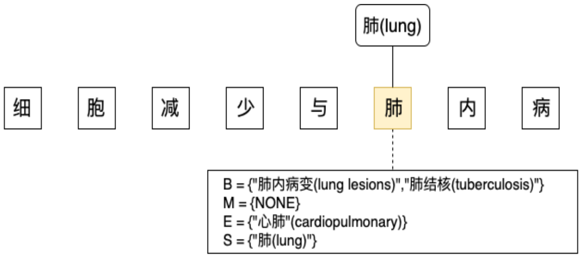

肺(lung)
细
胞
减
少
与
肺
内
病
变
B = {"肺内病变(lung lesions)","肺结核(tuberculosis)"}
M = {NONE}
E = {"心肺"(cardiopulmonary)}
S = {"肺(lung)"}

Fig. 3. SoftLexicon method.

In the second step, transform each word in the word set using a pre-trained word vector; then perform weight normalization on all words in the four word sets, using a static weighting method based on statistics, that is, the frequency of each word in the static data. This frequency can reflect the importance of the word to a certain extent, and the frequency is obtained statistically when building a dictionary. The weighted method is as follows:

$$\boldsymbol{v}^s(S) = \frac{4}{Z}\sum_{w \in S} z(w)\boldsymbol{e}^w(w) \tag{3}$$

where S is the "BMES" word sets, $z(w)$ is the frequency of word w in the dictionary in the static statistics, Z is the sum of all words in the word set, and e^w is the lexical embedding lookup table.

Finally, the representations of the four word sets are combined into a one-dimensional feature, which is then stitched onto the representation of that character vector to obtain the final input vector.

$$\boldsymbol{e}^s(\mathrm{B,M,E,S}) = [\boldsymbol{v}^s(\mathrm{B}); \boldsymbol{v}^s(\mathrm{M}); \boldsymbol{v}^s(\mathrm{E}); \boldsymbol{v}^s(\mathrm{S})] \tag{4}$$

$$x^c \leftarrow [x^c; \boldsymbol{e}^s(\mathrm{B,M,E,S})] \tag{5}$$

where v^s is the result of the calculation in the previous step.

B. Global Pointer

In our work, we use the Global Pointer [21] as an entity recognition module, which is a span-based entity recognition method. Span-based methods identify named entities by classifying sub-sequences of sentences. This method outperforms sequence annotation-based methods in preventing error propagation and is able to easily detect nested named entities as they belong to different sub-sequences.

The Global Pointer concept is similar to a simplified multi-head attention mechanism, with as many heads as there are entities. For a sequence of length n, NER is performed by the αth category. The sequence has a total of $\frac{n(n+1)}{2}$ candidate entities, containing all possible entities. The entity recognition task is to select the actual entity from these $\frac{n(n+1)}{2}$ candidate entities. The Global Pointer scores each character to determine whether it is the beginning or the end of an entity. Based on this idea, the lexically enhanced vector token x^{ci} can be transformed into $\boldsymbol{q}_{i,\alpha}$ and $\boldsymbol{k}_{i,\alpha}$:

$$\boldsymbol{q}_{i,\alpha} = \boldsymbol{W}_{q,\alpha}\boldsymbol{x}^{ci} + \boldsymbol{b}_{q,\alpha} \tag{6}$$

$$\boldsymbol{k}_{i,\alpha} = \boldsymbol{W}_{k,\alpha}\boldsymbol{x}^{ci} + \boldsymbol{b}_{k,\alpha} \tag{7}$$

where $\boldsymbol{W}$ is the weight matrix and $\boldsymbol{b}$ is the bias. The $\boldsymbol{q}_{i,\alpha}$ and $\boldsymbol{k}_{i,\alpha}$ are the vector representations of the token which used to identify the entity of type α. Specifically, for span s $[i{:}j]$ of type α, the start and end positions are represented by $\boldsymbol{q}_{i,\alpha}$ and $\boldsymbol{k}_{i,\alpha}$. The score of an entity of type α can then be calculated as follows:

$$s_\alpha(i,j) = \boldsymbol{q}_{i,\alpha}^{\top}\boldsymbol{k}_{j,\alpha} \tag{8}$$

In the inference step, the segments for which the condition $s_\alpha(i,j) > 0$ is satisfied are considered the output entities of type α.

The use of Global Pointers alone is insufficient to accurately identify different types of entities as it does not take into account the length and span of the entities. For example, in the sentence "细胞减少与肺内病变程度密切相关 (cytopenia is closely related to the degree of lung lesions)", "细胞减少与肺 (cytopenia and lung)"may be mistaken as a single entity, as "细(fine)" is the start of the entity "细胞减少 (cytopenia)" while "肺(lung)" is the end of the entity "肺 (lung)". The model treats this combination of start and end positions as one entity. To overcome this issue, it is critical to incorporate relative position information into the Global Pointer method, which is more sensitive to the length and span of entities. Global Pointer uses rotational position encoding to encode relative position information. This encoding is based on a transformation matrix $\boldsymbol{R}_i$ with the property $R_i^{\top}R_j = R_{j-i}$, which can be applied to q and k respectively. The dot product between q and k is then transformed by the relative position information, resulting in a more accurate representation of the entity.

$$\begin{aligned} s_\alpha(i,j) &= \left(R_i\boldsymbol{q}_{i,\alpha}\right)^{\top}\left(R_j\boldsymbol{k}_{j,\alpha}\right) \\ &= \boldsymbol{q}_{i,\alpha}^{\top}R_i^{\top}R_j\boldsymbol{k}_{j,\alpha} \\ &= \boldsymbol{q}_{i,\alpha}^{\top}R_{j-i}\boldsymbol{k}_{j,\alpha} \end{aligned} \tag{9}$$

C. Loss Function

Entity recognition is a multi-label classification problem, so we use a multi-label classification loss function for category imbalance to perform multi-label classification for each label category in the $\frac{n(n+1)}{2}$category after obtaining scores for all segments of each label category. Because the number of entities present in each input text phrase is frequently fewer than $\frac{n(n+1)}{2}$, direct bi-classification causes serious category imbalance problems. Therefore, the method treats the problem as a two-by-two comparison of target and non-target category scores and uses cross-entropy to compute self-balancing weights to avoid the label category imbalance problem, which can be formulated as follows:

$$\mathcal{L} = \log\left(1 + \sum_{(i,j)\in P_\alpha} e^{-s_\alpha(i,j)}\right) + \log\left(1 + \sum_{(i,j)\in Q_\alpha} e^{s_\alpha(i,j)}\right) \tag{10}$$

where i, j represent the start and end indexes of a span, P_α refers to a set of spans that have entity type α, while Q_α refers to a set of spans that either do not have an entity type or have a different entity type from α. The function $s_\alpha(i,j)$ calculates the score for a span $s[i:j]$ to be an entity of type α.

IV. Experiments

A. Experimental Settings and Evaluation Metrics

The PyTorch deep learning framework is used to create the experimental model. For our experiments, we used pre-trained models including BERT-Base-Chinese [22] and RoBERTa-large [23]. To incorporate more medical information, we also used Bertcner [24], which was retrained on a medical corpus based on BERT-Base-Chinese.

Table I lists the hyper-parameter settings used in the model:

TABLE I. The Hyper-Parameter Settings

Parameters	**Value**
learning rate	2e-5
batch size	16
epochs	10
max sequence length	256
hidden size	768

We calculate the classification metrics including True Positive (TP), False Positive (FP), and False Negative (FN), and evaluate the performance of the model using the metrics of precision (P), recall (R) and Micro-F1($F1$):

$$\Pi\rho\varepsilon\chi\iota\sigma\iota o\nu = \frac{\sum_{i=1}^{n} \mathrm{TP}_i}{\sum_{i=1}^{n} \mathrm{TP}_i + \sum_{i=1}^{n} \mathrm{FP}_i} \tag{11}$$

$$\mathrm{Recall} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \tag{12}$$

$$\mathrm{F1} = 2 \cdot \frac{Precision \ \cdot \ Recall}{Precision + \ Recall} \tag{13}$$

B. Introduction of Nested Dataset

The Tianchi Chinese Biomedical Language Understanding Evaluation Benchmark, jointly provided by Peking University, Zhengzhou University, Pengcheng Laboratory, and Harbin Institute of Technology (Shenzhen), published the CMeEE dataset, with a total of 938 files annotated with 47,194 sentences divided into nine major medical entity categories: diseases (dis), clinical (sym), and so on. We conducted experiments and analyzed them on this dataset to validate the effectiveness of our model. The data sources of the dataset include clinical trials, electronic medical records, medical books and search logs from real-world search engines. As biomedical data may contain privacy information such as patients' names, ages and genders, all collected data are anonymous and reviewed by ethics committees to protect privacy.

The CMeEE dataset differs from traditional NER in that there is a nested relationship between entities, which is a common phenomenon in medical texts and makes the model processing more complex. The nested entity instances accounted for about 11% of the training set. The dataset contains 15,000 sentences in the training set, 5,000 sentences in the validation set, and 3,000 sentences in the test set. To protect privacy, all data were anonymized and reviewed by an ethics committee.

C. Baselines of Nested Dataset

The comparison models used are the benchmark models officially released by CBLUE, and these methods are implemented based on the pre-trained model, including:

1) BERT-Base-Chinese: The basic model used had 12 layers, 768 hidden layers, 12 headers, and 110 million parameters.

2) RoBERTa-large: RoBERTa removed the next sentence prediction target and dynamically changed the masking pattern applied to the training data.

3) Bertcner: A medical pre-training model was obtained by crawling a 1.05G clinical text consisting of different medical domain corpora obtained on the web to train the BERT model again.

4) ZEN [25]: A BERT-based Chinese text encoder was enhanced by n-gram representation, considering different character combinations in training.

5) Mac-BERT-Base/large [26]: The Mac-BERT was an upgraded BERT that included a new Masked Language Model (MLM) as a correction pre-training assignment, which reduced the differences between pre-training and fine-tuning.

6) PCL-MedBERT: A pre-trained medical language model was proposed by the Intelligent Medicine Research Group in the PengCheng Lab, which excelled in medical question matching and NER.

In addition to the officially provided baselines, the following experiments are used for comparison:

7) TPLinker: Wang et al. [27] developed a single-stage joint extraction approach for addressing entity relationship extraction designs that could find overlapping relationships between one or two entities while being unaffected by exposure bias. The approach utilized to identify entity pieces was chosen as a comparison method in this paper.

8) Muti-head: Li et al. [28] developed a training strategy based on fragment annotation to solve the lack of entity data annotation. The essential idea was to employ negative sampling, which prevented NER models from being trained on unlabeled items.

9) Biaffine: Yu et al. [29] introduced a new NER approach that treats NER as a problem of dependent syntactic analysis, using graph neural networks to model the global information of the input sequence.

D. Results and Analysis of Nested Dataset

As shown in Table II, the F1 scores for entity recognition using only pre-trained models including BERT-Base-Chinese, RoBERTa-large, Bertcner, ZEN, Mac-BERT-Base/large, and PCL-MedBERT range from 60.7% to 62.8%. Compared to the human score of 67%, there is a gap of at least 4.2%. One important reason for this gap is the presence of many nested entities in this dataset, which cannot be recognized by these basic models based on sequence modeling. Therefore, our model uses a span-based model called Global Pointer as the entity recognition module. This model can calculate whether each character in the input text can be the beginning or end of an entity. It can recognize small entities nested within larger ones without a limit on the nesting level. We conducted three sets of experiments to validate this approach. Specifically, GP+BERT-Base-Chinese improved performance by 4.66% as compared to utilizing only BERT-Base-Chinese. When compared to the model utilizing only Bertcner, using GP+Bertcner improved the score by 4.58%, while using Global Pointer with RoBERTa-large, the GP+RoBERTa-large model improved the score by 5.99%. Additionally, our model GP+ BERT-Base-Chinese outperformed two other span-based models, Muti-head and Biaffine, by 1.93% and 3.91%, respectively. The above results demonstrate the effectiveness of Global Pointer. This also demonstrates that if nested entities can be recognized, it can significantly improve the performance of model and make it more suitable for MNER.

TABLE II. RESULTS OF CMEEE DATASET

Methods	F1(%)
BERT-Base-Chinese	62.1
RoBERTa-large	62.1
Bertcner	62.8
ZEN	61.0
Mac-BERT-base	60.7
Mac-BERT-large	62.4
PCL-MedBERT	60.6
TPLinker	64.32
Muti-head	64.83
Biaffine	62.85
GP[a]+ BERT-Base-Chinese	66.76
GP[a]+BERT-Base-Chinese+ SoftLexicon	67.43
GP[a]+Bertcner	67.38
GP[a]+Bertcner+SoftLexicon	67.71
GP[a]+RoBERTa-large	68.09
GP[a]+RoBERTa-large+SoftLexicon	**68.13**
Human	67.0

[a] "GP" means Global Pointer

To verify that a character-based model with added lexical information would perform better, we conducted experiments using SoftLexicon as a vocabulary enhancement method. Compared to experiments using only Global Pointer, we found a generally improved performance, with an increase of approximately 0.1% to 0.67%. This demonstrates that, under the same experimental conditions, adding lexical information into word embeddings has a certain gain effect on entity recognition.

We also conducted comparative experiments using Bertcner, which was trained on medical corpora based on the BERT-base model. When using pre-trained models directly for experiments, Bertcner outperformed BERT-Base-Chinese by 0.7%. With the addition of Global Pointer, GP+Bertcner outperformed GP+BERT-Base-Chinese by 0.62%. When SoftLexicon was added, GP+Bertcner+SoftLexicon still resulted in a 0.28% improvement over GP+BERT-Base-Chinese+ SoftLexicon. These results demonstrate that using BERT models trained on medical corpora can further improve the recognition of medical entities.

E. Introduction of Non-nested Datasets

Three non-nested datasets provided by the CCKS competition were used in the experiments. the CCKS-2017 dataset has 300 electronic clinical records and 29,866 labeled entities, categorized into five entity types: treatments, signs and symptoms, diseases and diagnoses, examinations and tests, and body parts. the CCKS-2019 dataset has 23,401 labeled entities, annotated into six entity types: diseases and diagnoses, examinations, tests, procedures, medications, and anatomical sites. The CCKS-2020 dataset has 32,120 labeled entities, with the same entity type classification as CCKS2019.

F. Baselines of Non-nested Datasets

The comparative models are presented below:

1) RoBERTa-Bi-LSTM-CRF[30]: The model contains three layers, a character embedding layer, a Bi-LSTM layer, and a CRF layer, relying on character-based word representations learned from a supervised corpus. The Bi-LSTM-CRF model can improve medical named entity recognition by capturing contextual information using bi-directional LSTM, and by modeling dependencies between tags using CRF.

2) RoBERTa-Bi-GRU-CRF[31]: The neural network model integrated Bi-GRU and CRF for sequence labeling tasks. Bi-GRU is a gated recurrent unit, an improved RNN that can solve the gradient vanishing and long-term dependency problems. By feeding the Bi-GRU output into CRF, the Bi-GRU-CRF network may use both bi-directional context information and label constraints for sequence tagging at the same time.

3) Bertcner[24]: A medical pre-training model was obtained by crawling a 1.05G clinical text consisting of different medical domain corpora obtained on the web to train the BERT model again.

4) Ra-RC [32]: The model used RoBERTa as an encoder to capture contextual information and adds part-level features on top of it to enhance the understanding of Chinese language.

5) AR-CCNER [33]: The model used part-level characteristics to augment character semantic information and a self-attention technique to record character interdependence.

6) ACNN [34]: This method effectively learned global context information using an attention mechanism and multi-layer CNNs and captured both short-term and long-term contextual information.

7) BE-Bi-CRF-JN [35]: This method combined the original text in NER tasks with its medical encyclopedia knowledge by establishing connections and interactions to enhance the ability of entity recognition.

8) RGT-CRF [36]: This model used two sets of features, word-based and word-based features, to make full use of the characteristics of Chinese language. The model also used a rule generator to automatically construct rules to improve the generalization ability of the model.

G. Results and Analysis of Non-Nested Datasets

As shown in Tables III, IV and V, we conduct experiments on the CCKS2017, CCKS2019, and CCKS2020 datasets to evaluate our model's performance on non-nested datasets. The experimental results indicate that using Global Pointer as the entity recognition module has a significant improvement compared to using only pre-trained models. Compared with BERT-Base-Chinese, RoBERTa-large, and Bertcner, using the Global Pointer method shows an improvement of 2.37% to 5.35% on the three datasets.

TABLE III. RESULTS OF CCKS2017 DATASET

Methods	P(%)	R(%)	F1(%)
BERT-Base-Chinese	90.92	90.09	90.50
RoBERTa-large	91.99	93.01	92.49
Bertcner	91.10	90.44	90.76
RoBERTa-Bi-GRU-CRF	92.56	93.09	92.82
RoBERTa-Bi-LSTM-CRF	92.41	94.11	93.25
Ra-RC	94.14	92.39	93.26
ACNN	90.19	90.78	90.49
AR-CCNER	92.27	93.73	93.00
RGT-CRF	95.47	95.76	95.61
GP[b]+BERT-Base-Chinese	93.68	95.77	94.71
GP[b]+RoBERTa-large	94.34	95.40	94.86
GP[b]+Bertcner	94.07	95.62	94.83
GP[b]+BERT-Base-Chinese+SoftLexicon	94.51	95.64	95.06
GP[b]+RoBERTa-large+SoftLexicon	95.05	95.05	**95.56**
GP[b]+Bertcner+SoftLexicon	94.93	95.71	95.32

[b] “GP” means Global Pointer

TABLE IV. RESULTS OF CCKS2019 DATASET

Methods	P(%)	R(%)	F1(%)
BERT-Base-Chinese	79.94	74.63	77.19
RoBERTa-large	79.85	79.85	79.85
Bertcner	81.29	79.43	80.35
RoBERTa-Bi-GRU-CRF	72.95	79.78	76.21
RoBERTa-Bi-LSTM-CRF	79.7	80.75	80.22
Ra-RC	83.31	82.44	82.87
ACNN	83.07	87.29	85.13
BE-Bi-CRF-JN	83.16	86.67	84.88
RGT-CRF	85.36	84.99	85.17
GP[c]+BERT-Base-Chinese	81.87	81.44	81.49
GP[c]+RoBERTa-large	83.04	87.61	85.15
GP[c]+Bertcner	84.04	87.64	85.70
GP[c]+BERT-Base-Chinese +SoftLexicon	83.35	87.72	85.40
GP[c]+RoBERTa-large+SoftLexicon	84.72	87.33	**85.89**
GP[c]+Bertcner+SoftLexicon	82.60	88.83	85.52

[c] “GP” means Global Pointer

TABLE V. RESULTS OF CCKS2020 DATASET

Methods	P(%)	R(%)	F1(%)
BERT-Base-Chinese	87.78	88.24	88.01
RoBERTa-large	86.68	88.2	87.43
Bertcner	88.57	88.69	88.63
RoBERTa-Bi-GRU-CRF	76.74	88.09	82.02
RoBERTa-Bi-LSTM-CRF	87.05	87.67	87.35
BE-Bi-CRF-JN	82.52	85.05	83.76
RGT-CRF	90.85	91.57	91.2
GP[d]+BERT-base-Chinese	90.20	92.31	91.15
GP[d]+ RoBERTa-large	90.76	91.71	91.09
GP[d]+Bertcner	90.13	92.61	91.27
GP[d]+BERT-Base-Chinese+SoftLexicon	90.30	92.89	91.44
GP[d]+RoBERTa-large+SoftLexicon	90.76	93.61	**92.08**
GP[d]+Bertcner+SoftLexicon	89.74	93.72	91.60

[d] “GP” means Global Pointer

Compared with the mainstream model using BiLSTM-CRF, our GP+RoBERTa-large model has gained 1.61%, 4.93%, and 3.74% F1 performance improvement over RoBerta-Bi-LSTM-CRF on the CCKS2017, CCKS2019, and CCKS2020 datasets.

In addition, we have compared our model with some other methods, such as the Ra-RC model that uses bi-directional long short-term memory networks to learn radical features of Chinese characters, the AR-CCNER model that uses convolutional neural networks to extract aggressive features while using self-attention mechanism to capture dependencies between characters, and the ACNN model that uses a multi-layer CNN structure to capture short-term and long-term contextual relationships for experiments, our model still outperformed these models in terms of F1 performance.

To verify the effect after adding vocabulary information, SoftLexicon was added separately to introduce vocabulary information for experimentation. Compared with the GP model without using vocabulary information, there is an improvement of about 1%. We also compared our model with other models combining lattice method with Chinese character information such as RGT-CRF. The performance is comparable on CCKS2017 dataset and improved by 0.72% on CCKS2019 and by 0.88% on CCKS2020 in terms of F1.

Finally, we experimented incorporating medical information into MNER model by using Bertcner trained on medical corpus data to obtain semantic features. Compared with basic pre-training model, GP+Bertcner+SoftLexicon has an improvement of 0.1%-0.5% improvement on the F1 over GP+BERT-Base-Chinese+SoftLexicon. When comparing it with BE-Bi-CRF-JN, there are improvements of about 1% and 8% on CCKS2018 and CCKS2020 datasets respectively.

H. Limitation Analysis

Compared with other methods, our model achieved the best results. However, there are still many entities that have not been recognized. In order to promote future work on MNER, this section analyzes the reasons for identification errors. The causes of errors are broadly classified into the following two cases:

1) Ambiguity: Some entities may have different meanings or belong to different categories in different contexts, resulting in difficulties and inaccuracies in entity recognition. To solve this problem, contextual information should be used to determine the true meaning or category of the entity.

2) Inadequate medical knowledge: Because the medical field contains a large number of terms and complex concepts, identifying and classifying these entities can be difficult for non-professionals. For example, in order to correctly classify entities such as diseases, drugs, symptoms, and treatment plans, it is necessary to have a thorough understanding of these concepts. Furthermore, medical knowledge evolves quickly, with new research findings and treatment methods constantly emerging. In this case, relying solely on pre-trained models may not meet real-time demands; collaboration with domain experts can ensure timely model updates, allowing for the resolution of new challenges.

V. Conclusion

In this paper, we propose a model which uses the Global Pointer with a lexical enhancement method and demonstrate its effectiveness for Chinese MNER on nested and non-nested datasets. By using lexical enhancement to incorporate word lexicons into the character representations, our model can perform Chinese NER at the character level and avoid the word segmentation errors. By using Global Pointer, our model can recognize both nested and non-nested entities by enabling a global view that takes the beginning and end locations into account. Experiment results, conducted on the CMeEE, CCKS2017, CCKS2019, and CCKS2020 datasets, show that the proposed model has excellent performance on these four data sets. Our results establish a new benchmark for Chinese MNER and open avenues for further research and exploration. In future work, we will investigate the way to leverage unlabeled data and extend our work to more datasets.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62136002 and 61876027.

References

[1] Z. Jin, X. He, X. Wu and X. Zhao, “A hybrid transformer approach for Chinese ner with features augmentation,” Expert Systems with Applications, vol. 209, p. 118385, 2022.

[2] R. Ding, P. Xie, X. Zhang, W. Lu and L. Li, “A neural multi-digraph model for Chinese NER with gazetteers,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 2019.

[3] Y. Zhang and J. Yang, “Chinese NER using Lattice LSTM,” in the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, 2018.

[4] R. Chalapathy, E. Z. Borzeshi and M. Piccardi, “Bidirectional LSTM-CRF for clinical concept extraction,” in Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP), Osaka, Japan, 2016.

[5] H. N. Zhang, D. Y. Wu, Y. Liu and X. Cheng, “Chinese named entity recognition based on deep neural network,” Journal of Chinese Information Processing, vol. 31, no. 4, pp. 28-35, 2017.

[6] Y. Li, L. Zou, W. Liu, W. Liu, and X. Wang, "Research on chinese clinical named entity recognition: lattice lstm with contextualized character representations," JMIR Med Inform, 2020, 8(9): e19848.

[7] M. Gridach, "Character-level neural network for biomedical named entity recognition," Journal of biomedical informatics, 2017, 70: 85-91.

[8] T. H. Dang, H. Q. Le, T. M. Nguyen, and S. T. Vu, "D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information," Bioinformatics, 2018, 34(20): 3539-3546.

[9] J. Liu, S. Chen, Z. He, and H. Chen, "Learning BLSTM-CRF with multi-channel attribute embedding for medical information extraction," Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part I 7. Springer International Publishing, 2018: 196-208.

[10] J. Qiu, Q. Wang, Y. Zhou, T. Ruan, and J. Gao, "Fast and accurate recognition of Chinese clinical named entities with residual dilated convolutions," 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2018: 935-942.

[11] R. Zhang, P. Zhao, W. Guo, R. Wang, and W. Lu, "Medical named entity recognition based on dilated convolutional neural network," Cognitive Robotics, 2022, 2: 13-20.

[12] X. Du, Y. Jia, and H. Zan, "MRC-based medical NER with multi-task learning and multi-strategies," China National Conference on Chinese Computational Linguistics. Springer, Cham, 2022: 149-162.

[13] B. Alex, B. Haddow, and C. Grover, "Recognising nested named entities in biomedical text," Biological, translational, and clinical language processing. 2007: 65-72.

[14] J. R. Finkel, and C. D. Manning, "Nested named entity recognition," Proceedings of the 2009 conference on empirical methods in natural language processing. 2009: 141-150.

[15] M. Xu, H. Jiang, and S. Watcharawittayakul, "A local detection approach for named entity recognition and mention detection," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1237-1247.

[16] M. G. Sohrab, and M. Miwa, "Deep exhaustive model for nested named entity recognition," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 2843-2849.

[17] B. Wang, W. Lu, Y. Wang, and H. Jin, "A neural transition-based model for nested mention recognition," Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 1011-1017.

[18] C. Tan, W. Qiu, M. Chen, R. Wang, and F. Huang, "Boundary enhanced neural span classification for nested named entity recognition," Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(05): 9016-9023.

[19] Q. C. Quoc, and V. N. Van, "NER-VLSP 2021: two stage model for nested named entity recognition," VNU Journal of Science: Computer Science and Communication Engineering, 2022, 38(1).

[20] R. Ma , M. Peng , Q. Zhang , and X. Huang , " Simplify the usage of lexicon in chinese NER," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 5951-5960.

[21] J. Su, A. Murtadha, S. Pan, J. Hou, J. Sun, and W. Huang, "Global Pointer: novel efficient span-based approach for named entity recognition," arXiv preprint arXiv:2208.03054, 2022.

[22] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[23] Y. Liu, M. Ott, J. Du, M. Joshi, D. Chen, and V, Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[24] X Li, H Zhang, and X H Zhou, "Chinese clinical named entity recognition with variant neural structures based on BERT methods," Journal of biomedical informatics, 2020, 107: 103422.

[25] S. Diao, J. Bai, Y. Song, T. Zhang, and Y Wang, "ZEN: pre-training Chinese text encoder enhanced by n-gram representations," Findings of the Association for Computational Linguistics: EMNLP 2020. 2020: 4729-4740.

[26] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for Chinese natural language processing," Findings of the Association for Computational Linguistics: EMNLP 2020. 2020: 657-668.

[27] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "TPLinker: single-stage joint extraction of entities and relations through token pair linking," Proceedings of the 28th International Conference on Computational Linguistics. 2020: 1572-1582.

[28] Y. Li, L. Liu, and S. Shi, "Empirical analysis of unlabeled entity problem in named entity recognition," arXiv preprint arXiv:2012.05426, 2020.

[29] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 6470-6476.

[30] K. Xu, Z. Zhou, T Hao, and W. Liu, "A bidirectional LSTM and conditional random fields approach to medical named entity recognition," Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017. Springer International Publishing, 2018: 355-365.

[31] Q. Qin, S. Zhao, C. Liu, "A BERT-BiGRU-CRF model for entity recognition of Chinese electronic medical records," Complexity, 2021, 2021: 1-11.

[32] Y. Wu, J. Huang, C. Xu, H. Zheng, L Zhang, and J. Wan, "Research on named entity recognition of electronic medical records based on roberta and radical-level feature," Wireless Communications and Mobile Computing, 2021, 2021: 1-10.

[33] M. Yin, C. Mou, K. Xiong, J. Ren, "Chinese clinical named entity recognition with radical-level feature and self-attention mechanism," Journal of biomedical informatics, 2019, 98: 103289.

[34] J. Kong, L. Zhang, M. Jiang, and T. Liu, "Incorporating multi-level CNN and attention mechanism for Chinese clinical named entity recognition, " Journal of Biomedical Informatics, 2021, 116: 103737.

[35] Q. Wang, and E. Haihong, "Bi-directional Joint Embedding of Encyclopedic Knowledge and Original Text for Chinese Medical Named Entity Recognition," 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT). IEEE, 2021: 304-309.

[36] J. Li, R. Liu, C. Chen, S. Zhou, X. Shang, and Y. Wang, "An RG-FLAT-CRF Model for Named Entity Recognition of Chinese Electronic Clinical Records, "Electronics, 2022, 11 (8): 1282.

Distributed Focused Web Crawling for Context Aware Recommender System using Machine Learning and Text Mining Algorithms

Venugopal Boppana[1]*, Dr. P. Sandhya[2]
Research Scholar[1], Associate Professor[2]
School of Computer Science and Engineering, Vellore Institute of Technology,
Chennai Campus, Chennai, India[1, 2]

***Abstract*—In today's world, Recommender System (RS) is the most effective means used to manage the huge amount of multimedia content available on the internet. RS learns the user preferences and relationships among the users and items. It helps the users to discover new interesting items and make use of different media types such as text, audio, video and images. RS can act as an information filtering model which can overcome the issues related to over-fitting and excess information. In this work, a new distributed framework named DAE-SR (Deep AutoEncoder based Softmax Regression) is introduced for context-aware recommender systems which focus on user-item based interaction and offers personalized recommendations. The proposed model is implemented in PYTHON platform. The dataset used for experimentation is Foursquare dataset. The performance of the proposed context-aware RS is beneficial to both the users and service providers. Its helps in decision making process and can offer relevant recommendations to users. The performance is evaluated in terms of various metrics such as accuracy, recall, precision and so on. From the implementation outcomes, the proposed strategy achieved good accuracy (98.33%), precision (98%), run time (1.43 ms) and recall (98.1%). Thus, it is proved that the proposed DAE-SR classifier performs better compared to other models and offer dependable and relevant recommendations to users.**

Keywords*—Recommender Systems (RSs); context-aware; softmax regression; deep autoencoder; multimedia information

I. Introduction

With the exponential development of products and services over the internet, individuals face issues related to data deluge as each individual have to make choices in a reasonable manner to save their valuable sources such as money and time [1-3]. Individuals possess admirable communication skills for expressing the ideas among each other in a proper form. The use of contextual information made the designers to increase the abundance of communication between human and computer interaction which enables the growth of intellectual applications. Now, RSs can be useful in various modern applications which expose the user towards a massive collection of items. These structures usually offer a list of recommended items to the user in which each individual can predict or prefer an item [4-5]. These procedures are helpful to the users to choose the most suitable items and may ease the process of finding the desired items in the recommendation list.

The web crawler has turned out to be one of the key innovations for clients to naturally get data from assigned locales. The customary web crawler innovation has uncovered a few issues, for example, low material exactness because of basic sifting conditions regarding crawling themes, low productivity because of substance duplication and long website page refresh time [6-7]. Web crawler innovation is acquainted as a method to productively obtain and process data, and it includes different research territories, for example, distributed computing, data recovery, machine learning and web application [8-9]. Web crawlers are for the most part used to make a duplicate of all the visited pages for last handling. Additionally, crawlers can be utilized to assemble particular sorts of data from website pages, for example, gathering email addresses [10-11]. Quick fetching of relevant pages for the given topic with very less searches is possible with focussed crawler.

RS aims to seek the satisfaction of the individual/user via personalization and chooses the most appropriate services, items and provides it to the users considering the details such as user profile, opinions, preferences, purchase history, relationships with clients and interactions with services and products offered. The dual main entities of the RSs are the users, items which learn to create predictions by learning the parameters which minimize the loss among the actual and predicted preferences. Moreover, the RSs can be classified into 3 types such as CF (Collaborative Filtering), CB (Content-based) and HR (Hybrid Recommendation). Among these three categories, CB model gain advantages over the traditional RS system by exploiting the features of context-information as user activity, time, and location.

The integration of RS with context information influences the value of these models by enhancing the possible relevant recommendations in accordance to the varying user requirements. The CEP (Complex Event Processing) is integrated with RSs to provide personalized recommendations and allows the business users to put on event-driven rules for controlling and filtering the input parameters of recommendation engines [12-14]. The concept of context-based RSs minimizes the gap among the information models and users, therefore these information models can actively realize the user's context and offer better experience to users [15-16]. Recommender systems (RSs) are a subclass of data filtering frameworks that look to anticipate the rating that a client would provide for an item.

Recommender frameworks have been connected to variety of uses, e.g., films, music, news, books, look into articles, seek inquiries, social labels, and money related administrations. As a rule there are three different ways to plan recommender frameworks, i.e., cooperative separating, content-based filtering [17], and the hybrid filtering [18]. Recommender models have attracted consideration in both industry and academia. Such frameworks help to oversee data over-burden via assembling the data self-sufficiently and fitting it to singular interests [19]. A recommender framework would have the capacity to catch the clients' preferences. Social Recommender Systems (SRSs) or Recommenders for the Social Web demonstrate the client's preferences by utilizing the data he or she and their companions have distributed in online informal communities [20-21].

A. *Motivation*

Recommender Systems gained benefits by solving the problem of handling abundant data in the internet. Individuals plan to make effective and smart decisions on the huge number of available choices in various fields such as: a place the person wishes to visit, a movie the person likes to watch, a book he/she wants to read etc. The main motive behind this generation of recommender system is to offer the intelligence that benefits both the consumers and organizations. However, there is an emerging trend of using contextual information to afford meaningful personalized recommendations. The context enhances the basic interaction of user-item to realize recommendations of higher quality. Moreover, recommendation using content-based framework effectively use the information content describing the item or user such as item description or user profile to execute accurate recommendations. Also, content-based (CB) recommenders are more suitable with sparse datasets and may not suffer from the cold-start issue. As the textual data is used to represent the item description and user profile, this content based context-aware model make use of the common evaluation metrics to measure the similarity between the contents. The recommendations based on contextual information emphasize that the choice made by individuals is liable on certain context, rather than being invariant of it. Sometimes, the same users might select diverse services/products under a dissimilar context. Hence, this condition motivated the authors to develop a precise prediction of preferences by consumer depending on the appropriate contextual data which is being combined with the recommendation method.

B. *Contributions*

In this work, context-aware approach is developed by utilizing the contextual user information, in order to produce more accurate recommendation according to user’s preferences. This recommendation model makes use of the information from the dataset in order to create user recommendations on basis of given number of choices. The main novelty of the proposed DAE-SR recommendation model is to develop a context-aware/content-based RS that estimates the characteristics of restaurant and relevant context factors against the user preference model to create top recommendations for restaurant names based on the similarity score calculated. These scores can be used later to sort out the top restaurants for each user according to their expectation. Our proposed (DAE-SR) RS adopts a user preference model by using features of liked/visited restaurants by the user and takes relevant context factors, e.g., restaurant ratings, location, current weather, and time of the day. Most of the existing research works use CF based recommendation because of its higher performance accuracy. Since, CF approach is more effective and reliable one of the major drawbacks reported was the higher scalability. Hence, a new recommendation framework is presented in this work based on context-aware which improves the quality of recommendations. The major contributions of this paper are summarized as follows:

- To introduce a distributed framework for context-aware RS based on text mining and machine learning algorithms in order to afford quick and correct suggestions to user on good restaurants name.
- To present the content-based context-aware RS to create an item list that reveals the user’s preference and minimizes the error in the classification process.
- To develop a hybrid DAE-SR based recommendation on restaurant names which is trained with the meta-heuristic algorithm called JO (Jaya Optimization) algorithm which has good ability of exploration and exploitation.
- The three similarity measures such as Dice’s, Jaccard and Cosine similarity coefficients are used to compute the similarity of the content and to obtain high accuracy in receiving top recommendations.

C. *Organization*

The paper is organized as follows. In Section I, a brief introduction about the topic is given. Section II gives the works related to our proposed algorithm and Section III gives the proposed methodology. The result of the proposed algorithm is provided in Section IV and the entire work is concluded in Section V with future suggestion and references.

II. Related Works

The main motivation of related works section is to explain about the importance behind each research. In this this section, the discussion about the features of existing approaches, its applications, limitations, metrics and datasets used is clearly described in Table form.

Afolabi *et al.* [22] have presented a semantic web content mining scheme for RS in shopping by online. Through a web crawler, the web textual dataset was gathered built in Java. The combination of an existing ontology and a developed ontology was utilized for the textual data's semantic preprocessing. Next, the recommendation was created by the Naïve Bayes algorithm. The large dimensions of attributes were handled by the annotations to ensure the preprocessing models optimized semantic understanding.

Amato *et al*. [23] presented a technique SOS: A multimedia RS for Online Social systems. In this viewpoint, the RS was introduced for supporting the browsing of information accumulations and helping users to discover what they truly required. A new RS for big data applications was introduced in this research, which was able to deliver recommendations based on the interactions between users.

A hybrid learning based recommender framework was proposed by Tarus *et al.* [24]. In this paper, hybrid information based RS was introduced on the basis of sequential pattern mining (SPM) and ontology. The domain knowledge about the learning and learner resources was represented and modelled by the ontology, although the SPM algorithm determines the learners' sequential learning patterns.

RS with Linked Open Data (RS-LOD) and matrix factorization process (MF-LOD) was proposed by Natarajan *et al.* [25] for reducing the problems of cold- start and sparsity. The cold start issue was solved by the RS-LOD approach, and the data sparsity problem was eliminated by the MF-LOD approach. The hidden feedback data was elongated the user vector, and the similar semantic items were elongated each item. The collaborative filtering RS's precision was enhanced by the Semantic features from LOD.

Iqbal *et al.* [26] proposed a context-aware kernel-based recommendation scheme that creates the design over the user-item matrix of context rating. The importance of contextual information was considered for this recommendation. The KCR (Kernel Context Recommendation) algorithm was accurate, flexible and scalable enough to include the various contexts and was utilized for creating the practical recommendations.

A Preference Relation based PMF framework for RSs was proposed by Pujahari and Sisodia [27]. User's PR was taken as input, and the recommendation was also generated. The user's and item's neighborhood data were integrated into the design. The users' preferences towards items were obtained by the Probabilistic Matrix Factorization (PMF) system. This approach was introduced for Collaborative Filtering (CF) in RS. Aghdam [28] proposed a hierarchical HMM (hidden Markov model) in which the changes were identified in the preferences of user on time through modelling the user's latent context. The user as a hidden Markov process was modelled by the user-selected items.

Sangaiah, A.K *et al.* [30] developed a new technique with the help of ML (Machine Learning) methods to conserve the confidentiality of PBSs (Position Based Services) roaming users. The emergence of the three-phase procedure was used for the identification of PBS user position by means of combining k-NN (k-Nearest Neighbours) and DT (Decision Trees). The destination of the user was estimated by means of the HMM (Hidden Markov Models) and the position track sequence. The service policy followed was the mobile edge computing which was useful for ensuring the PBS timely delivery.

Sangaiah, A.K *et al.* [31] presented an energy-aware model (EAM) named green adversary for the application of smart industrial environment. The proposed model conserves the information and position confidentiality whereas the existing models make use of the cyber-physical systems by mutually using the software and hardware parts in which the energy consumption was minimized. However, this limitation was tested on the recommended approach related to the CPS (Cyber Physical Security) smart industrial applications and has removed the severe effects and also tested on CPS real-world applications.

Sangaiah, A.K *et al.* [32] presented whale optimization algorithm (WOA) to obtain optimal solution for network issues. WOA was powerful heuristic algorithms that boosted the exploration process and prevented bottleneck. However, the main drawback of this algorithm was a low convergence rate. Likewise, Bat algorithm (BA) was a novel meta-heuristic algorithm presented by Sangaiah, A.K *et al.* [33] to achieve optimal solutions in terms of sustainability and quality. It has the ability to monitor and control the network activities. Though, the performance of BA for huge data was not efficient. It was considered as a major disadvantage in BA.

Gupta Garima and Rahul Katarya [36] presented an Ensemble Particle Swarm Optimization (EnPSO) approach which was considered as an AutoML (AutomatedML) to minimize the model selection complexity. The presented EnPSO optimizes the recommendations intelligently using the identification of best ensemble model. The dataset used for testing was MovieLens in which the recommendations were improved with the AutoML system. The accuracy of the model was verified with the RMSE metric with lower error as 0.918 compared to the baseline IBCF (Item-Based Collaborative Filtering) with Singular Value Decomposition (SVD) having 0.961 higher errors. The main challenge was to find an effective way to reach the performing scheme with the vast search space. The limitation of the suggested approach was that EnPSO cannot provide effective performance gains within the constrained time for some cases.

Katarya Rahul, and Yamini Arora [37] developed a novel scheme called CapsMF (Capsule Networks Matrix Factorization) for product RS. The DNN (Deep Neural Network) text analysis architecture was enhanced with the Bi-RNN (Bi-directional Recurrent Neural Network) for the representation of text descriptions in robust form. The integration of Probabilistic MF with the DNN has the ability to generate recommendations in improved manner. The accuracy of the system was analysed with MAE (0.8878) and RMSE (1.157) shows improved outcomes. The major limitation observed was the training time was quite higher with the Caps Network during the experimentation process.

Katarya Rahul [38] presented a reliable RS model using the improved CF technique. A new self-devised method named P-distance algorithm was used to predict the ratings and acts as a dual-level filtering procedure to identify the nearest neighbours. The experiments were tested on 3 datasets such as Douban, Movielens and Jester. The performance was evaluated with MAE and RMSE metrics to show the effectiveness of the proposed algorithm. Sparsity was the major problem identified with this reliable RS.

Katarya Rahul [39] developed a hybrid RS utilizing KMC (K-means clustering) with ABC (Artificial Bee Colony) optimization procedure. The hybrid KM-ABC approach offers accurate movie predictions by considering the user ratings.

Various metrics such as accuracy, recall, precision, MAE were evaluated for verifying the system performance on the Movielens dataset. The experimental results on the dataset indicate that the proposed KM-ABC based recommender system has gained improved performance regarding reliability, accuracy, and personalization for the recommendation of movies. The drawback associated with this system was that only user ratings were considered and the other user characteristics were not considered.

Yadav *et al.* [40] presented a review on RS using FL (fuzzy logic). With the combination, with fuzzy logic three categories were reviewed such as content-based, memory-based and model-based systems. FL was utilized to assign the fuzzy values to the user items and represents the preference of user in robust form. The Fuzzy recommender was used to compute similarity using the fuzzy values rather than the user-based crisp data. This review offers significant knowledge about the various fuzzy based RS models. The fuzzy logic imitates the decision making process of humans which cover all the intermediate values.

Katarya Rahul [41] developed an improved RS utilizing dual-level MF (matrix factorization). This improved framework incorporates product ontologies for learning the items recommendations. The proposed matrix – factorization based RS initiates each item by generating similarity measures with each other. The similarity was created with the help of genres and a matrix was created using the given user ratings, so that the user gets his corresponding recommendation. Sparsity was the major problem identified with this reliable RS. The dataset used for experimentation was MovieLens dataset and the metrics used for evaluation were MAE and RMSE.

Katarya Rahul and Om Prakash Verma [42] introduced a web-based RS which was mainly based on successive information of user's navigation on web pages. FCM (Fuzzy C Means) approach was integrated to receive top-N clusters. Weights were calculated for each of the page category and the top page recommendations were predicted for the targeted user. The experimentation was performed on the real-world MSNBC dataset. Compared to other clustering approaches the proposed FCM model has gained 33%accuracy. The proposed RS required lower computation speed of 402 seconds whereas the other methods show similar as well as enhanced outcomes.

Katarya Rahul and Om Prakash Verma [43] presented a hybrid model for music recommendations. The proposed HMRS (Hybrid Music RS) represents the combination of both collaborative-content features. The algorithms such as BF (Bellman-Ford) and DFS (Depth First Search) were utilized with the multi-layered context graph. Here the similarity score was computed using PC (Pearson Correlation) coefficient. The overall results were optimized with the PSO algorithmic procedure. The music dataset used was Last.fm and the accuracy of the system was evaluated with recall metric on Top-N recommendations.

Katarya Rahul *et al.* [44] developed a location-based RS framework with the help of improved RW (Random Walk) model. Both the structural as well the attribute properties were utilized to recommend the location in SNs (Social Networks). The computations of accurate recommendations on locations were done with the improved RW model. The experimentation was conducted with three real-world datasets such as Gowalla, Brightkite and Foursquare. The accurateness of the system was evaluated with precision, F1 score and recall metrics.

Katarya Rahul *et al.* [45] presented a HF (Hybrid Filtering) approach in analysing the user behaviour in content-aware RS. The dual filtering strategies used are the pre-filtering and post-filtering models. The HF model eliminates the recommended items which has the lesser chance of relevance. Based on the significance of the contextual attribute both the filters were combined which forms a hybrid model. This HF approach minimizes the sparsity problem also it was quicker than the post-filtering method. The experimentation was performed on 50 movies that were selected equally from Movielens dataset and the precision calculation determines the accurateness of the system. Sanchez et al. [46] presented a RS for food delivery on the basis of number of orders. Nearest-neighbour (NN) algorithm was used for the evaluation of users preferred restaurants and buying patterns. Teixeira et al. [47] introduced a diabetic friendly restaurant recommendation using MAS (Multi-Agent System) for multi-criteria decision making. The users with diabetic illness can be benefitted and can find out suitable nearby restaurants. Table I represents the review on various recommendation systems.

A. *Research Gaps*

Most of the works developed in the context aware recommender system concentrates only on improving the accuracy of recommendation. But, the major purpose is to improve the recommendation accuracy when considering the design objectives such as a user's item's context. Moreover, the main challenge of a RS is to generate a significant recommendations using context based rating of user and item information. Hence, the recommender system should be accurate, flexible and that should predict the contexts accurately. Moreover, some of the recommender systems are providing the improved accuracy but not providing the rating of contexts. Some of the recommendation systems shows decreased computational complexity but fails in the accurate prediction and also results in the difficulty of generating recommendations.

The running time of the previous RS is noted to be higher while providing the recommendations which depend on the previous preferences of users. The preference may sometimes change based on time because of the mood change of users otherwise the change of contexts. Numerous earliest techniques are presented based on the statement of every user having the static pattern. But without these changes, the recommendation does not equal the user's choices. The context aware dependent recommendation systems are dealing with this problem through the contextual data. The contextual data usage is difficult because of attaining all contextual data for processing. Moreover, the different types of contexts used in the RS extend its dimensionality. Hence, to overcome all these issues, the authors of this work presented a Distributed Focused Web crawling with Machine Learning and Text Mining algorithms for the effective context aware recommendation system.

TABLE I. REVIEW ON VARIOUS RECOMMENDATION SYSTEMS

Author and Reference	Technique Used	Dataset Used	Applications	Metrics	Limitation
Afolabi *et al.* [22]	Semantic Web Mining + NB classifier	Wed textual data collected using Web crawler	Product Recommendation	Precision, Recall, F-score	Results are not optimal.
Amato *et al*. [23]	SOS	YFCC100M	RS for big data applications	MAE, RMSE	Information overload
Tarus *et al.* [24]	Hybrid recommendation framework	Real world dataset	e-learning	MAE, Precision, Recall	Machine learning techniques are not used.
Natarajan *et.al.* [25]	RS-LOD	Netflix and MovieLens datasets	Social network, e-commerce	MAE, RMSE, Precision, Recall, F1-score	Machine learning and Deep learning techniques are not used.
Iqbal *et al.* [26]	KCR	LDOS-CoMoDa, DePaulMovie datasets	Movie Recommendations	RMSE, F1-Measure	Do not consider the context under which ratings are provided.
Pujahari and Sisodia [27]	PMF	MovieLens 1M and MovieLens 20M datasets	Movie Recommendations	MAP, NDCG	Along with user preferences there will be presence of side information
Aghdam [28]	Hierarchical HMM	Last.fm and Netflix datasets	Music and Movie Recommendations	Precision, F-measure, Recall	Sparsity problem
Sangaiah, A.K *et al.* [30]	PBS	Real-time dataset	Industrial informatics	Search space	Confidentiality issues
Sangaiah, A.K *et al.* [31]	EAM	-	Real-world applications	Query Time	Confidentiality issues
Sangaiah, A.K *et al.* [32]	WOA	Test data	Real-world applications	Convergence	Computational complexity
Sangaiah, A.K *et al.* [33]	BA	-	Real-world applications	Distance, Priority	Computational complexity
Gupta Garima and Rahul Katarya [36]	EnPSO	MovieLens dataset	Movie Recommendation	RMSE, MSE, MAE	Vast search space
Katarya Rahul, and Yamini Arora [37]	CapsMF	AA (Apps for Android), AIV (Amazon Instant Video)	Product Recommendation	RMSE, MAE	Higher training time
Katarya Rahul [38]	P-distance algorithm	Douban, Movielens and Jester	Movie Recommendation	MAE, RMSE	Sparsity
Katarya Rahul [39]	KM-ABC	Movielens	Movie Recommendation	accuracy, recall, precision, MAE	Only user rating is considered
Yadav *et al.* [40]	FL	-	Review on various FL	-	-
Katarya Rahul [41]	Two-Level MF	Movielens	Movie Recommendation	RMSE, MAE	Sparsity
Katarya Rahul and Om Prakash Verma [42]	FCM	real-world MSNBC dataset	Acquiring user's social information	Accuracy, Time	Reliability issues
Katarya Rahul and Om Prakash Verma [43]	HMRS	Last.fm	Music Recommendations	Recall	Run time complexity
Katarya Rahul *et al.* [44]	Improved RW	Gowalla, Brightkite and Foursquare	Recommend location in social networks	Precision, recall, F1-score	Computational Intelligence and Data mining techniques are not used.
Katarya Rahul *et al.* [45]	HF	Movielens	Movie Recommendation	Precision	Time complexity
Sanchez et al. [46]	NN	DeliveryFood application	Food-delivery	No. of neighbours (k) and percentage of orders (pmin)	Run time complexity
Teixeira et al. [47]	MAS	Real-time dataset	Diabetic-friendly restaurant recommendation	-	Reliability issues

B. Research Questions

The recommended system aims to execute certain tasks in assisting smart recommendations on restaurants. In this work, context aware-RS have played a most significant role in generating recommendations. Thus, some of the following research questions presented are:

- How are the contexts collected?
- Which contexts are used in smart learning?
- What data mining and recommendation techniques used to processes the contexts?
- What are the recommended activities?

The existing approaches utilized for the recommendation system was discussed in Section II with its research gaps. To avoid the issues presented in the existing RS, a new approach is proposed and the research questions are framed which is solved in this presented work (Section III).

III. Proposed Methodology

The recommendation system uses one of the exceptional pieces of the researches, which can be used to offer better solution for challenges also it uses supplementary refined machine learning algorithms for example, ANN. To analyse the performance of the Context-aware recommendation systems a combination of different approaches is used. The initial attempt is to model the criteria rating, approximately powerful machine learning approaches are used for this process. In order to enhance the prediction accuracy the deep learning model is used in this work. This paper make use of deep auto encoder and Softmax regression trained (DAE-SR) with a Jaya optimization (JO) algorithm for providing effective recommendations. We selected the Jaya optimization algorithm, since it has a high tendency to high convergence rates, low training time, and escape local minima compared to other optimization algorithms.

Fig. 1 shows the overall workflow diagram for the proposed strategy. For the web data acquisition, crawler is a significant tool. However the frequent updates of the data sources, distribution channels and web data structures are resulted in high costs of crawler program maintenance and development. From the raw web data the textual information's keywords are extracted by means of measuring keyword comprehensive weights via informative features. Textual information's keywords reflects the ideal way for data mining also establishes the knowledge representation model, after that constructs the index library. At last, for the deep machine learning strategies build a prediction model to obtain relevant recommendations. The prediction is performed based on the strategy named as DAE-SR to obtain effective results. The textual information pre-processing consists of following phases. They are:

- Information Collection and Processing.
- Pre-processing.
- Similarity Calculation.
- Recommendation Prediction.

A. *Information Collection and Processing*

For collecting the news the web crawler provide important theoretical support. Web crawler is a central part of search engines. The use of web crawler is not difficult when getting the news from the internet. Implementing a new crawler and not extending the existing crawler module by the user is the major problem with crawler. It is used to rescue the web pages also attach them into the local source. In a centralized location, collecting and processing the entire contents of web is the general-purpose of web crawlers. Every day a huge volume of web pages are frequently added and data is persistently changing due to the rapid emergence of internet. On the basis of event pattern or event procedure rule, the CEP is said to be a real-time data processing methodology. It is used to retrieve the high level knowledge from the large extent of data. It can also be utilized to analyze the trends, track the data from multiple sources, and patterns. The events that happened in past, are used in any order which can be allowed by CEP.

B. *Textual Information Pre-processing*

Pre-processing strategies named as stop words removal, word extraction, stemming and also term frequency–inverse document frequency (TF/IDF) are the important approaches in textual information pre-processing. Based on the words existing in the user opinion feedback reviews, the words are tokenized, count of frequency of word and stem are done by using the pre-processing strategies. Fig. 2 shows the process of pre-processing.

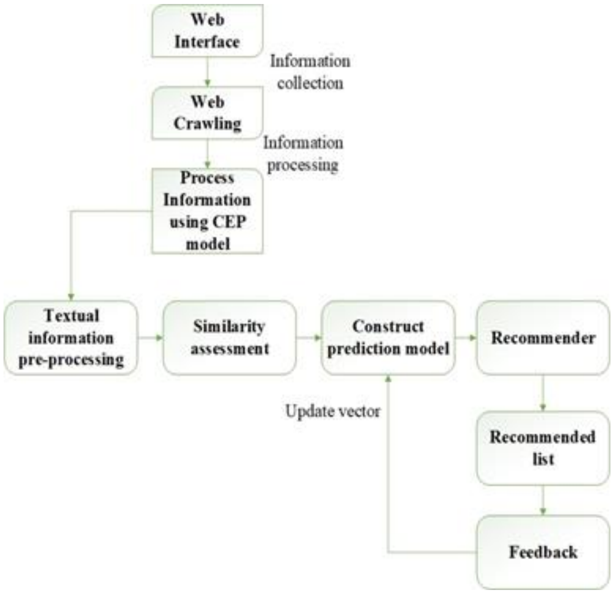


Fig. 1. Workflow diagram for proposed methodology.

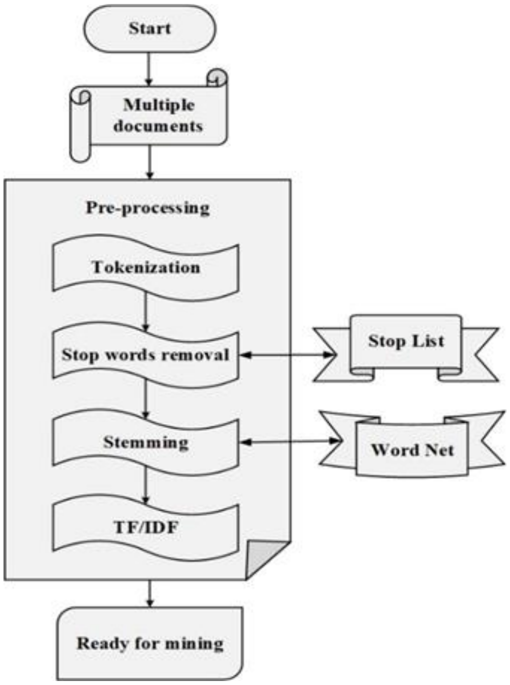


Fig. 2. Model of text-mining pre-processing.

Pre-processing plays a vital role in text mining (TM) applications. Pre-processing is the first step, which involves the combination of other steps such as token extraction, stemming, removal of stop words, and TF/IDF tasks.

1) Word (tokenization) extraction: The procedure of converting the sensitive data into tokens is called as tokenization. Here the significant amount of text structure is reduced to a word, phrase, or symbol. Using NLP (Natural Language Processing), the word in the document set *D* is tokenized.

2) Stop words: Pronouns, prepositions and articles are generally present in documents. These words may not add much meaning and considered as useless. Some of the stop words used are ‘with, an, is, the, has, in, be and, at’ etc. In text mining applications stop words are not considered as keyword, so it is eliminated from the document.

3) Stemming: The process of extracting the base form of words by eliminating the affixes and stem the root of a word into the original word is termed as stemming, example; programs, programming, programmed → ‘program’ etc.

4) Term frequency–inverse document frequency computation (TF-IDF): For text mining, TF-IDF is commonly applied as a weighting factor. A word how important to a document is given in this method; it is also known as numerical statistics.

a) Term weighting systems: Term weights schemes i.e. Document Frequency (DF), Term Frequency (TF) and Inverse Document Frequency (IDF) are frequently used. For the arithmetic representation of collection of web pages, the VSM (Vector Space Model) is the best widely held and proficient approach. Each web page is deliberated as a vector of terms in VSM model such that $d=(t_1,t_2,......,t_n)$, and an equivalent weights vector $w=(w_1,w_2,......,w_n)$, where $w_{1,}w_2,.......w_n$ the weights of $t_1,t_2,......,t_n$ are correspondingly, based on the used term weighting system. Table II gives the example of two dimensional based VSM model.

b) Term frequency (TF): In the web page, TF is related through the weight of a definite term. Normalized TF is measured in this paper. By finding the amount of raw frequency of the term t in web page w is calculate weight of term t is displayed in Eq. (2). The square root of the addition of the square of frequencies of all terms in the web page is known as Euclidean norm.

$$TF_{t,w} = \frac{fr_{t,w}}{\sqrt{\sum_{t=1}^{n} fr_{t,w}^{\;2}}} \tag{1}$$

Table III shows the example of collection of web pages, in this highest TF value is mentioned as bold and underlined cells is the second highest value in the matrix. As shown in Table III, on page 3 the term ‘payment’ has the highest *TF* value (0.7071). The term ‘payment’ is placed two times in page 3 and 4. The TF value is normalized to webpage’s Euclidean norm.

TABLE II. TWO DIMENSIONAL MATRIX BASED VSM MODEL

Web Page \Term	**account**	**service**	**banking**	**payment**	**cheque**	**insurance**	Page Length*
Page 1		1		1		1	3
Page 2	1		1		1	1	4
Page 3	1	1		2	1	1	6
Page 4	1		1	2	2	1	7

Term Frequency **	3	2	2	5	4	4
Page Frequency ***	3	2	2	3	3	4

* The number of distinctive terms in the web page
** Summation of term occurrences in entire web pages.
*** The number of web pages in which the term appears.

TABLE III. COLLECTION OF WEB PAGE AND HIGHEST TF VALUE

Web Page \Term	**account**	**service**	**banking**	**payment**	**cheque**	**insurance**
Page 1		0.5773		0.5773		0.5773
Page 2	0.5000		0.5000		0.5000	0.5000
Page 3	0.3535	0.3535		0.7071	0.3535	0.3535
Page 4	0.3015		0.3015	0.6030	0.6030	0.3015

c) Document frequency: The number of web pages within the collection is represented as DF (weight of term *t*), here the term *t* is found. It is a global weighting term.

$$DF_t = \sum_{p=1}^{N} \begin{cases} 1 & t \in p \\ 0 & t \notin p \end{cases} \qquad (2)$$

TF–IDF: The terms within the collection of web pages are denoted by TF-IDF, which is an effective ranking measure. It also reproduces the statement that less frequent word in the collection is more significant word in web page and viz. The dot product of TF and IDF of a term is denoted as TF-IDF. Table IV gives the example on the basis of TF–IDF weight scheme.

$$TF - IDF_{t,w} = TF_{t,w} \times IDF_t \qquad (3)$$

Where

$$IDF_t = \log\left(N \middle/ DF_t \right) + 1$$

The term ‘payment’ appears three times in the document, so it has highest TF-IDF value shown in Table IV. Also, the term ‘banking’ and ‘service’ appears twice in the collection.

C. Similarity Calculation

After the text pre-processing the opinions are grouped together according to their similarities, while the huge number of reviews are received. Similarity measurement is necessary in text mining approaches. Similarities among the string are calculated by similarity measures. Dice’s coefficient, Jaccard similarity coefficient and cosine similarity are defined below:

1) Dice’s coefficient: This metric is utilized to measure the documents similarity, and the inventive formula is given in Eq. (4).

$$SQ = \frac{2|B \cap C|}{|B| + |C|} \qquad (4)$$

Where, $|B|$ *and* $|C|$ signifies the number of terms present in documents. *SQ* represents the similarity quotient.

2) Cosine Similarity (CS): Two vectors of an inner product space similarity are calculated by this method, cosine of the angle is measured. Cosine of 0° value is 1, for some other angle it is <1. Mostly, in positive space cosine similarity is used, its results efficiently limited in (0, 1). Cosine Similarity [CS or *cos(θ)*] can be computed using the following eqn. 5 & 6,

$$CS = cos\,\theta = \left\{ \frac{Dot\ product(V,M)}{\|V\| * \|M\|} \right. \qquad (5)$$

$$cos\,\theta = \frac{\sum_{i=1}^{n} V_i M_i}{\sqrt{\sum_{i=1}^{n} V_i^2}\sqrt{\sum_{i=1}^{n} M_i^2}} \qquad (6)$$

Where, V_i and M_i signifies the components of vector V and M.

3) Jaccard Similarity (JS) Coefficient: JS is utilized to match the diversity and similarity of model sets. The other names are Jaccard coefficient/ index. The size of intersection divided by the size of union of the sample sets is known as Jaccard coefficient and computes the similarity among finite set of samples as shown in Eq. (7),

$$JS(B,C) = \frac{|B \cap C|}{|B \cup C|} \qquad (7)$$

Where, B and C represent the dual sets.

D. Recommendation Prediction: Hybrid DAE-SR Model

The performance of proposed recommendation model is investigated via neural systems to analyse the textual item content. In this paper, we proposed Softmax regression-based deep auto encoder network (DAE-SR) model trained with Jaya Optimization (JO) algorithm. Compared to the existing models activated in RS, in which the DAE provides better outcomes because of its high capability to reconstruct the inputs. Meanwhile, the softmax regression is utilized for classifying the ratings. We picked the JO algorithm since it has a high propensity to escape nearby minima, low preparing occasions, high assembly rates contrasted with other enhancement calculations, which is utilized to produce improved recommendation prediction.

TABLE IV. EXAMPLE ON THE BASIS OF TF–IDF WEIGHT SCHEME

Web Page \ Term	account	service	banking	payment	cheque	insurance
Page 1		0.7510		0.6494		0.5773
Page 2	0.5624		0.6505		0.5624	0.5000
Page 3	0.3976	0.4599		**0.7954**	0.3682	0.3535
Page 4	0.3391		0.3922	0.6783	0.6783	0.3015

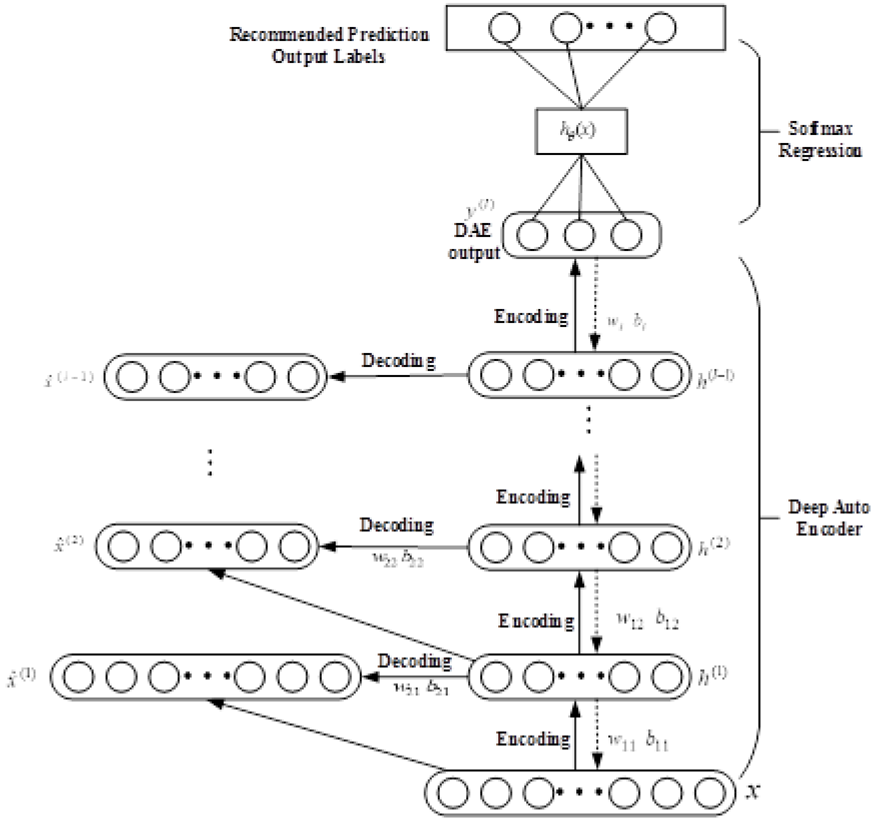


Fig. 3. Architecture of DAE-SR model.

Fig. 3 represents the architecture of DAE-SR model. It can be seen that DAE and softmax regression can be perfectly linked as a soft-hybrid model, which can make full use of labelled instances and achieve supreme accuracy with Jaya optimization algorithm. Here, we mainly explain the hybrid model of DAE-SR for recommended prediction. The deep auto encoder which is the popular deep learning model can learn discriminative features to produce autonomous distributed depictions of contents of items and users. Table V signifies the details of hyperparameters.

TABLE V. HYPERPARAMETER DETAILS

Hyperparameter	Value
No. of hidden layers	3
Batch size	16
Number of epochs	50
Activation	softmax
Learning rate	0.001
Optimizer	Jaya Optimization (JO)
Epoch	100

A DAE is said to be a deep neural network, which is fabricated via stacked Auto encoders. Pre-training that stacked AE from the bottom to top is a major phase of training a DAE. The result features of the hidden layer designated as input of the top-layer of the DAE, when the constraints of bottom layer of the DAE are decided. Through the Jaya optimization algorithm the procedure of the error back propagation is originated to fine-tune the constraints of the complete network. The number of layer fix as l also the function of objective can be displayed as in Eq. (8):

$$s\left(w_l, w_{i,k}\,|_{k=1}^{l-1}, b_{i,k}\,|_{k=1}^{l-1}\right) = \underset{w_l, w_{i,k}, b_{i,k}}{\arg\min} \frac{1}{2N}\sum_{i=1}^{N}|y_i - g_l(f_l(h_i^{l-1}))|^2 \tag{8}$$

Where, activation esteem of $(l-1)^{th}$ hidden layer indicated by $h_i^{l-1} = f_{l-1}(f_{l-2}(\cdots f_1(x_i)))$, and y_i signifies the label of x_i. w_l is the final layer weight, $w_{i,k}$ and $b_{i,k}$ represent the weight and bias of k^{th} layer, correspondingly. Moreover, the constraints of DAE are restructured as:

$$w_l := w_l + \Delta w_l = w_l - \mu d^l h^{l-1} \tag{9}$$

$$w_{1,k} := w_{1,k} + \Delta w_{1,k} = w_{1,k} - \mu d^k h^{k-1} \tag{10}$$

$$b_{1,k} := b_{1,k} + \Delta b_{1,k} = b_{1,k} - \mu \sum_{j=1}^{R} d^k \tag{11}$$

Where, the value $d^l = (h^l - Y)h^l(1-h), d^k = w^t_{k1} d^{k+1}(1-h^k)(when\, k < l)$, learning rate is represented as μ and size of input features denoted as R. The repetitions of $w_{1,k}$ and $b_{1,k}$ are performed till the function of objective reaches the max-epoch/iteration. Hence, the output of DAE is given to the SR model for further processing. Softmax Regression (SR) is a single-layer neural network used to evaluate the conditional probabilities linked with all the possible number of classes. A logistic regression display is the most straight forward type of a neural system. It comprises of an info layer with numerous traits and a bias unit, and just a single yield layer, or class. It is basically a double classifier. The Softmax Regression calculation applies binary strategic regression to different classes without a moment's delay. Let $D = \{(z^{(1)}, y^{(1)}, ..., (z^{(n)}, y^{(n)})\}$ be the training set, $z^{(i)}(i \in \{1,2,....,n)\}$ signifies the training data, $y^{(j)}(j \in \{1,2,.....,n\})$ signifies the label of training data. Given a test instance z, to evaluate the posterior probability $p(y = k \mid z)$. Softmax regression consists of input, classifier and output. A Softmax function is defined as the following Eq. (12):

$$P\left(y = \frac{j}{z^{(i)}}\right) = \phi_{soft\max}\left(z^{(i)}\right) = \frac{e^{z^{(i)}}}{\sum_{j=0}^{k} e^{z_k^{(i)}}} \quad (12)$$

Where,

$$z^{(i)} = w_0x_0 + w_1x_1 + ... + w_mx_m = \sum_{l=0}^{m} w_l x_l = w^T x \quad (13)$$

The Softmax function computes the probability that the training sample x_i belongs to class j given the weight and net input z_i. The process of finding the optimal weight parameters is the need for training DAE which minimize the objective function, where the models are trained by JO algorithm [29]. Compared to other optimization algorithms, this JO algorithm is said to be a powerful global optimization approach which can be used to solve the constraint as well as unconstrained problems using the benchmark function. Some of the advantages of JO algorithm are: It does not have any specific algorithm parameters to conduct the real computational experiments and has better algorithm convergence. The supreme reason to choose JO is the victorious nature of this algorithm which makes it more powerful than other evolutionary and swarm intelligence algorithms. The main objective of JO algorithm is, for a particular problem if the solution is once achieved the optimal result is reached simultaneously thereby avoiding the worst result. JO algorithm is most pretentious in application viewpoint and endeavours extraordinary accomplishments to discover the genuine solution successfully. Compared to other optimization procedures this JO algorithm utilizes only dual parameters such as number of iterations and size of population. An imperative benefit of using this algorithm is that it minimizes the time required for the optimization process and ignores the endeavour of varying constraints. Table VI specifies the pseudocode of Jaya Optimization.

TABLE VI. PSEUDOCODE OF JAYA OPTIMIZATION

```
Jaya optimization algorithm procedure
Initialize: population size P, number of iterations I
Initialize P solutions randomly
Find $f(z_i)\ \forall i = 1,2,...,P$
Sort the population as $z_1$ -best solutions, $z_P$ -worst solutions
t=1
while $(t \le I)$ do
    for $i = 1,...,P$ do
        for $j = 1,...,D$ do
            Set $rn_1 \in [0,1]$
            Set $rn_2 \in [0,1]$
            $z'_{i,j} = z_{i,j} + rn_1 \cdot (z_{best,j} - |z_{i,j}| - rn_2 \cdot (z_{worst,j} - |z_{i,j}|))$
        end for
        if $f(z'_i) \le f(z_i)$ then
            $z_i = z'_i$ (Update)
        end if
    end for
    $t = t + 1$
end while
```

Suppose if the objective function is $f(z)$ with dimensional D factors represented as $(j = 1,2,..,D)$and the estimation value is $z_{i,j}$ for j^{th} variable of i^{th} competitor solution. The position of i^{th} solution candidate is $z_i = (z_{i,1}, z_{i,2}, ..., z_{i,D})$. The solution of best competitor can be expressed as $z_{best} = (z_{best,1}, z_{best,2}, ..., z_{best,D})$ has the finest estimation of $f(z)$in the present population whereas the solution of worst competitor in the present population is $z_{worst} = (z_{worst,1}, z_{worst,2}, ..., z_{worst,D})$. The solution of $z_{i,j}$ is simplified using the expression,

$$z'_{i,j} = z_{i,j} + rn_1 \cdot (z_{best,j} - |z_{i,j}| - rn_2 \cdot (z_{worst,j} - |z_{i,j}|)) \quad (14)$$

Where, the values of best and worst solutions of the j^{th} variable can be represented as $z_{best,j}$ and $z_{worst,j}$. The dual random numbers are rn_1 and rn_2 within the range [0,1], the updated solution is $z'_{i,j}$ and the absolute value of $z_{i,j}$ is signified as $|z_{i,j}|$. In each iteration, the attraction towards the best solution is denoted as$(z_{best,j} - |z_{i,j}|)$. Once the finest solution is achieved via the JO algorithm, it moves towards the finest outcome neglecting the worst solution. Thus, the proposed DAE-SR model obtains top recommendations on the restaurant names by considering the user reviews. Here, DAE-SR is trained on a data. This network gains knowledge from this data, which is compiled as "weights" of the network. These weights optimized with the JO algorithm have improved the performance of DAE-SR network in recommendation system. This optimization has good exploration and exploitation ability and hence avoid local optima problem with good convergence rate. The method presented in this paper gives a new way of improving the accuracy of the system.

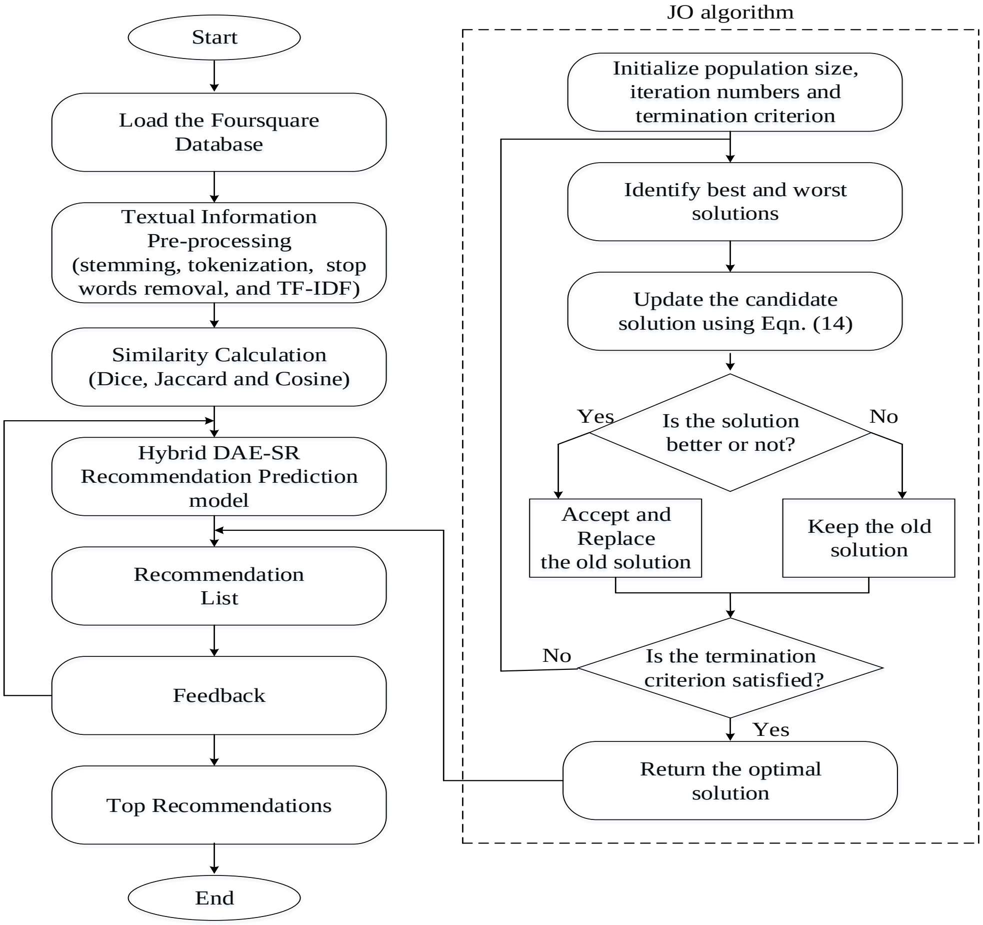

Start
Load the Foursquare Database
Textual Information Pre-processing (stemming, tokenization, stop words removal, and TF-IDF)
Similarity Calculation (Dice, Jaccard and Cosine)
Hybrid DAE-SR Recommendation Prediction model
Recommendation List
Feedback
Top Recommendations
End
JO algorithm
Initialize population size, iteration numbers and termination criterion
Identify best and worst solutions
Update the candidate solution using Eqn. (14)
Is the solution better or not?
Yes
No
Accept and Replace the old solution
Keep the old solution
Is the termination criterion satisfied?
No
Yes
Return the optimal solution

Fig. 4. Flowchart of proposed model.

Fig. 4 displays the flowchart of proposed model. The process starts by collecting the data from the Foursquare dataset. Next step is pre-processing of textual information with the stemming, tokenization, stop words removal and TF-IDF approaches. Three different similarity measures such as dice, Jaccard and cosine metrics are used to evaluate the similarity among the features. Among the three measures, cosine similarity has gained improved performance which can be used for further processing. Finally, the recommendation prediction is performed with the DAE-SR machine learning model. Here, the proposed DAE-SR model is optimized using JO algorithm that generates optimal best solutions with the feedback model and the Top recommendation on restaurant names is displayed. Some of the advantages of the proposed work are:

- Savings in cost and processing time.
- No cold-start issue and work well with sparse datasets.
- Quicker than traditional recommendation systems.
- Effective decision making and.
- Useful for people who visits new places.

E. *Answers to the Research Questions*

The presented context-aware recommendation model extracts valuable information from learner contexts and also leads to obtain better recommendations. In this wok, the contexts were collected using Search engine (Foursquare dataset). Next, the proposed RS named (DAE-SR) adopts a user preference model by using features, e.g., restaurant ratings, location, current weather, and time of the day. The recommendation technique used to process the context is DAE-SR with JO algorithm. The main recommended activity of the proposed model is to generate relevant and personalized recommendation about restaurants.

F. *Application of the Proposed Work*

Restaurant Recommendation is very useful to users because users can get information easily about different restaurants worldwide. These recommendations offer information's like popular restaurant names, menus, location, phone numbers and directions at a glance via the user's smartphone. Nowadays, there are number of social networking sites in which users post reviews about restaurants and rate businesses. However, before the creation of these online sites people went to restaurants occasionally visiting an interesting place or hearing reviews from neighbours, friends etc. With the technological advancements, there are number of online websites in which the people can check the details about the restaurant before visiting it directly. The information can be available in the form of ratings, price ranges, reviews, and hours of operation. After analysing all these factors, users can make valuable decisions. Checking the information via these online sites seem to be a time consuming task since users have to visit number of pages about different recommendations in order to find the right restaurant of their choice. Above all, when people visit a new place, travel or move to new place users prefer the same search procedure again. Therefore, the proposed DAE-SR model offers an effective recommendation system for the selection of restaurants. Using this we can build a customized real-time restaurant recommendation system. With this proposed system, we can provide recommendation for restaurants that will suit people's preference. For example, consider a newcomer who has just visited Chennai. He wants to have *'pricey but good selection of beer snobs'*. He is unsure about the good restaurants around his location. By inputting, this search query into our proposed model, he can get the top restaurants nearer to his location that are highly rated and get him interested in using our recommendation system as a user. Fig. 11 shows the best example as the application of the work in real world. Experimental results have shown that the proposed DAE-SR model achieved better performance than the existing recommendation procedures.

IV. Experimental Results

The recommender system helps the users in an effective manner to obtain useful and personalized information, in order to make comprehensive decisions in the day to day life of consumers. In experimental analysis, enactment of proposed DAE-SR is associated with the existing classifiers for example, K-Nearest Neighbour (KNN), Deep Neural Network (DNN) and Artificial Neural Network (ANN). For evaluating the proposed approach, the data is being collected from the public source and the dataset is named as Foursquare dataset. The restaurant reviews are discussed in this dataset. It contains restaurant name, location, day and time and the reviews. In terms of accuracy, precision and recall the proposed strategy is estimated. The implementation tool used in this work is PYTHON. The performance is compared based on the similarity. Dice's coefficient, Cosine similarity and Jaccard Similarity Coefficient are the tree similarity methods, which are used in the proposed strategy. Cosine similarity accuracy is better than the other similarity.

A. Description about Dataset

The dataset used for the implementation process is the Foursquare dataset. This is a publicly available data source being collected from https://www.kaggle.com/danofer/foursquare-nyc-rest?select=readme.txt. However, this dataset covers the details of check-in, tag and tip data of the restaurant locations in NYC gathered from October 24, 2011 to February 20, 2012. The foursquare dataset comprises of 3,112 users, 3,298 venues (locations) with 27,149 check-ins and 10,377 tips. Customer satisfaction can be analysed based on the reviews of reviewers acquired by the restaurants. Hence, the successful growth of restaurants is mainly based on the reviewer's reviews. It contains the details such as restaurant name, location, day and time and the reviews. Here, 70% of data is used for training and remaining 30% of data is used for testing purpose.

B. Evaluation Metrics

The procedure of analysing, collecting and reporting the data regarding the system performance is termed as performance evaluation. The performance of machine learning models is measured by means of the suitable evaluation metrics such as accuracy, recall and precision.

1) Precision value: It is the ratio of total number of recommendations that are relevant among the number of total recommendations provided.

$$Pr ecision = \frac{TP}{TP + FP} \tag{15}$$

2) Recall value: It is the ratio of number of recommendations provided among the total relevant recommendations.

$$Re call = \frac{TP}{TP + FN} \tag{16}$$

3) Accuracy value: This metric gives the required related documents from the total documents. The accuracy metric tries to evaluate the effective decision making of recommendation algorithms. It can be used to estimate the amount of correct as well as incorrect classifications as relevant or irrelevant items which can be predicted by the recommender system and hence useful for user tasks such as finding good items.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{17}$$

(TP- True positive, TN- True Negative, FP- False Positive, FN- False Negative).

C. Performance Analysis

The analysis of proposed DAE-SR is shown and compared with DNN, ANN, KNN for Dice's Coefficient, Jaccard similarity and cosine similarity are explained in this section.

Table VII gives the performance value of DNN, ANN, KNN and proposed DAE-SR for Dice's Coefficient. Fig. 5 displays the comparison analysis of dice similarity coefficient. For DNN, ANN, KNN and proposed DAE-SR classifier the precision values obtained are 85.4, 88.5, 91 and 95.16%, followed by recall value 80.2, 84.9, 87.9 and 94.22% respectively. The accuracy values for DNN, ANN, KNN and proposed DAE-SR classifier are 85.3, 90, 90.4, and 94.66%. Compared with the existing techniques our proposed approach gained better performance.

TABLE VII. DICE'S COEFFICIENT BASED PERFORMANCE

Performance Metrics	KNN (%)	ANN (%)	DNN (%)	DAE-SR (Proposed) (%)
Precision	85.4	88.5	91	95.16
Recall	80.2	84.9	87.9	94.22
Accuracy	85.3	90	90.4	94.66

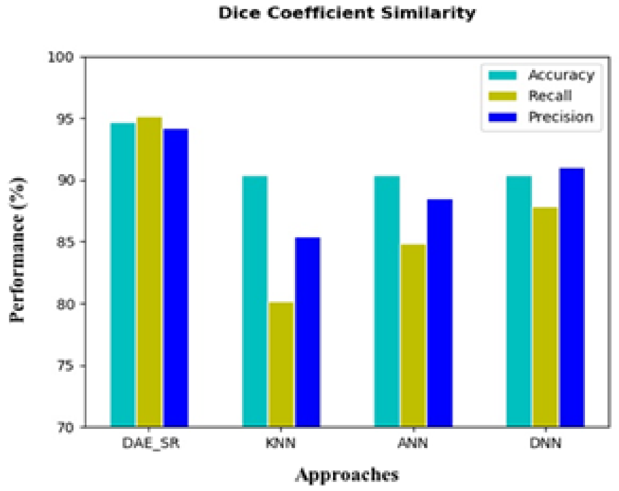


Fig. 5. Performance analysis of DNN, ANN, KNN and proposed DAE-SR for dice's coefficient.

TABLE VIII. PERFORMANCE ANALYSIS VALUE OF JACCARD SIMILARITY

Performance Parameter	KNN	ANN	DNN	DAE-SR (Proposed)
Precision	87.9	90	92.3	97
Recall	83.4	86.2	89	96.7
Accuracy	86.7	91.5	93	97.25

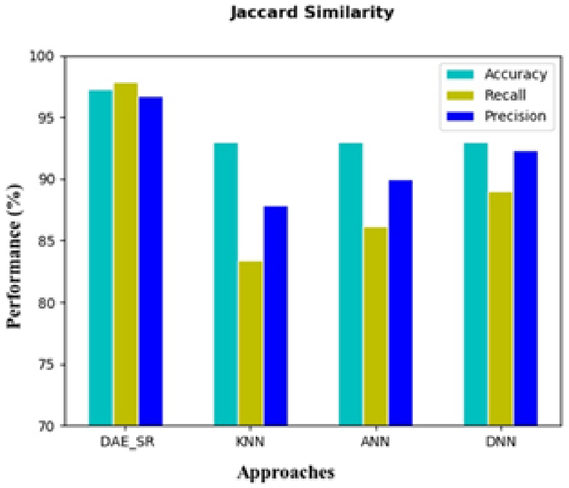


Fig. 6. Graphical representation of jaccard similarity performance.

Table VIII represents the Jaccard Similarity performance value comparison of different classifiers with proposed classifier. Fig. 6 shows the analysis of DNN, KNN, ANN and proposed DAE-SR for Jaccard similarity. For DNN, ANN, KNN and proposed DAE-SR classifier precision value are 87.9, 90, 92.3 and 97%, recall value are 83.4, 86.2, 89 and 96.7% also accuracy value are 86.7, 91.5, 93 and 97.25%. It is obviously agreed, compared with other classifier the proposed approach has better performance.

TABLE IX. PERFORMANCE OF COSINE SIMILARITY

Performance Metrics	KNN	ANN	DNN	DAE-SR (Proposed)
Precision	90	92	94	98
Recall	85	88	92	98.1
Accuracy	89	93	96	98.33

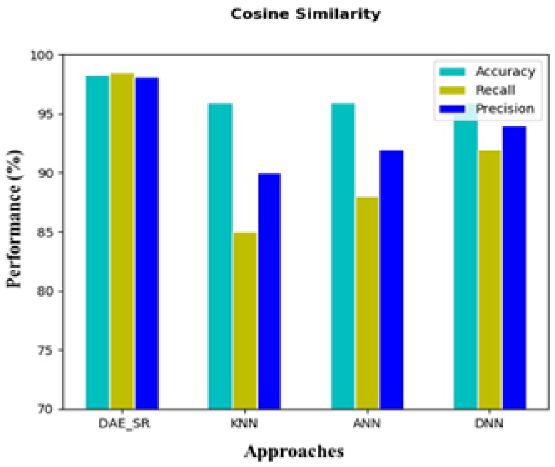


Fig. 7. Graphical representation of DNN, ANN, KNN and proposed DAE-SR for cosine similarity performance.

Table IX signifies the performance of Cosine similarity. Fig. 7 displays the analysis of DNN, ANN, KNN and proposed DAE-SR classifier on cosine similarity. The accuracy acquired with DNN, ANN, KNN and DAE-SR is 89, 93, 96 and 98.33% respectively. From the above results, it is finely known compared with other classifier the proposed DAE-SR approach achieved better performance. Also, cosine similarity based performance is high when compared to the other similarity. As in our proposed method, the generating value accuracy is better in Cosine similarity. Table X represents the comparative results of RMSE and MAE. The proposed DAE-SR model shows lower error values which clearly signify the accurateness of the system compared to the existing methods.

Fig. 8 displays the comparison outcomes of RMSE and MAE of the proposed approach with existing approaches. Minimum or decreased in RMSE and MAE is considered as better outcome. Error minimizing gives better accuracy for the proposed scheme compared to other approaches. In figure 8, compared to two existing approaches, the proposed strategy produced minimum error outcomes.

TABLE X. COMPARATIVE RESULTS OF RMSE AND MAE

Approaches	RMSE	MAE
DAE-SR (Proposed)	0.15	0.02
Natarajan *et al.* (2020) [25]	0.9	0.8
Iqbal *et al.* (2019) [26]	0.8832	1.092

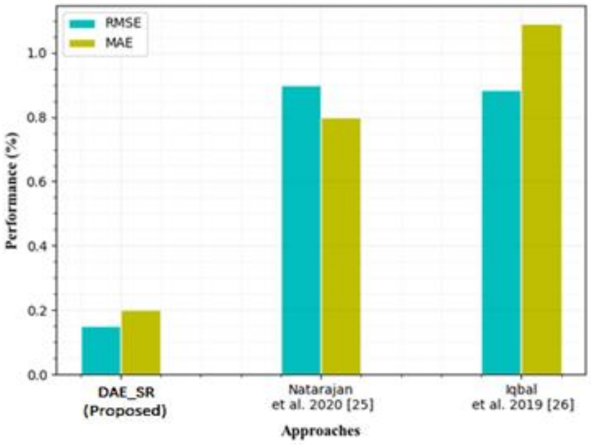


Fig. 8. Comparison of different performance measures.

TABLE XI. COMPARISON OF PERFORMANCE OUTCOMES OVER DIFFERENT EXISTING WORKS

Method/Metric	Precision	Recall	Accuracy
DAE-SR (proposed)	98%	98.1%	98.33%
Natarajan *et al.* (2020) [25]	56.4%	56%	84.1%
Afolabi *et. al.* (2019) [22]	46.9%	42.2%	-

Table XI gives the various performance outcomes comparison regarding different existing works. Fig. 9 shows the precision, recall, and accuracy comparison of DAR-SR with different existing works. In this research, we have used DAE-SR for the effective recommended system. The proposed approach produced better outcomes than the other two previous works. Our proposed model achieves maximum 98% precision, 98.1% recall and 98.33% accuracy values. The proposed scheme achieves above 98% performances because of the best classification approach DAE-SR with JO.

TABLE XII. ACCURACY COMPARISON WITH EXISTING METHODS

Authors	Techniques	Accuracy (%)
Venugopal and Sandhya	**DAE-SR (proposed)**	98.33%
Natarajan *et al.*[25]	RS-LOD	84.1%
Ray *et al.*[34]	Ensemble	92.36%
	RNN	86.00%
	GRU	90.00%
	LSTM	89.00%
	Bi-LSTM	89.00%

Table XII signifies the accuracy comparison with existing methods. The accuracy of the proposed DAE-SR model is compared with various state-of-the-art models to evaluate the performance of the proposed recommendation procedure. The existing models on recommendation system gained lower classification accuracy with RS-LOD, RNN, Ensemble, LSTM, GRU and Bi-LSTM as 84.1%, 86.00%, 92.36%, 89.00%, 90.00% and 89.00% respectively. The proposed model achieved supreme accuracy of (98.33%) because the context-aware RS based on text mining and machine learning algorithms afford quick and correct suggestions to user on good restaurants name. Table XIII specifies the run time comparison with existing methods. From the table it is observed that the proposed DAE-SR recommendation model requires less run time 1.43 milliseconds for computation whereas the other models specify higher computational time which reduces the system efficacy.

TABLE XIII. RUN TIME COMPARISON WITH EXISTING METHODS

Techniques	Run Time
DAE-SR (proposed)	1.43 ms
IDA-CF [35]	2.6 ms
CF [35]	12 sec
NRA [35]	27 sec
FRS [35]	28 sec

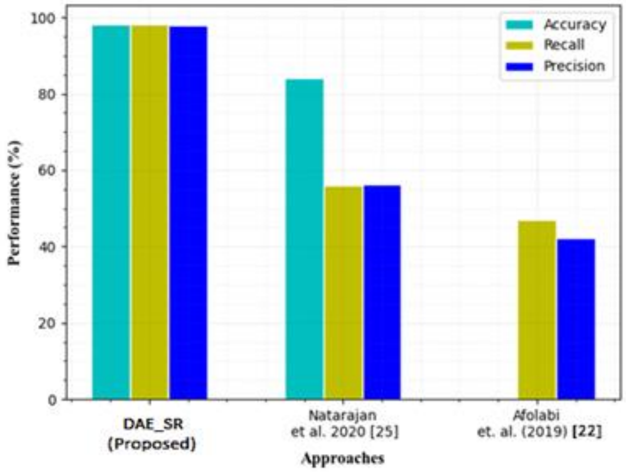


Fig. 9. Performance comparison of DAE-SR.

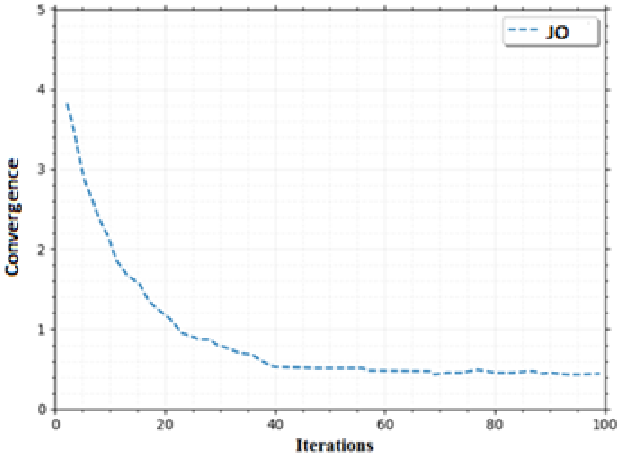
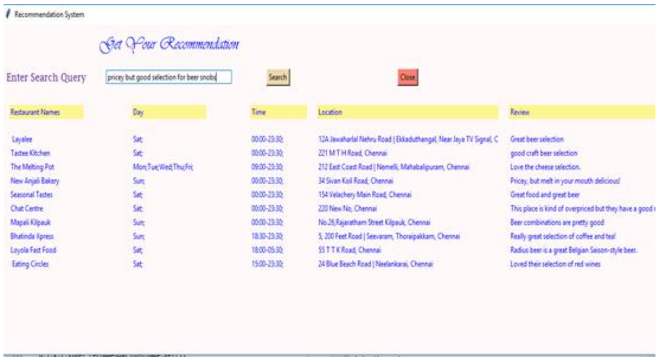


Fig. 10. Convergence graph for JO.

Fig. 10 demonstrated the convergence graph for the JO algorithm. It illustrated that, the proposed JO algorithm achieved the optimum convergence after the 40 iterations, because the JO is a potential optimizer for an engineering issues. Compared to other algorithms, the exploration and exploitation ability of JO algorithm is very impressive and superior. And it has a very good balance among exploitation and exploration.

D. Recommendation Analysis

The following section gives the recommendation analysis in terms of a graphical user interface (GUI). Similarity-based outcomes are taken here for the statistical measures and compared with the existing classifiers. The proposed model DAE-SR achieved better performance, and it produced accurate recommendation outcomes.



Enter Search Query: pricey but good selection for beer snobs

Restaurant Names	Day	Time	Location	Review
Layalee	Sat;	00:00-23:30;	12A Jawaharlal Nehru Road | Ekkaduthangal, Near Jaya TV Signal, C	Great beer selection
Tastee Kitchen	Sat;	00:00-23:30;	221 M T H Road, Chennai	good craft beer selection
The Melting Pot	Mon;Tue;Wed;Thu;Fri;	09:00-23:30;	212 East Coast Road | Nemelli, Mahabalipuram, Chennai	Love the cheese selection.
New Anjali Bakery	Sun;	00:00-23:30;	34 Sivan Koil Road, Chennai	Pricey, but melt in your mouth delicious!
Seasonal Tastes	Sat;	00:00-23:30;	154 Velachery Main Road, Chennai	Great food and great beer
Chat Centre	Sat;	00:00-23:30;	220 New No, Chennai	This place is kind of overpriced but they have a good
Mapali Kilpauk	Sun;	00:00-23:30;	No.26,Rajaratham Street Kilpauk, Chennai	Beer combinations are pretty good
Bhatinda Xpress	Sun;	18:30-23:30;	5, 200 Feet Road | Seevaram, Thoraipakkam, Chennai	Really great selection of coffee and tea!
Loyola Fast Food	Sat;	18:00-05:30;	55 T T K Road, Chennai	Radius beer is a great Belgian Saison-style beer.
Eating Circles	Sat;	15:00-23:30;	24 Blue Beach Road | Neelankarai, Chennai	Loved their selection of red wines

Fig. 11. Content based restaurant recommendation.

Recommendation of the good restaurants name, location and the reviews of the recommender is displayed in this analysis. Fig. 11 shows the content based restaurant recommendation. The recommendation text box shows the restaurant name, location, day and time also the customer reviews. There are large amount of data is there. The user's information that is related to the restaurant is shown by GUI.

E. Discussion

The proposed study introduces a DL based restaurant recommendation system named DAE-SR which extracts user preferences from online comments and recommends top restaurants to the users. With the technological advances and changes in human lifestyle makes the selection of appropriate restaurants as very difficult due to the wide range of ethnicities, ingredients, culinary styles, personal tastes and cultures. The proposed DAE-SAR based recommendation systems offers valuable decisions to users on where to eat. The proposed DAE-SR model attained an increased accuracy of 98.33% whereas the existing models RNN, LSTM, Ensemble, GRU, BiLSTM and RS-LOD gained lower results such as 86%, 89%, 92.36%, 90%, 89% and 84.1% respectively. Thus, the presented model obtained improved accuracy due to better convergence, low run-time and GUI modelling.

V. Conclusion

Recommender system provides precise online recommendations to the web users, it can also be known as knowledge management and information distribution tool. In this paper, a novel distributed framework for context aware recommender system is proposed based on text mining and machine learning algorithms. Based on the user opinion feedback reviews from the social networking services the top restaurants are recommended to the customers. The existing works on recommendation system proved that the use of individual CF based recommendation provide recommendations only via ratings/user interests and items. In this work, a new RS model is presented which proved that the contextual information is more relevant in the generation of recommendations under diverse conditions such as ubiquity and mobility. The proposed DAE-SR classifier offers suitable suggestions to users on the restaurant recommendations compared to the existing DNN, KNN and ANN methods. Experimental outcomes reveal that the proposed DAE-SR based recommendation system outperforms the other classifiers in terms of accuracy, precision, and recall values based on the similarity measures. The accuracy (98.33%) of proposed strategy DAE-SR outperforms the other strategies with cosine similarity. The main advantage of DAE-SR is, it may not suffer from cold-start issue and work well with sparse datasets. The limitation of this study is that only one dataset is used for analysis. Moreover, it is limited by the total amount of available content for providing meaningful recommendation. In future work, we can evaluate our proposed method with different datasets for different applications.

References

[1] C. Rodney, "Modern communication technologies and the new world information order". International journal of communication vol. 2, no. 1, pp. 147-159, 2005.

[2] J. Ahokas and T. Kiiski, "Cybersecurity in ports". Publication of the HAZARD Project vol. 3, 2017.

[3] F. Iannelli, A. Koher, D. Brockmann, P. Hövel and I.M. Sokolov, "Effective distances for epidemics spreading on complex networks". Physical Review E vol. 95, no. 1, pp. 012313, 2017.

[4] R. Pastor-Satorras, C. Castellano, P.V. Mieghem and A. Vespignani, "Rev. Mod. Phys" vol. 87, pp. 925, 2015.

[5] A.S. Manek, P.D. Shenoy, M.C. Mohan and K.R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier". World Wide Web vol. 20, no. 2, pp. 135-154, 2017.

[6] G. Scavo, Z.B. Houidi, S. Traverso, R. Teixeira and M. Mellia, "WeBrowse: Leveraging User Clicks for Content Discovery in Communities of a Place". Proceedings of the ACM on Human-Computer Interaction CSCW vol. 1, no. 93, 2017.

[7] D.L. Quoc, C. Fetzer, P. Felber, "UniCrawl: a practical geographically distributedweb crawler". In: IEEE International Conference on Cloud Computing, IEEE, pp. 389–396, 2015.

[8] Y. Tang, L. Wei, W. Wang and P. Xuan, "Towards Intelligent Web Crawling–A Theme Weight and Bayesian Page Rank Based Approach". In International Conference on Web Information Systems Engineering Springer, Cham pp. 192-201, 2017.

[9] A. Darer, O. Farnan and J. Wright, "Automated Discovery of Internet Censorship by Web Crawling" 2018. arXiv preprint arXiv: 1804.03056.

[10] P. Bandagale, N.R. Sawantdesai, R.U. Paradkar and P.P. Shirodkar, "Survey on Effective Web Crawling Techniques". International Journal of Computer & Mathematical Sciences IJCMS, vol. 6, no. 10, October 2017.

[11] S.P. Venugopal, "Focused crawling from the basic approach to context aware notification architecture". Indonesian Journal of Electrical Engineering and Computer Science vol. 13, no. 2, pp. 492-498, 2019 10.11591/ijeecs.v13.i2.

[12] F. Xiao, C. Zhan, H. Lai, L. Tao and Z. Qu, "New parallel processing strategies in complex event processing systems with data streams". International Journal of Distributed Sensor Networks vol. 13, no. 8, pp. 1550147717728626, 2017.

[13] G. Cugola and A. Margara, "Processing flows of information: from data stream to complex event processing". ACM ComputSurv vol. 44, no. 3, pp. 15, 2012.

[14] O.J. Lee and J.E. Jung, "Sequence clustering-based automated rule generation for adaptive complex event processing". Future Generation Computer Systems vol. 66, pp. 100-109, 2017.

[15] K. Patroumpas, T.K. Sellis, "Event processing and real-time monitoring over streaming traffic data, in: S.D. Martino, A. Peron, T. Tezuka (Eds.)", Proceedings of the 11th International Symposium on Web and Wireless Geographical Information Systems, W2GIS 2012, in: Lecture Notes in Computer Science, Springer, Naples, Italy vol. 7236, pp. 116–133, 2012.

[16] I. Flouris, N. Giatrakos, A. Deligiannakis, M. Garofalakis, M. Kamp and M. Mock, "Issues in complex event processing: Status and prospects in the big data era". Journal of Systems and Software vol. 127, pp. 217-236, 2017.

[17] P. Gopalan, L. Charlin and D.M. Blei, "Content-based recommendations with poisson factorization". In Proceedings of NIPS, pp. 3176–3184, 2014.

[18] J. He, H.H. Zhuo and J. Law, "Distributed-Representation Based Hybrid Recommender System with Short Item Descriptions" 2017. arXiv preprint arXiv: 1703.04854.

[19] C. Li, G. Chen and F. Wang, "Recommender systems based on user reviews: the state of the art". User Modeling and User-Adapted Interaction vol. 25, no. 2, pp. 99-154, 2015.

[20] P. Bhattacharjee and A. Awekar, "Batch Incremental Shared Nearest Neighbour Density Based Clustering Algorithm for Dynamic Datasets". In European Conference on Information Retrieval, Springer, Cham, pp. 568-574, 2017.

[21] Y. Xia, G.D. Fabbrizio, S. Vaibhav and A. Datta, "A Content-based Recommender System for E-commerce O ers and Coupons" 2017.

[22] I.T. Afolabi, O.S. Makinde and O.O. Oladipupo, "Semantic Web mining for Content-Based Online Shopping Recommender Systems". International Journal of Intelligent Information Technologies (IJIIT) vol. 15, no. 4, pp. 41-56, 2019.

[23] F. Amato, V. Moscato, A. Picariello and F. Piccialli, "SOS: A multimedia recommender System for Online Social networks". Future generation computer systems, Springer 2017.

[24] J.K. Tarus, Z. Niu and A. Yousif, "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining". Future Generation Computer Systems vol. 72, pp. 37-48, 2017.

[25] S. Natarajan, S. Vairavasundaram, S. Natarajan and A.H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data". Expert Systems with Applications vol. 149, pp. 113248, 2020.

[26] M. Iqbal, M.A. Ghazanfar, A. Sattar, M. Maqsood, S. Khan, I. Mehmood and S.W. Baik, "Kernel context recommender system (KCR): A scalable context-aware recommender system algorithm". IEEE Access vol. 7, pp. 24719-24737, 2019.

[27] A. Pujahari and D.S. Sisodia, "Pair-wise Preference Relation based Probabilistic Matrix Factorization for Collaborative Filtering in Recommender System". Knowledge-Based Systems pp. 105798, 2020.

[28] M.H. Aghdam, "Context-aware recommender systems using hierarchical hidden Markov model". Physica A: Statistical Mechanics and its Applications vol. 518, pp. 89-98, 2019.

[29] H.M. Pandey, "Jaya a novel optimization algorithm: What, how and why?" In 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence) IEEE pp. 728-730, 2016.

[30] A.K. Sangaiah, D.V. Medhane, T. Han, M.S. Hossain and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics". IEEE Transactions on Industrial Informatics vol. 15, no. 7, pp. 4189-4196, 2019.

[31] A.K. Sangaiah, D.V. Medhane, G.B. Bian, A. Ghoneim, M. Alrashoud and M.S. Hossain, "Energy-aware green adversary model for cyberphysical security in industrial system". IEEE Transactions on Industrial Informatics vol. 16, no. 5, pp. 3322-3329, 2019.

[32] A.K. Sangaiah, A.A.R. Hosseinabadi, M.R. Shareh, S.Y.B. Rad, A. Zolfagharian and N. Chilamkurti, "IoT resource allocation and optimization based on heuristic algorithm". Sensors vol. 20, no. 2, pp. 539, 2020.

[33] A.K. Sangaiah, M. Sadeghilalimi, A.A.R. Hosseinabadi and W. Zhang, "Energy consumption in point-coverage wireless sensor networks via bat algorithm". IEEE Access vol. 7, pp. 180258-180269, 2019.

[34] B. Ray, A. Garain, and R. Sarkar, "An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews". Applied Soft Computing vol. 98, pp. 106935, 2021.

[35] B. Ramzan, I.S. Bajwa, N. Jamil and F. Mirza, "An Intelligent Data Analysis for Hotel Recommendation Systems using Machine Learning" 2019. arXiv preprint arXiv:1910.06669.

[36] G. Gupta and R. Katarya, "EnPSO: An AutoML Technique for Generating Ensemble Recommender System." Arabian Journal for Science and Engineering pp. 1-19, 2021.

[37] R. Katarya and Y. Arora, "Capsmf: a novel product recommender system using deep learning based text analysis model". Multimedia Tools and Applications vol. 79, no. 47, pp. 35927-35948, 2020.

[38] R. Katarya, "Reliable Recommender System Using Improved Collaborative Filtering Technique". System Reliability Management: Solutions and Technologies vol. 113, 2018.

[39] R. Katarya, "Movie recommender system with metaheuristic artificial bee". Neural Computing and Applications vol. 30, no. 6, pp. 1983-1990, 2018.

[40] D.K. Yadav and R. Katarya, "Study on Recommender System using Fuzzy Logic". In 2018 Second International Conference on Computing Methodologies and Communication (ICCMC), IEEE, pp. 50-54, 2018.

[41] R. Katarya, "An improved recommender system using two-level matrix factorization for product ontologies". In 2017 International Conference on Intelligent Sustainable Systems (ICISS), IEEE, pp. 223-226, 2017.

[42] R. Katarya and O.P. Verma, "An effective web page recommender system with fuzzy c-mean clustering". Multimedia Tools and Applications vol. 76, no. 20, pp. 21481-21496, 2017.

[43] R. Katarya and O.P. Verma, "Efficient music recommender system using context graph and particle swarm". Multimedia Tools and Applications vol. 77, no. 2, pp. 2673-2687, 2018.

[44] R. Katarya, M. Ranjan and O.P. Verma, "Location based recommender system using enhanced random walk model". In 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, pp. 33-37, 2016.

[45] R. Katarya, O.P. Verma and I. Jain, "User behaviour analysis in context-aware recommender system using hybrid filtering approach". In 2013 4th International Conference on Computer and Communication Technology (ICCCT), IEEE, pp. 222-227, 2013.

[46] C.N. Sánchez, J. Domínguez-Soberanes, A. Arreola and M. Graff, "Recommendation System for a Delivery Food Application Based on Number of Orders". Applied Sciences vol. 13, no. 4, pp. 2299, 2023.

[47] B. Teixeira, D. Martinho, P. Novais, J. Corchado and G. Marreiros, "Diabetic-Friendly Multi-agent Recommendation System for Restaurants Based on Social Media Sentiment Analysis and Multi-criteria Decision Making." In Progress in Artificial Intelligence: 21st EPIA Conference on Artificial Intelligence, EPIA 2022, Lisbon, Portugal, August 31–September 2, 2022, Proceedings, Cham: Springer International Publishing, pp. 361-373, 2022.

Convolutional Neural Network Model based Students’ Engagement Detection in Imbalanced DAiSEE Dataset

Mayanda Mega Santoni[1], T. Basaruddin[2], Kasiyah Junus[3]
Doctoral Student, Faculty of Computer Science, University of Indonesia, Depok, Indonesia[1]
Faculty of Computer Science, University of Indonesia, Depok, Indonesia[2, 3]

***Abstract*—The COVID-19 pandemic has significantly changed learning processes. Learning, which had generally been carried out face-to-face, has now turned online. This learning strategy has both advantages and challenges. On the bright side, online learning is unbound by space and time, allowing it to take place anywhere and anytime. On the other side, it faces a common challenge in the lack of direct interaction between educators and students, making it difficult to assess students’ engagement during an online learning process. Therefore, it is necessary to conduct research with the aim of automatically detecting students’ engagement during online learning. The data used in this research were derived from the DAiSEE dataset (Dataset for Affective States in E-Environments), which comprises ten-second video recordings of students. This dataset classifies engagement levels into four categories: low, very low, high, and very high. However, the issue of imbalanced data found in the DAiSEE dataset has yet to be addressed in previous research. This data imbalance can cause errors in the classification model, resulting in overfitting and underfitting of the model. In this study, Convolutional Neural Network, a deep learning model, was utilized for feature extraction on the DAiSEE dataset. The OpenFace library was used to perform facial landmark detection, head pose estimation, facial expression unit recognition, and eye gaze estimation. The pre-processing stages included data selection, dimensional reduction, and normalization. The PCA and SVD techniques were used for dimensional reduction. The data were later oversampled using the SMOTE algorithm. The training and testing data were distributed at an 80:20 ratio. The results obtained from this experiment exceeded the benchmark evaluation values on the DAiSEE dataset, achieving the best accuracy of 77.97% using the SVD dimensional reduction technique.**

Keywords*—*Convolutional neural networks; imbalanced data; deep learning; PCA; COVID-19; online learning; students’ engagement; SVD; SMOTE

I. Introduction

During the COVID-19 pandemic, the education sector has been compelled to adopt online learning. The conventional classroom learning has transformed into online learning or "school from home." E-learning has become a standard solution for learning, and virtual conference technologies, such as Zoom, Google Meet, and others, have given online learning flexibility and accessibility from anywhere and at any time, suitable with the current digital era. However, despite the numerous advantages of online learning, one significant obstacle that needs to be addressed is the lack of direct interaction between teachers and students. During virtual conferences, some students may not turn on their cameras, making it challenging to determine their presence and participation in the online class. Consequently, it becomes difficult for teachers to observe the level of student engagement during online learning, especially during screen sharing to explain teaching materials. This situation presents a common obstacle in online learning. To address this obstacle, it is necessary to conduct research to develop methods of automatic students’ engagement detection during the online learning process.

Students’ engagement detection is an essential factor in improving the learning process. It is a qualitative indicator in the learning process [1]. It entails three structured learning dimensions: behavioral engagement, emotional engagement, and cognitive engagement [2]. While all the three dimensions of engagement are crucial for measuring students' level of involvement in the learning process, emotional engagement is the most widely studied. Detecting students' emotional engagement is particularly important in education because it has a significant impact on their learning rate and overall academic performance. Whitehill et al. [3] showed that both human and automatic engagement judgments are correlated with task performance. The study found that post-test student performance could be predicted based on engagement labels with similar accuracy to pre-test results.

The problem of automatically detecting students’ engagement in online learning based on video data can be solved using a machine learning approach. Zang et al. [3] investigated engagement detection in online learning through a data-driven approach based on facial expressions and mouse usage behavior. Their study demonstrated that utilizing multiple features for detection could significantly improve the accuracy of engagement detection. In contrast to previous studies that solely relied on students’ facial expressions, they also took into account students' mouse usage behavior in their approach. Bhardwaj et al. [4] proposed a deep learning model named Convolutional Neural Network (CNN) for students’ engagement detection, while Selim, et al. [5] conducted students’ engagement detection in online learning using Hybrid EfficientNetB7 together with TCN, LSTM, and Bi-LSTM. Khenkar et al. [6] also proposed an engagement detection method based on micro-body gestures using 3D Convolutional Neural Network (CNN).

Ashwin et al. [7] also conducted engagement detection using CCTV video recordings in a computer laboratory, in which case CCTV video recordings were successfully used to analyze students' engagement. Convolutional Neural Networks (CNN) were successfully implemented with a good level of accuracy in identifying students' engagement levels. This study's results revealed a positive correlation between students' scores (student learning) and students' predicted engagement levels. Meanwhile, Sharma et al. [8] detected students' engagement using video recordings of students' learning through emotional analysis and tracking of eye gaze and head movements based on two machine learning algorithms, namely the Haar Cascade algorithm (for face and eye detection) and the Convolutional Neural Network algorithm (CNN) (for emotion classification). Based on these studies, CNN is a powerful deep learning model that has been successfully used in various studies to detect students' engagement levels in online learning. By analyzing emotional features, tracking eye gaze directions, and estimating head movements, CNN could predict students' engagement levels, which is essential for improving the effectiveness of online learning.

One of the widely used datasets for video-based students' engagement detection is the DAiSEE dataset (Dataset for Affective States in E-Environments). The DAiSEE dataset was first introduced in the study of Gupta et al. in 2016 [9]. The benchmark accuracy value of the DAiSEE dataset for the affective level of engagement was 51.07%. Based on the benchmark evaluation result, there are still many opportunities for improving the classification performance of the DAiSEE dataset. The data distribution for each label of the affective level of engagement is unequal, with 1% for very low engagement, 5% for low engagement, 50% for high engagement, and 45% for very high engagement. This data imbalance can result in errors in the classification model, leading to overfitting or underfitting. One solution to address this issue is to balance the data using undersampling or oversampling techniques [10], [11]. Ali et al. [12] presented a data-level approach and an algorithm-level approach for handling class imbalance problems. Bach et al. [13] examined some undersampling and oversampling methods for highly imbalanced data. The conclusion of their research was that the Synthetic Minority Oversampling Technique (SMOTE) boosted by the Edited Nearest Neighbours (ENN) method allowed for an improvement in classification precision. Fernandez et al. [14] also revealed through their research that the SMOTE algorithm improved performance in supervised learning problems.

Therefore, imbalances in the DAiSEE dataset must be addressed. The current research's objective was to perform data balancing and feature selection to improve the benchmarking performance of the video-based students' engagement detection model on the DAiSEE dataset.

This article is organized as follows. Section II explains related works from previous studies. Section III describes the proposed model and methodology for students' engagement detection. Section IV presents the results of the methodology implementation. Section V provides a discussion of the results, and Section VI presents the conclusions of this study.

II. Related Works

Many studies related to the detection of students' engagement have been carried out. Bhardwaj et al. [4] used two datasets. The first one is the FER-2013 dataset, which is an image dataset used to train the CNN model, and the second one is the MES dataset, which is a tabular dataset used to do weight and subsequent calculations of the MES (Mean Engagement Score). The engagement level of students is classified into two classes: "engaged" and "not engaged." The proposed model achieved an accuracy level of 93.6%, a precision level of 98.48%, and a recall level of 87%. The proposed automated approach will certainly help educational institutions achieve an improved and innovative online learning method.

Selim et al. [5] also used the DAiSEE dataset to detect students' engagement and compared the performance of the proposed method with the VRESEE dataset. They proposed a Hybrid EfficientNetB7 model combined with TCN, LSTM, and Bi-LSTM. EfficientNet was pre-trained on the ImageNet dataset, which includes eight models ranging from EfficientNet B0 to EfficientNetB7. The study also compared the proposed and previous models on the DAiSEE dataset. The results of the three proposed models were as follows: EfficientNetB7+TCN, EfficientNetB7+Bi_LSTM, and EfficientNetB7+LSTM were at the levels of 64.67%, 67.39%, and 67.48%, respectively, outperforming the state-of-the-art ResNet+TCN model that was at 63.59%. When evaluating the proposed models on the VRESEE dataset, the highest accuracy achieved was 94.47% (from the use of EfficientNetB7+Bi_LSTM).

Paidja et al. [15] used the DAiSEE dataset for engagement emotion classification. They proposed a Convolutional Neural Network (CNN) model and performed feature extraction using five facial landmarks and the Euclidean distance between points and center points from the facial image. They also compared CNN with other machine learning algorithms, such as Support Vector Machine (SVM) and Deep Neural Network (DNN). The accuracy results obtained indicated that CNN successfully recognized engagement emotions better than the other methods. However, the limitation of their research was that it did not use the entire DAiSEE dataset as only 77 out of 9068 videos were used.

Abedi et al. [16] described the improvement of the state-of-the-art technology for detecting students' engagement using a ResNet and TCN Hybrid Network. This research also used the DAiSEE dataset and evaluated the performance of the ResNet+TCN method, comparing it to several previous studies on the DAiSEE dataset. The experimental results showed that the proposed ResNet+TCN model could improve the classification accuracy performance by 63.9%. It is very challenging to detect the minority engagement level with a very small sample in a supervised classification problem.

Zhang et al. [17] proposed an Inflated 3D Convolutional Network (I3D) for automatic students' engagement. The research also used the DAiSEE dataset for students' engagement detection, coupled with the use of OpenFace and AlphaPose for feature extraction. The proposed method achieved an accuracy of 52.35%.

Bajaj [18] et al. proposed SOTA hybrid ResNet+TCN for the detection of students’ affective states. They also used the DAiSEE dataset with ResNet for feature extraction and TCN (Temporal Convolutional Network) for classification. The accuracy level reached by the study was 53.6%. The biggest challenge posed by this dataset is high class data imbalance.

Liao et al. [19] used the DAiSEE dataset and presented the Deep Facial Spatiotemporal Network (DFSTN) model for engagement prediction. To extract facial spatial features, they utilized pre-trained SE-ResNet-50 (SENet). The experiment obtained an accuracy of 58.84%.

Hasnine et al. [20] examined the extraction and visualization of students' emotions for engagement detection in online learning. The proposed model for emotion extraction and engagement detection consists of several steps. First, the OpenCV Face Recognition is implemented to detect emotions and eyes. This step results in emotion weight and eye gaze weight. These weights are used to calculate the Concentration Index (CI), which is then used to determine a student's engagement level based on specific rules. If the CI is greater than 65%, the student is detected to be highly engaged. If the CI is between 25% and 65%, the student is considered to be engaged. Otherwise, if the CI is less than 25%, the student is detected to be disengaged.

Brenner et al. [21] presented a social robot system that could detect a person's engagement by utilizing proxemics, body posture, and attention features. The proposed model achieved precision, recall, and F1 score results of 0.81, 0.82, and 0.81, respectively. The intended use of the proposed system is to design robots whose behaviors indicate awareness of a person's engagement.

Previous research commonly used video recording data to detect students' engagement. The DAiSEE dataset is one of the most popular datasets used in previous studies [5], [9], [15]–[19], [22]. Other datasets that have been used for this purpose include the EmotiW 2018 dataset [23], the EmotiW 2020 dataset [24], the Engagement Recognition (ER) database [25], the UPNA head pose dataset [26], and other video recordings [3], [7], [8], [20], [27].

The limitations of previously discussed DAiSEE dataset studies are related to model performance. The performance of detection models such as one with the DAiSEE dataset improved in accuracy from an average benchmark accuracy of 57.9% in 2016 [9] for baseline benchmarking, to 63.9% in 2020 [26], to 67.48% in 2022 [5]. The levels of accuracy are still relatively low, however, so there remain many challenges to overcoming this problem. In addition, the selection of the features to be extracted to increase the model accuracy still needs improvement.

III. Methodology

The research methodology used is illustrated in Fig. 1. It is important to note that the facial images appearing in this discussion were taken from the DAiSEE open dataset developed by [9].

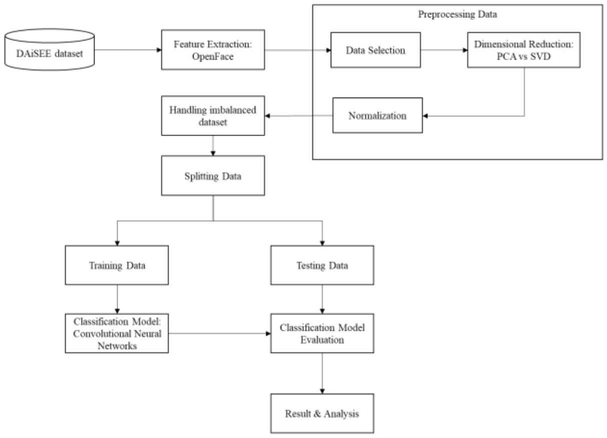

Preprocessing Data
DAiSEE dataset
Feature Extraction: OpenFace
Data Selection
Dimensional Reduction: PCA vs SVD
Handling imbalanced dataset
Normalization
Splitting Data
Training Data
Testing Data
Classification Model: Convolutional Neural Networks
Classification Model Evaluation
Result & Analysis

Fig. 1. Research methodology.

The explanation for each stage in the research methodology is as follows:

A. Dataset

The dataset used in this study was composed of secondary data from an open dataset called DAiSEE (Dataset for Affective States in E-Environments). The dataset was downloaded from https://people.iith.ac.in/vineethnb/resources/daisee/index.html. DAiSEE is a multi-label video classification dataset comprising 9,068 recorded video clips from 112 students, aimed at identifying students' affective states, including boredom, confusion, engagement, and frustration. Each affective state is labeled into four levels: very low, low, high, and very high. The videos were annotated by psychology experts and a crowd. This study focused solely on engagement levels, which were denoted by numbers: 0 for very low, 1 for low, 2 for high, and 3 for very high.

B. Feature Extraction

The subsequent step, feature extraction, was carried out using the OpenFace library. This open-source library is widely used for face recognition purposes, with the capabilities of facial landmark detection, head pose estimation, facial expressions (facial action units) recognition, and eye gaze estimation.

C. Data Pre-Processing

The next stage, data pre-processing, aimed to prepare data for the modeling stage. This stage involved three steps: data selection, feature dimensional reduction, and data normalization. The outcomes of this stage were feature matrices that could be utilized in the subsequent stage.

D. Imbalanced Dataset Handling

The oversampling or undersampling techniques could be employed to address data imbalances, which could lead to prediction errors in the model. Undersampling aimed to balance the data by reducing the number of instances in the majority class to match the number in the minority class. On the other hand, oversampling balanced the data by increasing the instances in the minority class to match those in the majority class.

E. Data Splitting

The feature matrix with balanced data was used in the training and testing processes of the classification model. The training data were used to form the classification model, while the testing data were used to evaluate the performance of the model formulated.

F. Classification Model

Fig. 2 is an illustration of the classification model formulation process. The video recording data collected with feature extraction were used as input data in the CNN classification model. The output of the CNN classification model was the prediction class or engagement level of the input video data.

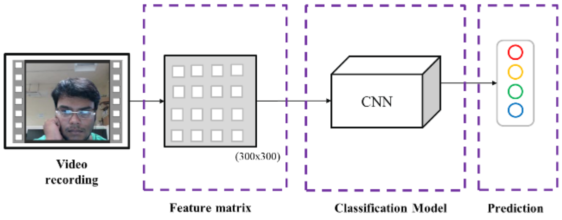


Fig. 2. The classification model formulation process.

G. Classification Model Evaluation

After obtaining the classification model through the training process, the model was tested using the testing data. The test results were then evaluated using metrics such as accuracy, precision, recall, and F1-score. These values were used to determine the performance of the proposed method. To further evaluate the classification model, the confusion matrix was referred to.

The confusion matrix in Fig. 3 is a matrix visualization of the prediction number and the actual data number on the classification model used. True positive (TP) is the number of correctly predicted data in the positive class. False positive (FP) is the number of incorrectly predicted data in the positive class. True negative (TN) is the number of correctly predicted data in the negative class. False negative (FN) is the number of incorrectly predicted data in the negative class [28].

		Prediction	
	Class	Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Fig. 3. Confusion matrix in binary classes.

The accuracy evaluation value compares the number of correctly predicted data with the entire data being tested. It can be calculated using Equation 1 based on the confusion matrix in Fig. 3.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

The precision evaluation value compares the number of correctly predicted data in the positive class with the overall positive predicted results. It can be calculated using Equation 2.

$$Precision\ (Pc) = (TP) / (TP + FP) \quad \square re$$

The recall or sensitivity evaluation value compares the number of correctly predicted data in the positive class with all the actual data in the positive class. It can be calculated using Equation 3.

$$Recall\ (Rc) = (TP) / (TP + FN) \quad (3)$$

Meanwhile, the F1-score evaluation value compares the average weighted precision and recall. It is better in measuring a classification model's performance than the precision or recall value. It can be calculated using Equation 4.

$$F1\text{-}Score = 2 * [(Pc * Rc) / (Pc + Rc)] \quad (4)$$

IV. Results

A. Dataset

The current study used an open dataset named DAiSEE, which is a video dataset that recognizes students' affective levels, including engagement. Each video has a clip ID and engagement level label: very low, low, high, or very high. Fig. 4 shows some examples of the downloaded DAiSEE dataset.

Fig. 4. Some examples of the downloaded DAiSEE dataset.

The data distribution for each engagement level can be seen in Table I. According to the table, the data for each level of engagement were highly imbalanced. Low and very low engagement levels were minority classes with data presentation of 0.7% and 5.1% of the total available data, respectively. If data of this sort are processed, it will cause errors in the classification model due to overfitting. This can be addressed by balancing the data with undersampling or oversampling techniques.

TABLE I. DISTRIBUTION OF DATA ON THE DAISEE DATASET FOR EACH LEVEL OF ENGAGEMENT

Engagement Level	Number of Videos	Percentages
0 (very low)	61	0.7%
1 (low)	455	5.1%
2 (high)	4422	49.5%
3 (very high)	3987	44.7%
Total	8925	100%

B. Feature Extraction

The OpenFace library extracted facial features from each video frame. It can be downloaded on the following GitHub page: https://github.com/TadasBaltrusaitis/OpenFace. Fig. 5 is an example of video output generated from the OpenFace library.

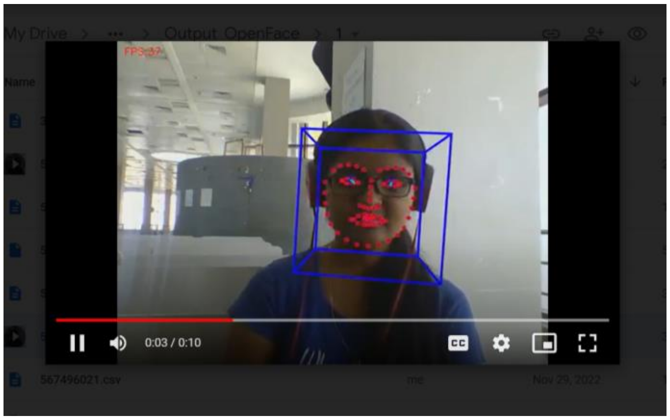


Fig. 5. An example of video output generated from the openface library.

In addition to the video output above, each video generated a CSV file. The file would display the columns frame, face ID, timestamp, confidence, success, and 709 facial feature values covering facial landmark detection, head pose estimation, eye gaze estimation, and estimation of facial expressions in the forms of facial action unit (AUs) features. The CSV file was also modified to store data of file name and the level of engagement for each frame. The DAiSEE dataset comprises 10-second videos with a frame rate of 30 fps, producing 300 frames for each video.

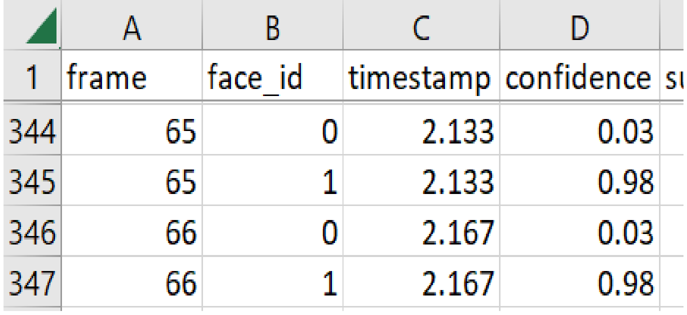
C. Data Pre-Processing

As shown in Fig. 1, there were three pre-processing stages: data selection, dimensional reduction, and data normalization.

1) Data selection: The first data selection stage involved selection of videos to facilitate the computational process. The video selection process was carried out in the following sub-stages:

- Find id_people from the video name (taken from the first five digits)
- Search for unique id_people
- Count and sort in the ascending the number of videos for each unique id_people
- Add up the cumulative value to the threshold = 61 (the total value of the minority class)
- Choose a video name based on the selected unique id_people

The results of feature extraction using the OpenFace library had a confidence column. This column refers to the confidence level of the model as to whether the detected face_id was a face or not. The confidence value ranged from 0 to 1. The closer it was to 1, the more confident the model was that the detected object was a face. On the other hand, the closer it was to 0, the less confident the model was that the detected object was a face.



	A	B	C	D	
1	frame	face_id	timestamp	confidence	su
344	65	0	2.133	0.03	
345	65	1	2.133	0.98	
346	66	0	2.167	0.03	
347	66	1	2.167	0.98	

Fig. 6. An example of a frame that detected two face objects.

As shown in Fig. 6, in frame 65, two objects, face_id 0 and face_id 1, were detected at the timestamp of 2.133 seconds. face_id 0 was detected at a confidence level of 0.03, and face_id 1 was at 0.98. If more than one object was found in a frame, data selection would be performed, where the object with the highest confidence value was to be selected. The data distribution for each engagement level before and after data selection in stages 1 and 2 can be seen in Table II.

TABLE II. DATA DISTRIBUTION BEFORE AND AFTER THE DATA SELECTION PROCESS

Engagement Level	**Imbalanced Data**		**Stage 1 Data Selection**		**Stage 2 Data Selection**	
	A	***B***	***A***	***B***	***A***	***B***
0 (very low)	61	0.7%	61	22.6%	59	23.4%
1 (low)	455	5.1%	63	23.3%	56	22.2%
2 (high)	4422	49.5%	70	25.9%	64	25.4%
3 (very high)	3987	44.7%	76	28.1%	73	29.0%
Total	8925	100%	270	100%	252	100%

A = Number of Videos, B = Percentages

2) Dimensional reduction: The length of the feature vector generated for each frame was very large, i.e., 1x709. Therefore, it became necessary to reduce the dimensions of the features to obtain unique features that could be used as differentiators for each level of engagement. The algorithms used at this stage were PCA (Principal Component Analysis) and SVD (Singular Value Decomposition). The explained variance value refers to the percentage value of the variance from the initial data. The number of components extracted covered a minimum of 80% of the explained variance in the data. In other words, at least 80% of the variance of the data was successfully captured. The greater the value of the explained variance, the better the original data were represented. Based on Table III, component = 300 was chosen because it had the highest explained variance value for both PCA and SVD. In addition, it was chosen so that each video produced would form a feature matrix with a square size of 300 x 300. Thus, using PCA and SVD, the number of features was reduced from 709 to 300.

TABLE III. THE EXPLAINED VALUES OF PCA AND SVD

Number of Components	PCA	SVD
2	81.91727	72.52122
3	96.99637	90.64553
10	99.80054	99.77922
50	99.99682	99.99676
100	99.99963	99.99962
200	99.99996	99.99996
300	99.99998	99.99998

3) Normalization: The feature matrices produced in the dimensional reduction stage had different ranges of values. It became necessary to normalize the data to prevent them from turning into noise in the model training process. The data normalization method used in this independent study was the min-max normalization method. This normalization method produced new feature values that had the same range from 0 to 1.

D. Imbalanced Dataset Handling

Based on Table II, the data for each level of engagement needed to be more balanced. It was necessary to balance the data to avoid overfitting prediction results. SMOTE (Synthetic Minority Over-sampling Technique), which synthesizes new data by re-sampling the minority class data to balance the data to the majority class, was used as an oversampling technique. A comparison was made between the number of data before and after applying SMOTE (see in Table IV).

TABLE IV. THE NUMBER OF DATA BEFORE AND AFTER SMOTE APPLICATION

Engagement Level	Before SMOTE	After SMOTE
0 (very low)	59	73
1 (low)	56	73
2 (high)	64	73
3 (very high)	73	73
Total	252	292

E. Data Splitting

Before entering the training stage of the classification model, the pre-processed data were divided into training and testing datasets, with 80% of the data being used for training and the remaining 20% for testing. The training data were used to form a supervised learning classification model, while the testing data were used to evaluate the classification model. The distribution of training and testing data for each engagement level can be found in Table V.

TABLE V. THE NUMBER OF TRAINING AND TESTING DATA

Engagement Level	Training Data	Testing Data	Total
0 (very low)	58	15	73
1 (low)	59	14	73
2 (high)	58	15	73
3 (very high)	58	15	73
Total Videos	233	59	292

F. Classification Model

The CNN classification model consisted of two stages, namely feature extraction and classification. The former consisted of four convolutional and pooling layer combinations as can be seen in Fig. 7. The feature maps on convolutional layer 1, layer 2, layer 3, and layer 4 were 32, 64, 128, and 256, respectively. The kernel size was 5 with the activation function using "ReLU". The latter used max-pooling with a pooling size of 2 x 2.

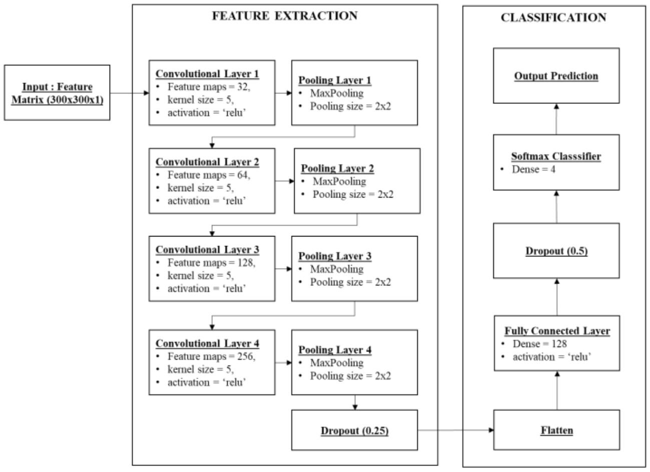


Fig. 7. Classification model of CNN.

The learning parameters used during the model-building process were batch size, epoch, learning rate, and optimizer. The trial-error approach was used to make the parameter selection. Table VI details the parameters of the CNN model used.

TABLE VI. THE AMOUNT OF TRAINING AND TESTING DATA

Parameter	Parameter Values
Number of epochs	800, 1600
Optimizer	Adam
Batch size	32, 16, 8, 4, 2
Learning rate	10^{-5}, 10^{-4}

In learning with artificial neural networks, the best model is often not found in the most recent epoch. Therefore, checkpoints and early stopping are used in the training process. A checkpoint is a CNN model that records each time the loss value decreases by a specified difference. In this way, if the loss value tends to increase or stagnate, the CNN model that manages to achieve the lowest loss value will be stored. Early stopping is a technique to stop the CNN learning process when the loss value has not shown a significant decrease in the number of certain epochs or when the model is said to have converged. This method is used because it can optimize the maximum number of epochs but saves more training time by stopping CNN training when it shows no improvement in learning. In early stopping, there is the patience parameter (p), which is used to determine the conditions for stopping training when it is found that the number of epochs remains the same as there is no decrease in the loss value. The patience value used in this independent study was half the number of epochs.

V. Discussion

Based on the previous discussion, dimensional reduction was carried out with two approaches: PCA and SVD. Therefore, in evaluating this classification model, a comparison was made between the classification models from PCA-reduced data and SVD-reduced data.

Table VII shows the best evaluation value for each experiment. It can be seen that PCA-CNN had the highest accuracy of 72.88% in model 19, with an average accuracy value of 69.66% and parameter values as follows: optimizer = Adam, epoch = 1600, learning rate = 10-4, and batch size = 4. In comparison, model 8 had a higher maximum accuracy of 74.58% but with a standard deviation value greater than that of model 19 (3.54 > 3.34). The smaller standard deviation value was chosen because it means that the accuracy value in the experiment was closer to the average value.

TABLE VII. The Best Evaluation Value for Each Experiment

Parameter/Evaluation	**PCA**				**SVD**			
Model	**2**	**8**	**14**	**19**	**25**	**28**	**35**	**38**
optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Epoch	800	800	1600	1600	800	800	1600	1600
Learning rate	10-5	10-4	10-5	10-4	10-5	10-4	10-5	10-4
Batch Size	16	4	4	4	2	8	2	8
Average Accuracy	60.64	68.14	60.51	69.66	61.02	70.34	64.41	71.02
Standard Deviation	2.76	3.54	4.01	3.34	6.61	3.58	7.19	3.17
Minimum Accuracy	55.93	59.32	55.93	69.66	52.54	64.41	55.93	67.8
Maximum Accuracy	64.41	74.58	67.8	72.88	74.58	76.27	77.97	77.97

For experiments using SVD-CNN, the highest accuracy value was found in model 38, with a maximum accuracy value of 77.97% and an average accuracy value of 71.02%. The best parameter values obtained in this model were as follows: optimizer = Adam, epoch = 1600, learning rate = 10-4, and batch size = 8. Model 38 was found to have the smallest standard deviation value. This model was quite stable in providing accuracy evaluation values from the ten iterations performed for each model.

Regarding the learning rate parameter, the SVD-CNN and PCA-CNN experiments both had a learning rate of 10-4, producing the best accuracy model. Compared to the learning rate of 10-5, the learning rate of 10-4 provided faster computation time because the lower the learning rate, the higher the accuracy of the network, which means that the training process takes longer. For epoch size, the SVD-CNN and PCA-CNN experiments had the same number of epochs, 1600, which produced the best accuracy model. As can be seen in Table VII, when the epoch number of 800 was applied, the optimal accuracy value had yet to be reached. However, in terms of computational time, the larger the epoch number, the greater the time required. If we look at the batch size parameter, the SVD-CNN and PCA-CNN experiments had different parameter values generated by their best models. SVD-CNN had the best batch size of 8, while PCA-CNN had the best batch size of 4. The smaller the batch size, the more batches will be generated, requiring greater computation time. Table VII shows that SVD-CNN was better than PCA-CNN in terms of accuracy, batch size, number of epochs, and learning rate and required shorter computation time.

TABLE VIII. COMPARISON OF PRECISION, RECALL, AND F1-SCORE FOR EACH EXPERIMENT

			Engagement Level				
Experiment	*Model*	*Evaluation*	*0 (very low)*	*1 (low)*	*2 (high)*	*3 (very high)*	*Average*
PCA-CNN	Model 19	Precision	0.62	0.73	0.92	0.73	0.75
		Recall	0.87	0.57	0.73	0.73	0.73
		F1-Score	0.72	0.64	0.81	0.73	0.73
SVD-CNN	Model 38	Precision	0.85	0.83	0.81	0.67	0.79
		Recall	0.73	0.71	0.87	0.80	0.78
		F1-Score	0.79	0.77	0.84	0.73	0.78

Table VIII presents the comparison of precision, recall, and F1-score between PCA-CNN and SVD-CNN experiments. The two best models, models 19 and 38, had precision values higher than the recall values. This indicates false negatives or prediction errors in the actual engagement level data. Meanwhile, the F1-score indicates the average comparison value of weighted precision and recall. In model 19, engagement level 2 (high engagement) had the highest F1-score, i.e., 0.81. This high F1-score indicates that the classification model had fairly good precision and recall values.

Based on the accuracy, precision, recall, and average F1-score values, SVD-CNN performed better than PCA-CNN. Not only were they used for dimensional reduction, SVD and CNN were also used to select important features from the overall features. If analyzed from the variance value generated at the data pre-processing stage, PCA-CNN and SVD-CNN had the same variance value at component value = 300. The higher the variance value, the better the data representation to obtain unique information from the data. Meanwhile, if analyzed from the correlation value between features and engagement level, SVD-CNN had a higher correlation value than PCA-CNN. The features obtained from the SVD results had the best correlation value with the engagement level data. Therefore, in this study, SVD-CNN was superior to PCA-CNN.

The comparison of the values achieved by previous models and those by the model proposed in the DAiSEE dataset can be seen in Table IX. The PCA-CNN and SVD-CNN models with data balancing produced the highest accuracy performance at 72.88 and 77.97, respectively, with fewer features than the previous models.

TABLE IX. COMPARISON OF THE ACCURACY VALUES OF PREVIOUS MODELS WITH THE PROPOSED MODELS IN THE DAISEE DATASET

Model	**Feature (per frame)**	**Feature Dimensions**	**Accuracy**
I3D (Inflated 3D Convolutional Network) [17]	OpenFace (1x600) and AlphaPose (1x36) with feature selection	Not mention	52.35%
SOTA hybrid ResNet+TCN [16]	ResNet	1x512	53.6%
LRCN (Long-term Recurrent Convolutional Networks) [9] – **baseline benchmarking on DAiSEE**	**Not mention**	**Not mention**	**57.9%**
DFSTN (Deep Facial Spatiotemporal Network) [19]	SE-ResNet-50 (SENet)	1x2048	58.84%
ResNet + TCN [16]	ResNet	1x512	63.90%
3D DenseAttNet (DenseNet self-attention neural network) [22]	DenseAttNet	224x224x3	63.59%
EfficientNet B7 + LSTM [5]	EfficientNet B7	1x2560	67.48%
PCA-CNN with balanced data (proposed model)	OpenFace (1x709) with dimensional reduction PCA	1x300	72.88%
SVD-CNN with balanced data (proposed model)	OpenFace (1x709) with dimensional reduction SVD	1x300	77.97%

VI. CONCLUSION

In this study, we have successfully improved the benchmark performance of the DAiSEE dataset. The DAiSEE dataset experienced improvements from an average benchmark accuracy of 57.9% in 2016 [9] for baseline benchmarking, to 63.9% in 2020 [26], to 67.48% in 2022 [5]. We applied data balancing using oversampling and undersampling in the Convolutional Neural Network (CNN) classification model. The DAiSEE dataset also went through the pre-processing stages of data selection, dimensional reduction, and normalization. The features used in this study were taken from the OpenFace library, including 709 facial feature values from facial landmark detection, head pose estimation, eye gaze estimation, and facial expressions (facial action units (AUs)) estimations. Dimensional reduction was performed on the OpenFace features obtained using PCA and SVD techniques. A component number of 300 was applied in the PCA and SVD dimensional reduction, which means that the number of unique features was reduced from 709 to 300.

The PCA-CNN model had the highest accuracy rate of 72.88%, and the SVD-CNN model did 77.97%. The best CNN model parameter values were as follows: learning rate = 10-4, optimizer = Adam, epoch = 1600, and batch size = 4 (PCA-CNN) and 8 (SVD-CNN). The two PCA-CNN and SVD-CNN best models had precision values higher than the recall values (0.75 > 0.73 for PCA-CNN and 0.79 > 0.78 for SVD-CNN). This indicates that there were false negative events such as prediction errors in the actual engagement level data. Meanwhile, the highest F1-score values were 0.73 (PCA-CNN) and 0.78 (SVD-CNN), which shows that the classification models had fairly good precision and recall values.

From all the experiments that have been carried out, it can be concluded that SVD-CNN had better performance than PCA-CNN in evaluating average, maximum, precision, recall, and F1-score values accuracy. If analyzed from the variance value generated at the data pre-processing stage, PCA-CNN and SVD-CNN had the same variance value at component value = 300. The higher the variance value, the better the data representation to obtain unique information from the data. Meanwhile, if analyzed from the correlation value between features and engagement level, SVD-CNN had a higher correlation value than PCA-CNN. Moreover, it can also be interpreted that the features obtained from the SVD results had the best correlation value with the level of engagement contained in the data. Therefore, in the current study, SVD-CNN was superior to PCA-CNN.

Although this study has provided better evaluation results than previous studies on the DAiSEE dataset, there remains a room for improvement for further research. It is necessary to explore alternative approaches to determining the optimal component values to produce features that have a more significant impact on the classification model. Additionally, conducting a more in-depth analysis of features beyond facial expressions can increase the accuracy of students' engagement detection.

Acknowledgment

This research was partially supported by an internal publishing grant from the Faculty of Computer Science, University of Indonesia.

References

[1] F. D'Errico, M. Paciello, and L. Cerniglia, "When emotions enhance students' engagement in e-learning processes," Article in Journal of E-Learning and Knowledge Society, vol. 12, no. 4, pp. 9–23, 2016, doi: 10.20368/1971-8829/1144.

[2] J. A. Fredricks, P. C. Blumenfeld, and A. H. Paris, "School Engagement: Potential of the Concept, State of the Evidence," 2004.

[3] Z. Zhang, Z. Li, H. Liu, T. Cao, and S. Liu, "Data-driven Online Learning Engagement Detection via Facial Expression and Mouse Behavior Recognition Technology," Journal of Educational Computing Research, vol. 58, no. 1, pp. 63–86, Mar. 2020, doi: 10.1177/0735633119825575.

[4] P. Bhardwaj, P. K. Gupta, H. Panwar, M. K. Siddiqui, R. Morales-Menendez, and A. Bhaik, "Application of Deep Learning on Student Engagement in e-learning environments," Computers and Electrical Engineering, vol. 93, Jul. 2021, doi: 10.1016/j.compeleceng.2021.107277.

[5] T. Selim, I. Elkabani, and M. A. Abdou, "Students Engagement Level Detection in Online e-Learning Using Hybrid EfficientNetB7 Together With TCN, LSTM, and Bi-LSTM," IEEE Access, vol. 10, pp. 99573–99583, Sep. 2022, doi: 10.1109/access.2022.3206779.

[6] S. Khenkar and S. K. Jarraya, "Engagement detection based on analyzing micro body gestures using 3D CNN," Computers, Materials and Continua, vol. 70, no. 2, pp. 2655–2677, 2022, doi: 10.32604/cmc.2022.019152.

[7] T. S. Ashwin and R. M. R. Guddeti, "Unobtrusive students' engagement analysis in computer science laboratory using deep learning techniques," in Proceedings - IEEE 18th International Conference on Advanced Learning Technologies, ICALT 2018, Aug. 2018, pp. 436–440. doi: 10.1109/ICALT.2018.00110.

[8] P. Sharma, S. Joshi, S. Gautam, S. Maharjan, V. Filipe, and M. C. Reis, "Student Engagement Detection Using Emotion Analysis, Eye Tracking and Head Movement with Machine Learning."

[9] A. Gupta, A. D'Cunha, K. Awasthi, and V. Balasubramanian, "DAiSEE: Towards User Engagement Recognition in the Wild," Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.01885.

[10] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," ACM Computing Surveys, vol. 52, no. 4. Association for Computing Machinery, Aug. 01, 2019. doi: 10.1145/3343440.

[11] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," GESTS International Transactions on Computer Science and Engineering, vol. 30, 2006.

[12] A. Ali, S. Mariyam Shamsuddin, A. Ralescu, and A. L. Ralescu, "Classification with class imbalance problem: A review," Classification Int. J. Advance Soft Compu. Appl, vol. 5, no. 3, 2013, [Online]. Available: https://www.researchgate.net/publication/288228469.

[13] M. Bach, A. Werner, J. Żywiec, and W. Pluskiewicz, "The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis," Inf Sci (N Y), vol. 384, pp. 174–190, Apr. 2017, doi: 10.1016/j.ins.2016.09.038.

[14] A. Fernández, S. García, F. Herrera, and N. V Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," 2018.

[15] A. N. R. Paidja and F. A. Bachtiar, "Engagement Emotion Classification through Facial Landmark Using Convolutional Neural Network," in Proceedings - 2022 2nd International Conference on Information Technology and Education, ICIT and E 2022, 2022, pp. 234–239. doi: 10.1109/ICITE54466.2022.9759546.

[16] A. Abedi and S. S. Khan, "Improving state-of-the-art in Detecting Student Engagement with Resnet and TCN Hybrid Network," in 18th Conference on Robots and Vision (CRV), Apr. 2021, pp. 151–157. [Online]. Available: http://arxiv.org/abs/2104.10122.

[17] H. Zhang, X. Xiao, T. Huang, S. Liu, Y. Xia, and J. Li, "An Novel End-to-end Network for Automatic Student Engagement Recognition," in IEEE7539th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC), 2019, pp. 342–346.

[18] K. K. Bajaj, I. Ghergulescu, and A.-N. Moldovan, "Classification of Student Affective States in Online Learning using Neural Networks," Nov. 2022, pp. 1–6. doi: 10.1109/smap56125.2022.9942163.

[19] J. Liao, Y. Liang, and J. Pan, "Deep facial spatiotemporal network for engagement prediction in online learning," Applied Intelligence, vol. 51, no. 10, pp. 6609–6621, Oct. 2021, doi: 10.1007/s10489-020-02139-8.

[20] M. N. Hasnine, H. T. T. Bui, T. T. T. Tran, H. T. Nguyen, G. Akçapõnar, and H. Ueda, "Students' emotion extraction and visualization for engagement detection in online learning," in 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Procedia Computer Science, 2021, vol. 192, pp. 3423–3431. doi: 10.1016/j.procs.2021.09.115.

[21] M. Brenner, H. Brock, A. Stiegler, and R. Gomez, "Developing an engagement-aware system for the detection of unfocused interaction," in 2021 30th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2021, Aug. 2021, pp. 798–805. doi: 10.1109/RO-MAN50785.2021.9515353.

[22] N. K. Mehta, S. S. Prasad, S. Saurav, R. Saini, and S. Singh, "Three-dimensional DenseNet self-attention neural network for automatic detection of student's engagement," Applied Intelligence, vol. 52, no. 12, pp. 13803–13823, Sep. 2022, doi: 10.1007/s10489-022-03200-4.

[23] Y.-Y. Li and Y.-P. Hung, "Feature Fusion of Face and Body for Engagement Intensity Detection," in 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 3312–3316.

[24] O. Copur, M. Nakıp, S. Scardapane, and J. Slowack, "Engagement Detection with Multi-Task Training in E-Learning Environments," Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.04020.

[25] O. M. Nezami, M. Dras, L. Hamey, D. Richards, S. Wan, and C. Paris, “Automatic Recognition of Student Engagement using Deep Learning and Facial Expression,” CoRR, vol. abs/1808.02324, Aug. 2018, [Online]. Available: http://arxiv.org/abs/1808.02324.

[26] S. S. Mane and A. R. Surve, “Engagement Detection using Video-based Estimation of Head Movement,” in 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018), 2018, pp. 1745–1749.

[27] K. Altuwairqi, S. K. Jarraya, A. Allinjawi, and M. Hammami, “A new emotion–based affective model to detect student’s engagement,” Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 1, pp. 99–109, Jan. 2021, doi: 10.1016/j.jksuci.2018.12.008.

[28] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.05756.

A Hybrid Model for Covid-19 Detection using CT-Scans

Nagwa G. Ali[1], Fahad K. El Sheref[2], Mahmoud M. El khouly[3]
Department of Information Technology-Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt[1, 3]
Department of Information System-Faculty of Computers and Artificial Intelligence, Beni-Suef University, Egypt[2]

***Abstract*—Although some believe it has been wiped out, the coronavirus is striking again. Controlling this epidemic necessitates early detection of coronavirus disease. Computed tomography (CT) scan images allow fast and accurate screening for COVID-19. This study seeks to develop the most precise model for identifying and classifying COVID-19 by developing an automated approach using transfer-learning CNN models as a base. Transfer learning models like VGG16, Resnet50, and Xception are employed in this study. The VGG16 has a 98.39% accuracy, the Resnet50 has a 97.27% accuracy, and the Xception has a 96.6% accuracy; after that, a hybrid model made using the stacking ensemble method has an accuracy of 98.71%. According to the findings, hybrid architecture offers greater accuracy than a single architecture.**

Keywords—Covid-19; coronavirus; transfer-learning; CT-scan and ensemble method

I. Introduction

COVID-19 is caused by the virus SARS-CoV-2. Lung infections can cause anything from a simple cold to potentially fatal illnesses. Symptoms of the respiratory system are usually seen with coronavirus infections. In addition, people can occasionally get minor, self-limiting ailments like the flu that have detrimental effects. Fever, a cough, and trouble breathing might be caused by respiratory issues, exhaustion, and a sore throat [1]-[3].

Huge losses have been caused by the global spread of the COVID-19 pandemic. The urgent issue that health care and medical organizations are addressing is the quick detection of COVID-19 [1]. COVID-19 presents a serious threat to public health and the country's social and economic life because of the quick rise in the number of newly verified and reported cases.

An automated system that can measure the patient's infected zone and assess the progress of infected people by CT scan image analysis and clinical diagnosis is required. Even for skilled medical professionals, diagnosing COVID-19 is challenging [4], [5].

The two most common methods for diagnosing COVID-19 are computed tomography and X-rays. Medical health guidelines advocate chest imaging as a quick and efficient treatment, and it has been acknowledged as the first instrument in viral screening in a series of papers. Chest X-rays and CT scans have promising results despite frequently being examined by qualified radiologists. Computer-aided diagnosis (CAD) is required to reduce error rates while saving time and money [6] since radiologists see many patients daily and the diagnostic procedure takes a long time.

CT scans provide a series of slices of a specific location without including the numerous physiological features, in contrast to traditional X-rays. Compared to traditional X-rays, CT scans offer a considerably more thorough picture of the diagnosis [7].

Deep learning has been used in several studies to analyze radiological images. They were created to address the drawbacks of COVID-19 medical procedures that depend on radiological imaging. CNN is the method that is most effective for identifying the most significant deep learning algorithms. Consequently, the data processing field has shown a lot of interest in deep learning algorithms, especially at CNN [8].

This paper proposes a highly accurate automated computer-aided diagnostic approach for COVID-19 classification. In order to create a hybrid model using stacking ensemble with improved performance on the SARS-COV-2 dataset, a study was done to examine the implementation of three pre-trained convolutional neural network (CNN) models based on transfer learning. This work's main novelty and contributions are as follows:

- Reducing the number of false negative and positive values in the modelling process.
- The base and hybrid models were trained for 100 epochs.
- Comparing the recommended model to that of current DL frameworks in order to determine its effectiveness.
- Comparing the hybrid model's performance to the base-line models.
- Applying the stacking ensemble method in a new way instead of the familiar ways of stacking, which saves more time.

The following parts are organized as follows: The state-of-the-art studies are summarized in Section II, and resources and methods are covered in Section III. Results and analysis are presented in Section IV. The summary and recommendations for future work are presented in Section V.

II. Related Work

Many studies were conducted to diagnose COVID-19 either using CT scan images or using x-rays. This section summarizes the most recent studies, focusing on CT scan images.

Kogilavani, S. V., et al. [8] proposed using many CNN architectures, such as MobileNet, DeseNet121, VGG16, NASNet, Xception, and EfficientNet. The accuracy rates were 96.38%, 97.53%, 97.68%, 89.51%, 92.47%, and 80.19%, respectively.

Ebenezer Jangam et al. [9] developed a stacked ensemble method using VGG-19, ResNet-101, DenseNet-169, and WideResNet-50-2 as four pre-trained computer vision models. With the help of three separate chest CT scans, the suggested model was trained and assessed. The suggested technique has an accuracy of 0.8473, 0.99, and 0.935 on these datasets.

Rohit Kundu et al. [10] presented a bagging ensemble method of three transfer learning models, including Inception v3, ResNet 34, and DenseNet 201, which has been utilized to improve the performance of the individual models. The proposed model had a 97.81% accuracy rate.

A deep learning model, called truncated VGG16, was created by Mukul Singh et al. [11] to screen COVID-19 CT scans. For feature extraction from CT scan pictures, the VGG16 architecture has been fine-tuned; principal component analysis (PCA) is also used to choose features. For the final classification, four distinct classifiers are compared. On 208 tested images, the best classifier was a bagging ensemble with SVM, which obtained an F1 score of 95.3%, an accuracy of 95.7%, a precision of 95.8%, and an area under the curve (AUC) of 0.958.

For the diagnosis of COVID-19 from CT scans, Bejoy Abraham et al. [12] suggested using the kernel support vector machine with features extracted from five distinct CNNs (MobilenetV2, Darknet53, Shufflenet, Xception, and EfficientnetB0). The technique achieved 0.916 accuracy, a 0.91 F-score, and 0.917 sensitivity.

Nirmala Devi Kathamuthu, et al. [13] suggested applying a variety of foundation models, including VGG16, VGG19, Inception V3, Densenet 121, Xception, and Resnet50; the best model of these models attained an accuracy of 98% using ct-scan images. The difference between this model and ours is the data augmentation technique used, the training parameters, and the fact that we finally built a hybrid model using the stacking ensemble method.

Gifani et al. [14] employed fifteen pre-trained convolutional neural networks (CNNs) architectures: EfficientNets (B0-B5), NasNetLarge, NasNetMobile, ResNet-50, SeResnet 50, Xception, DenseNet121, ResNext50, and Inception ResNet v2. Thus, to further improve recognition performance, they created an ensemble approach based on majority voting on the ideal concoction of deep transfer learning outputs. The experimental results show that the majority voting of five deep transfer learning models with EfficientNetB0, EfficientNetB3, EfficientNet5, Inception ResNet v2, and Xception outperform individual transfer learning structures in terms of precision (0.857), recall (0.854), and accuracy (0.85) metrics for diagnosing COVID-19 from CT scans.

Horry, Michael J., et al. [15] developed a conceptual transfer learning framework to support COVID-19 detection with image categorization utilising deep learning models for X-ray, ultrasound, and CT scans, including Resnet50, Inception V3, Xception, VGG16/VGG19, InceptionResNet, DenseNet, and NASNetLarge. They settled on the VGG19 model, which was then adjusted with the proper parameters to reach extremely high levels of COVID-19 detection versus pneumonia and normal in all three types of lung images, with a precision of up to 86% for X-rays, 100% for ultrasounds, and 84% for CT scans.

The authors of [16] propose an augmented CNN to identify COVID-19 on CT scan and X-ray chest images and to differentiate COVID-19 patients from non-COVID-19 cases. While using these augmented images to train CNN, a classification accuracy of 98.97 percent for X-ray images and 95.38 percent for CT scan images was achieved.

Tanvir Mahmud et al. [17] presented the hybrid neural network CovTANet for the early prediction and diagnosis of COVID-19 using CT scans. A segmentation network was used to predict the lesions with a 95.8% accuracy rate.

A transfer learning strategy was employed by Chun Li et al. [18] to propose a way of training the model using a few CT images. The evaluation accuracy of this method was 87.6% for COVID-19 severity using pre-trained ChexNet.

In order to detect COVID-19 from a CT scan, Varan Singh Rohila et al. [19] offer a DCNN model known as ReCOV-101 that makes use of ResNet-101 as its basis. To increase the dataset, data augmentation, transfer learning, and the 94.9 percent accurate "skip connection" method are employed. From the previous review, further work needs to be done to increase the reliability and accuracy of COVID-19 detection using CT scan images.

Limitations of the previous studies include:

- The difficulty of sharing medical data is data privacy.
- The datasets that are currently accessible are incorrect, unclear, noisy, and incomplete.
- Most studies in the literature are conducted using datasets from different internet sources.

III. Method

The method used to achieve the objectives of the study is briefly described in this section. The diagram of the suggested method is shown in Fig. 1. Each model was trained individually on the training set; after that, the models were concatenated using stacking ensemble learning and fine-tuned using the training set. Fig. 3 and Fig. 4, which are subfigures of Fig. 1, explain the proposed model in more detail.

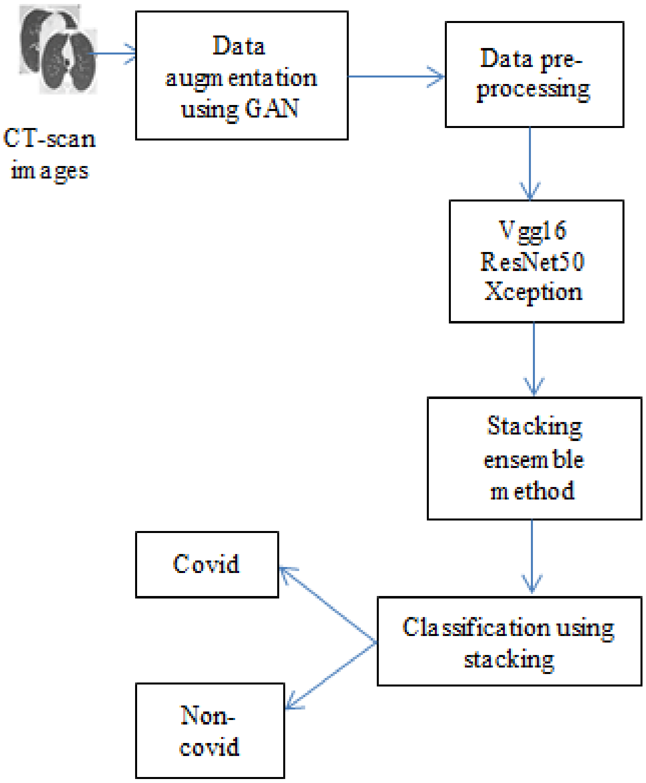


Fig. 1. Proposed model architecture.

A. Dataset

The SARS-CoV-2 CT scan dataset [20] was used in the proposed work to identify COVID-19 cases. This dataset is the most global because it contains more CT scan images that are clear and free of noise, and it is also used in more studies. The dataset consists of 2482 CT scan images, 1252 positive for COVID-19 (+ve) and 1230 negative for COVID-19 (-ve). These data were compiled from information provided by actual patients who went to hospitals in Sao Paulo, Brazil. This dataset aims to advance research and develop artificial intelligence (AI) tools that can detect SARS-CoV-2 or COVID-19 infection by looking at CT images.

B. Data Augmentation

A form of neural network design called a generative adversarial network (GAN) has much potential for artificial intelligence. The min-max two-player zero-sum game is an important factor in GAN. In this game one player profited by the same amount from the other's loss. The actors in this scenario are two separate GAN networks known as the discriminator and generator.

The primary goal of the discriminator, indicated as D, is to determine whether a sample is real or fake[21]. A fake sample of an image is produced by the generator, known as G, in contrast, in order to deceive the discriminator.

The discriminator determines the probability that a particular sample is real. The probability value will probably be greater for a real sample. A probability value that is very close to 0 indicates fake samples. When the discriminator is no longer able to distinguish between a real and fake sample and the probability value is close to 0.5, the generator can have an ideal answer [21]**.** Fig. 2 shows a sample of COVID and non-COVID images generated by GAN.

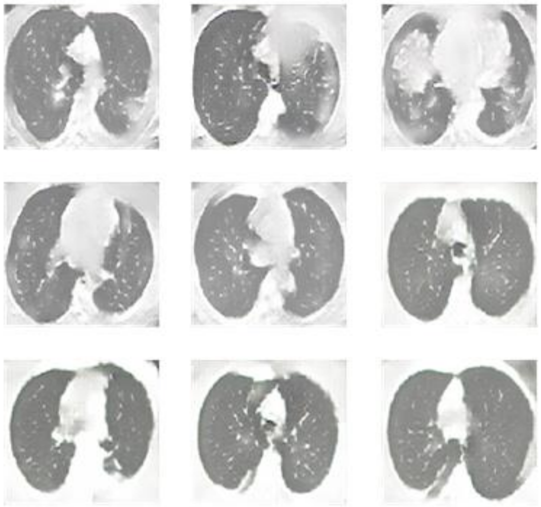


Fig. 2. Sample of COVID and non-COVID images after augmentation.

C. Data Pre-processing

Pre-processing has been done on the CT scan images. The image must fit the input size required to train the deep learning algorithm and make predictions using the data. To test the input size of the network, the data is rescaled. The image size in the proposed system has a dimension of 224 x 224. The data is consequently rescaled to match the supplied dimensions. These images were divided into two groups with an 80/20 split: a training set and a testing set. Table I shows the dataset after augmentation and preprocessing.

TABLE I. SARS COV-2 DATASET AFTER PREPROCESSING

Dataset	Covid	Non-Covid	Total
Training	1239	1239	2478
Testing	311	310	621
Total	1550	1549	3099

D. Classification Phase with Different Deep Learning Models

In order to classify the CT scan images, Xception[22], Resnet 50[23] ,and VGG 16 [22] are used as examples of deep learning architectures. These models are trained through transfer learning. The stacked model is then used to integrate the earlier models, which were each trained for 100 epochs, into one. Complete details are provided below.

E. Transfer Learning

Transfer learning using ImageNet was used to solve the problem of insufficient data. For each model, the weights from the ImageNet training were downloaded. The applied layer training procedure used the feature maps as input. To prevent losing any data during subsequent training rounds, freeze the layers of the model that have already been trained. A fully connected deep neural network was fed with the most recent feature map, which was flattened. The other layers were trained to extract more data from the later convolution layers because they were closer to the output features. As illustrated in Fig. 3, we added three additional layers to the top of each model: namely, dense with an output of 512, a dropout layer, and another dense layer with a sigmoid classifier. For each of the neural networks used in this study, a dropout layer of 0.25 was added to avoid overlapping [24]. The network was trained with a sigmoid classifier using an Adam optimizer for 100 iterations, with a batch size of 32 and a learning rate of 0.00001.

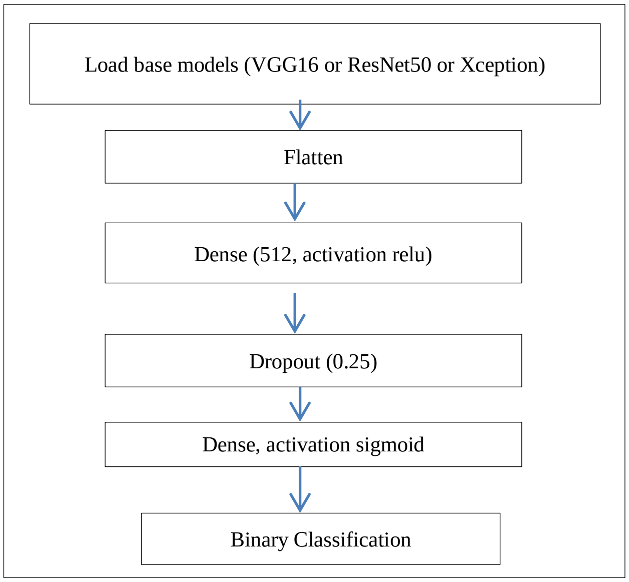

Load base models (VGG16 or ResNet50 or Xception)
Flatten
Dense (512, activation relu)
Dropout (0.25)
Dense, activation sigmoid
Binary Classification

Fig. 3. Proposed transfer learning base models.

F. Stacking Ensemble Method

Ensemble approaches come in a variety of forms, including average, weighted average, boosting, bagging, and stacking. In this paper, stacked generalization is used, which is an ensemble technique that learns how to integrate the predictions from various existing models in the most effective way [25].

This study employs a neural network as a meta-learner when neural networks are used as sub-models. In more detail, the sub-networks can be incorporated into a more extensive multi-headed neural network, which will then figure out how to integrate the predictions from each input sub-model most effectively. It makes it possible to think of the stacking ensemble as a single, enormous model.

All of the loaded models are utilized as a distinct input to the bigger stacking ensemble model, which can be defined just after models are loaded as a list. To prevent the weights from being modified while the new, larger model is being trained, each of the loaded models must have all of its layers marked as not trainable.

This method has the advantage of immediately giving the meta-learner the outputs of the sub-models. This new model will employ a different input head for each input layer from each sub-model. This indicates that any input data must be delivered to the model in multiples of k, where k is the total number of input models, which in this case are 3.

The results from all the models can then be combined. Here, a straightforward concatenation merge was employed. Then, a hidden layer is built to interpret this "input" for the meta-learner, and an output layer is defined to produce its own probabilistic prediction. The base model's training parameters were used to train the stack model as well. Fig. 4 shows the architecture of the stacked generalization.

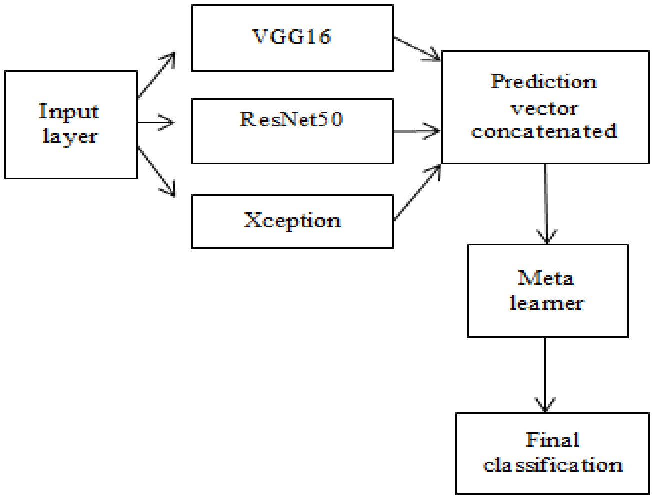


Fig. 4. Stacked generalization architecture.

IV. Results

This section discusses the results of both base and hybrid models using different evaluation measures.

A. Performance Evaluation Measures

A number of metrics, such as accuracy, precision, recall, and F1-score, can be used to evaluate the model's performance.

Accuracy: The ratio of correctly anticipated observations to all observations is the easiest and most obvious performance statistic. Given in the equation below:

$$\text{Accuracy} = TP+TN/TP+FP+FN+TN \tag{1}$$

where

- True positives (TP) are instances in which we made a prediction that someone had the disease and they actually did.
- True Negatives (TN): As expected, they are free of the illness.
- False positives (FP): Although we expected them to have the illness, they don't.
- False negatives (FN): Despite our prediction, they are infected.

The ratio of correctly predicted positive observations to all positively expected observations is known as precision. Given in the equation below:

$$\text{Precision} = TP/TP+FP \tag{2}$$

Recall is the percentage of accurately predicted positive observations among all observations in the current class. This is known as sensitivity. Given in the equation below:

$$\text{Recall} = TP/TP+FN \quad (3)$$

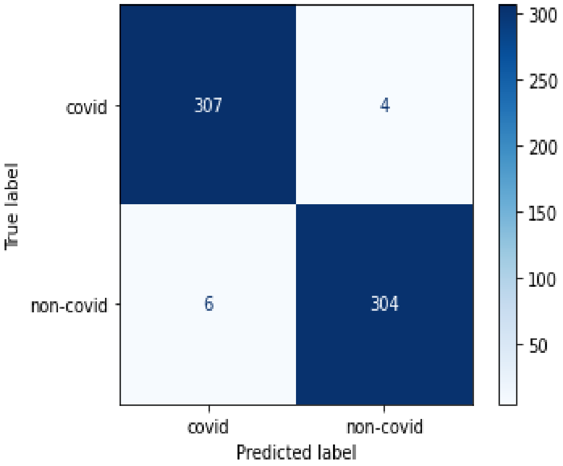
The mean of recall and precision is known as the F1 score. It is provided in the equation below:

$$\text{F1Score}=2*(\text{Recall}*\text{Precision})/(\text{Recall}+\text{Precision}) \quad (4)$$

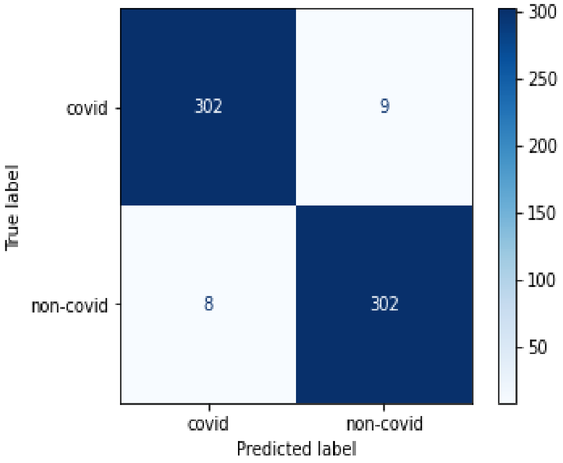
Table II displays the three used models and the hybrid model evaluation measure, while Fig. 5 displays the confusion matrix of the base and hybrid models.

TABLE II. EVALUATION MEASURES OF THE MODELS

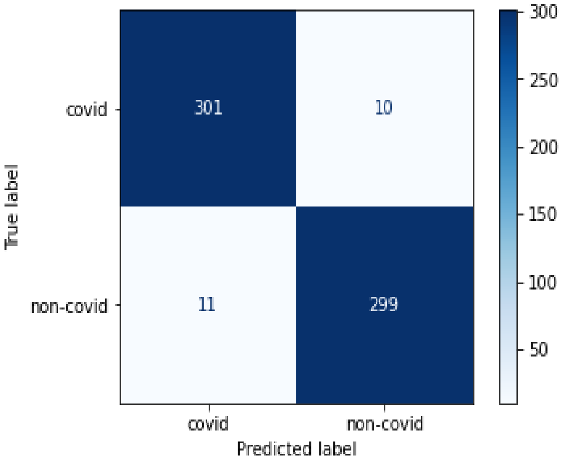
Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG16	98.39	98.7	98.06	98.39
Resnet50	97.26	97.1	97.4	97.26
Xception	96.62	96.76	96.45	96.6
Hybrid	98.71	98.39	99.0	98.7



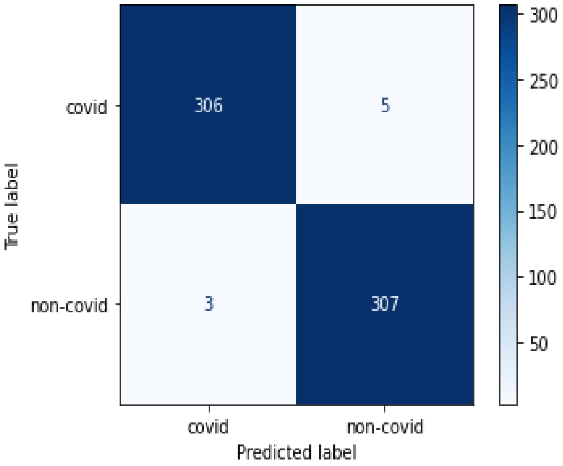
(a) VGG16 predicted 307 as COVID and 304 as non-COVID out of 621 images.



(b) Resnet50 predicted 302 as COVID and 302 as non-COVID out of 621 images.



(c) Xception predicted 301 as COVID and 299 as non-COVID out of 621 images.



(d) The stacked method predicted 306 as COVID and 307 as non-COVID out of 621 images.

Fig. 5. Base and hybrid model confusion matrix.

As shown in Table II and Fig. 5, the hybrid model achieved better results than the single model in terms of accuracy, recall, and f1-score.

V. DISCUSSION

Generally, the outcome is based on testing data, which is comprised of 20% of the entire images from the sars-cov2 dataset. Using the Keras library, the deep model employed in this study was developed. Keras is compatible with deep learning libraries such as TensorFlow and Theano. The model was developed using Google Colab, which enables you to create and run Python code in your browser and offers free access to the GPU.

A. Comparison

The suggested model is compared with other research done using the same dataset (Table III). This comparison shows that although many studies in the literature have combined more deep learning models, these studies are weak compared to ours. The best advantage of our proposed model is that we used GAN as an augmentation technique rather than traditional methods for increasing the number of images used to train the models, trained the model for 100 epochs with the suggested training parameters discussed in Fig. 3, and eliminated overfitting by applying the stacking ensemble method.

TABLE III. COMPARISON

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG16 [8]	97.68	_____	______	____
VGG19+DenseNet169+ResNet101[9]	94	90	98	94
Inceptionv3+Resnet34+DesNet201[10]	97.81	97.77	97.81	97.77
Five pre-trained CNN models +KSVM[21]	91.6	____	91.7	91
VGG16[22]	98	97.99	97.99	97.90
Proposed model	98.71	98.39	99.0	98.7

VI. CONCLUSION AND FUTURE WORK

In this proposed work, greater transfer learning models such as VGG16, Resnet50, and Xception are trained individually on the training data before being combined to enhance the model's performance and prevent over fitting. Kaggle's collection of CT scan images contains fewer images than necessary, so data augmentation using GAN is used to obtain more images. Performance metrics, including precision, F1-score, and recall, are used to assess the model's performance. The stacked model outperforms the single model in terms of accuracy, coming in at 98.71. By comparing the proposed method with other studies applied to the same dataset, it is obvious that the proposed method gives high recall, accuracy, precision, and the f1-score.

The future work of this study will train the suggested model for more epochs using different preprocessing techniques, apply the suggested model to another dataset, and attempt to use a different optimizer.

Data Availability: The data that support the findings of this study are openly available upon request from the authors.

Funding: The authors received no specific funding for this work.

REFERENCES

[1] M. Sandeep Kumar et al., "Social economic impact of COVID-19 outbreak in India," Int. J. Pervasive Comput. Commun., vol. 16, no. 4, pp. 309–319, 2020, doi: 10.1108/IJPCC-06-2020-0053.

[2] M. Venkatasen et al., "Forecasting of the SARS-CoV-2 epidemic in India using SIR model, flatten curve and herd immunity," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020, doi: 10.1007/s12652-020-02641-4.

[3] K. Banu Priya et al., "Pediatric and geriatric immunity network mobile computational model for COVID-19," Int. J. Pervasive Comput. Commun., vol. 16, no. 4, pp. 321–330, 2020, doi: 10.1108/IJPCC-06-2020-0054.

[4] S. K. Mathivanan et al., "Adoption of E-Learning during Lockdown in India," Int. J. Syst. Assur. Eng. Manag., p. 13198, 2021, doi: 10.1007/s13198-021-01072-4.

[5] S. Rajendran and P. Jayagopal, "Accessing Covid19 epidemic outbreak in Tamilnadu and the impact of lockdown through epidemiological models and dynamic systems," Meas. J. Int. Meas. Confed., vol. 169, no. August 2020, p. 108432, 2021, doi: 10.1016/j.measurement.2020.108432.

[6] S. Hasija, P. Akash, M. Bhargav Hemanth, A. Kumar, and S. Sharma, "A novel approach for detection of COVID-19 and Pneumonia using only binary classification from chest CT-scans," Neurosci. Informatics, vol. 2, no. 4, p. 100069, 2022, doi: 10.1016/j.neuri.2022.100069.

[7] P. Silva et al., "COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis," Informatics Med. Unlocked, vol. 20, p. 100427, 2020, doi: 10.1016/j.imu.2020.100427.

[8] S. V. Kogilavani et al., "COVID-19 Detection Based on Lung Ct Scan Using Deep Learning Techniques," Comput. Math. Methods Med., vol. 2022, 2022, doi: 10.1155/2022/7672196.

[9] E. Jangam and C. S. R. Annavarapu, "A stacked ensemble for the detection of COVID-19 with high recall and accuracy," Comput. Biol. Med., vol. 135, no. June, p. 104608, 2021, doi: 10.1016/j.compbiomed.2021.104608.

[10] R. Kundu, P. K. Singh, M. Ferrara, A. Ahmadian, and R. Sarkar, "ET-NET: an ensemble of transfer learning models for prediction of COVID-19 infection through chest CT-scan images," Multimed. Tools Appl., vol. 81, no. 1, pp. 31–50, 2022, doi: 10.1007/s11042-021-11319-8.

[11] M. Singh, S. Bansal, S. Ahuja, R. K. Dubey, B. K. Panigrahi, and N. Dey, "Transfer learning–based ensemble support vector machine model for automated COVID-19 detection using lung computerized tomography scan data," Med. Biol. Eng. Comput., vol. 59, no. 4, pp. 825–839, 2021, doi: 10.1007/s11517-020-02299-2.

[12] B. Abraham and M. S. Nair, "Computer-aided detection of COVID-19 from CT scans using an ensemble of CNNs and KSVM classifier," Signal, Image Video Process., vol. 16, no. 3, pp. 587–594, 2022, doi: 10.1007/s11760-021-01991-6.

[13] N. D. Kathamuthu et al., "A deep transfer learning-based convolution neural network model for COVID-19 detection using computed tomography scan images for medical applications," Adv. Eng. Softw., vol. 175, no. October 2022, p. 103317, 2023, doi: 10.1016/j.advengsoft.2022.103317.

[14] P. gifani, A. Shalbaf, and M. Vafaeezadeh, "Automated detection of COVID-19 using ensemble of transfer learning with deep convolutional neural network based on CT scans," Int. J. Comput. Assist. Radiol. Surg., vol. 16, no. 1, pp. 115–123, 2021, doi: 10.1007/s11548-020-02286-w.

[15] M. J. Horry et al., "COVID-19 Detection through Transfer Learning Using Multimodal Imaging Data," IEEE Access, vol. 8, pp. 149808–149824, 2020, doi: 10.1109/ACCESS.2020.3016780.

[16] D. Giri, K. R. Choo, and S. Ponnusamy, of the Seventh International Conference on Mathematics and Computing. 2021.

[17] T. Mahmud et al., "CovTANet: A Hybrid Tri-Level Attention-Based Network for Lesion Segmentation, Diagnosis, and Severity Prediction of COVID-19 Chest CT Scans," IEEE Trans. Ind. Informatics, vol. 17, no. 9, pp. 6489–6498, 2021, doi: 10.1109/TII.2020.3048391.

[18] C. Li, Y. Yang, H. Liang, and B. Wu, "Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets[Formula presented]," Knowledge-Based Syst., vol. 218, p. 106849, 2021, doi: 10.1016/j.knosys.2021.106849.

[19] V. S. Rohila, N. Gupta, A. Kaul, and D. K. Sharma, "Deep learning assisted COVID-19 detection using full CT-scans," Internet of Things (Netherlands), vol. 14, p. 100377, 2021, doi: 10.1016/j.iot.2021.100377.

[20] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification," medRxiv, p. 2020.04.24.20078584, 2020, [Online]. Available: https://www.medrxiv.org/content/10.1101/2020.04.24.20078584v2%0Ahttps://www.medrxiv.org/content/10.1101/2020.04.24.20078584v2.abstract

[21] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of Generative Adversarial Networks (GANs): An Updated Review," Arch. Comput. Methods Eng., vol. 28, no. 2, pp. 525–552, 2021, doi: 10.1007/s11831-019-09388-y.

[22] B. Nigam, A. Nigam, R. Jain, S. Dodia, N. Arora, and B. Annappa, "COVID-19: Automatic detection from X-ray images by utilizing deep learning methods," Expert Syst. Appl., vol. 176, no. January, p. 114883, 2021, doi: 10.1016/j.eswa.2021.114883.

[23] V. Sangeetha and K. J. R. Prasad, “Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2- carbaldehydes,” Indian J. Chem. - Sect. B Org. Med. Chem., vol. 45, no. 8, pp. 1951–1954, 2006, doi: 10.1002/chin.200650130.

[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” pp. 1–18, 2012, [Online]. Available: http://arxiv.org/abs/1207.0580.

[25] A. I. Naimi and L. B. Balzer, “Stacked generalization: an introduction to super learning,” Eur. J. Epidemiol., vol. 33, no. 5, pp. 459–464, 2018, doi: 10.1007/s10654-018-0390-z.

A New Big Data Architecture for Analysis: The Challenges on Social Media

Abdessamad Essaidi, Mostafa Bellafkih
RAISS Laboratory-Department of Mathematics and Computer Science,
National Institute of Posts and Telecommunications (INPT), Rabat, Morocco

***Abstract*—The streams of social media big data are now becoming an important issue. But the analytics method and tools for this data may not be able to find the useful information from this massive amount of data. The question then becomes: how do we create a high-performance platform and a method to efficiently analyse social networks' big data; how to develop a suitable mining algorithm for finding useful information from social media big data. In this work, we propose a new hierarchical big data analysis for understanding human interaction, and we present a new method to measure the useful tweets of Twitter users based on the three factors of tweet texts. Finally, we use this test implementation score, in order to detect useful and classification tweets by interested degree.**

Keywords—Social media; useful information; big data analysis; stream processing; classification tweets

I. Introduction

Big data on social networks and the progress of tools for calculating and analyzing these data have occupied the first place of investment in the last few years [1]. As a result, much of the literature devoted to Big Data in social media oscillates between two approaches. That is why, in recent years, there is so much interest in public debate on social media [2]. For this, an analysis of the information induced by the use of this type of data was strongly linked by social data. Also, and above all, the appearance of a new form of society "led by the analysis of Big Data." Following this proliferation of information on social media, it seemed useful to us to propose a new analysis method based on big data to draw up an initial assessment of the effects of big data on its practices [3], its objects and its results.

To this end, we analyzed the social network community around two main questions that we felt should not be separated: how does Big Data transform society? [4], How does this data affect personnes practice (Covid-19 use case)?, This double approach – which we wanted to keep in the preparation of this article – shows the fact that the proliferation of information in social networks is today confronted with a real change in the processes of transformation of societies which affects it so directly.

The main objective of this article is how big data in social networks affect the practice of society based on the new architecture of Big Data. The development and analysis of this massive data would thus provide those who use it with new knowledge. This new knowledge will affect the behavior of individuals, their interactions, their uses. We also provide a detailed comparison of the various tools and architectures used to process data stream [5]. Lastly, we offer our own architecture based on comparisons made. So how does big data in social media affect society practice?

To talk about this issue in depth, this paper starts with a review of previous social media research on human interaction data and social big data stream processing architectures on social media platforms, followed by the discussions of our proposed big data analysis architecture, and we suggest a new method with new measuring process based on important aspects, such as number of likes, comments and retweets on the Twitter platform that can be used to calculate the LCR score for tweet text. Finally, we use our data mining algorithm to predict the tweet texts (useful or not), and as the key to crucial insights on human behavior.

II. Research Background

A. *Existing Systems*

Most works focus on how big data affect the practice of society focused on the concept of analyzing content and patterns of interaction in social networks.

Felt [6], this qualitative research is a study on the analysis of social media data in addition to traditional, qualitative methods to analysis of big data.

Ghani et al. [7] based their study on an overview of recent works and provided a general perspective on the subject of social media big data analytics research. This study also provides a comparison of possible big data analysis techniques and their quality attributes.

Bello-Orgaz et al. [8] have worked this study on the revising new methodologies designed to enable efficient data mining and merging information from social media and new apps and frameworks that are now emerging under the social media "umbrella", social media as well as big data paradigms.

Immonen et al. [9] based their work on the introducing a new framework to assessment and manage quality of social networks data at each treatment phase of the Big Data. The proposed solution delivers real-time validated data to social network users, which results in better decision-making. This data is extracted from social media and used to determine customer satisfaction with the quality of a product.

B. *Social Media*

The social media platform has become the primary contact with our family, friends and colleagues. People share all the information in a lot of different forms, like messages, video as well as audio on the social networks to shares their feeling distinguished or bad. The world average for Internet users is 2.5 hours a day on platforms social media. Social networking platform users produce a significant amount of information and data that cannot be processed through traditional data analytics. Therefore, social networking platforms are making substantial investments in this information and big data because it provides insights into the analysis of media, which shapes public opinion, etc. [10].

Social media messages are rich with possibilities for data mining and analysis. This shift on social media provides new challenges for researchers interested in analyzing public publications and messages of social networking user platforms to better understand human interaction and improve the human condition [6].

Social media platforms like Twitter were viewed as a critical source of useful information.

During Covid-19, people post updates regarding their statuses on social media platforms like Twitter, request assistance and other relevant information, report damage to infrastructure, injured people, etc.

Information posted on social media during crises, their worth varies considerably. Most publications do not contain any useful information nor are they useful for disaster response. Most of the posted messages contain a deluge of noise that is of a personal nature, do not contain any useful information, or are irrelevant [11].

The proposed model for obtaining information on social media, which is shown in Fig. 1, aims to collect clean data consisting of posts containing potentially useful information. This information can be used for various purposes, such as helping injured people affected by Covid-19.

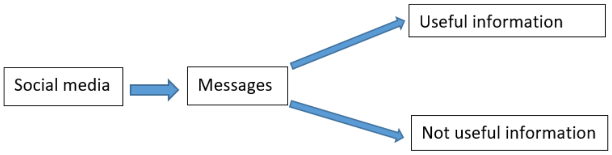


Fig. 1. Information on social media.

C. Big Data

Big data is a term that aims to describe massive volumes of data and offer an alternative to conventional solutions of analyse databases in order to make sense of it and make the most of it.

The Data mining on social media and efficient utilisation of this Big data became a significant issue for numerous research fields such as massive data processing, big data analytics techniques, machine learning, semantics of the data, information fusion, computational intelligence, and social networks.

The big data concept has been specified by the three essential dimensions 3V: volume, variety, and velocity that were set out in 2001 by Laney [12] as: “high-volume, high-variety and high-velocity information, new approaches to information processing and use this information for various purposes such decision making”.

This set of Big Data 3V models, which is shown in Fig. 2, provides a direct and broad definition related to what constitutes a big data-based problem, software, application, or framework, as well as what does not. In this context, it may be described in brief as follows:

- Volume: Designates big amounts of different data and from various sources, including mobile digital data development devices as well as digital devices. The benefit of collecting, processing and analyzing these large quantities of data creates a number of difficulties in obtaining for knowledge human interaction and enterprises [8-13].
- Variety: The variety of data refers to different types of structural heterogeneities within a dataset. These various types of information collected via social networks, smartphones or sensors, such as text, data logs, pictures, videos, audio, and the like. Additionally, these data is structured (such as relational databases data), semi-structured, or of an unstructured form [8-14].
- Velocity: Refers to the rate at which are generated and speed of data transfers. Proliferation rapid and growth of digital devices technologies as smartphones and sensors have resulted in record levels of data creation, and the various forms of streamed data from variety sources. As a result, velocity of big data proliferation at the time of the development process should be taken into consideration [14, 15].

Big Data represents the central set of technologies and parts for processing user-generated data used on social media platforms. Without regard on the type of data (structured, informal or semi-structured), they can be containing useful information.

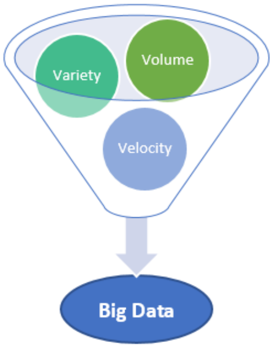


Fig. 2. The 3V model of big data.

III. Social Big Data Stream Processing Architectures

In this section, we provide a brief summary of Big Data characteristics generated by social media platforms, and we will look at the two major flow processing architectures, Kappa and Lambda—that are adopted for executing analytics.

When looking at popular online social media containing stacks of big data, makes the underlying processing of this data becomes challenging, and requires the implementation of an application-specific.

Big data technology influences current data management, and makes comparative analysis of these data essential to academic and industry communities.

Given the huge interest in social media platforms big data by universities, industry, a variety of solutions and techniques have been published over the past few years. With their progressive maturity, there is an increasing need to evaluate and compare such solutions. As a result, the science community was particularly interested in big data techniques. In fact, comparative analysis greatly facilitates the comparison of performances and provides useful information [16].

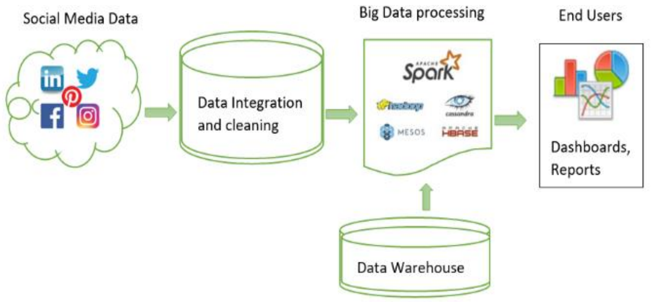


Fig. 3. Social media big data processing.

The social media businesses want to adapt real-time advertising placement algorithms analytics for insight generation, and Fig. 3 is shown the model of Social Media Big Data processing.

The analyzing big data continuously in real time is not an easy task.

There are many architectural propositions for analyzing big data in real-time streaming, but the most interesting one is Lambda and Kappa architecture.

A. Lambda Architecture

Lambda Architecture, suggested in 2011 by Marz and his team [17], offer the option to calculate arbitrary functions on an arbitrary dataset in real time by breaking down the problem into three layers (batch layer, speed layer and serving layer) [18], and provides a series of architectural principles that can be integrated and treatment of flow and batch data, in a low latency and a single Big Data architecture.

The Lambda Architecture consists of three main layers that interact with incoming data: the batch layer, serving layer, and speed layer. Fig. 4 illustrates the core technologies, components, processes, and responsibilities that constitute each layer of the Lambda Architecture.

The batch layer contains ever-increasing master dataset stored and pre-computes batch views on a distributed filesystem (such as HDFS) and produce batch views. MapReduce is used for processing the batch data, which is the Hadoop programming model. The serving layer indexes the batch views and does not require random writes, but must random reads. To implement the serving layer, usually technologies such as HBase, Storm and Cassandra are used. The last layer is the speed layer which only handles new data and uses an incremental model whereby the real-time views are incremented. At this layer stream processing system is generally based on Strom technology.

B. Kappa Architecture

The Kappa architecture, which was first described by Kreps, is perfectly suitable for processing data flows [19]. In this architecture, the key thought is to use one simple layer in real time for processing data flow and batch processing [16].

In the kappa architecture, everything is a stream [20], and you need is a stream processing engine data. So, what we would traditionally call batch processing of data is simply streaming through bounded datasets. However, Kappa is only focused and offers the option of creating a streaming and data batch treatment system based on the same technology. Similarly, queries search for a single location of the Kappa architecture instead of two of the Lambda architecture.

As shown in Fig. 5, the Kappa architecture is composed of the streaming layer module, responsible for orderly data events and one Serving layer, manages query results.

Summarizing, the kappa architecture focuses on the processing of data streams more than storage and this architecture is better adapted for cases where there is no need to permanently store the data.

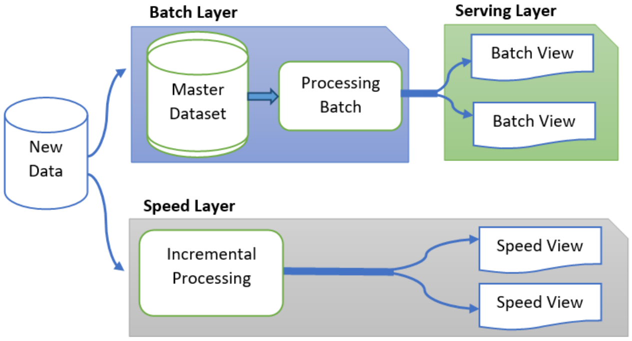


Fig. 4. Lambda architecture.

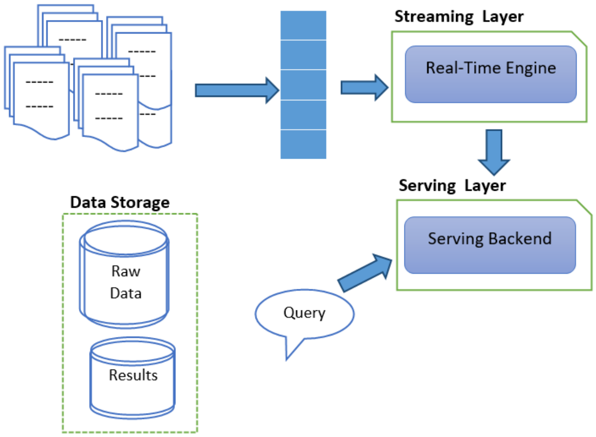


Fig. 5. Kappa architecture.

In the present paper, we have reviewed studies on the processing of big data mining on social media and focused on the concept of analyzing, we present our analysis processing, and we suggest a new method with new measuring process based on important aspects, such as number of likes, comments and retweets on the Twitter platform that can be used to predict the tweet texts (useful or not). Finally, we use our data mining algorithm of the tweet texts is used to calculate the LCR score for tweet text.

IV. Our Proposed Big Data Analysis Architecture for Understanding Human Interaction

Definition of the environment in which Big Data is processed continuously in real time, regardless of the type of data is no easy feat. There are much architectural proposals for real-time big data analysis, but the most interesting thing about our problem is developing architecture for extracting high-quality information about relevant messages [21]. In this section, we are presenting motivation to offer the architecture that best suits our use case and how it works.

The wide-spread popularity of social media has grown considerably the ratio of various streamed data, no matter what type of information or data (structured, semi-structured or non-structured). This information comes from regular people and this data may be in any of the next two statuses: data Useful (in motion) and not Useful (at rest) [22]. Therefore, various data quality distributions of generated by people using social networks are naturally fuzzy and non-structured [23], range from high-grade to low-grade. All this information can include the personal opinion of social media users [24], behaviors and thoughts, which makes it more and more important to extract useful information. Due to the availability of user-generated content can encapsulate helpful and high quality. Such increased in data generation has brought attention to the needs of analyzing big data in real time [25].

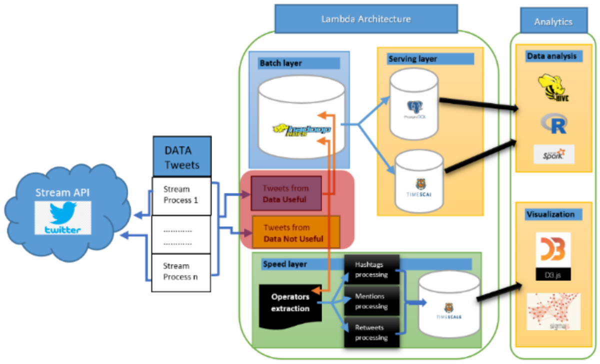


Fig. 6. Stream analytics system architecture for analyzing twitter text data.

As shown in Fig. 6, the practice example demonstrates the analysis process in Hadoop for analysis Twitter text data.

A researcher can retrieve data stored on Twitter through the public API provided by the social networking service Twitter to process specific information. However, in many cases, they do not provide data required by researchers. For example, may be needed for some additional data or perform cleaning and filtering operations to obtain useful data, such as the number of Retweets of user content or their popularity or reputation. Therefore, collecting useful data is a set of skills and methods and discipline of users in order to capture all the big data without any restrictions.

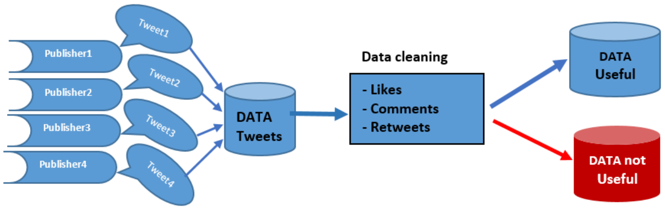


Fig. 7. Our proposed analysis procedures model.

As shown in Fig. 7, the Our Proposed Analysis Procedures to select the most relevant and interesting Tweets by cleaning and filtering using factors based on overall number of likes, overall number of comments and overall number of tweets. However, we need a more advanced method that can detect useful tweets.

LCR score, which is shown in equation (1), is our method used to measure the performance of the model.

$$f(L,C,R) = C1 \times Likes\_count + C2 * Comments\_count + C3 * Retweets\_\text{count} \quad (1)$$

Where f(L, C, R) is LCR score, C1, C2 and C3 is coefficients parameters and Likes_count, Comments_count and Retweets_count is the number of (Likes, Comments, Retweets).

The measuring process is shown in equation (1) that combines likes, comments, and retweets (LCR Score) for a tweet. And we rank those tweets based on the LCR score for each one. It should be noted that news tweets and it contains important information generally have a high number of retweets and that this type of score will be useful for our analysis of the information. We therefore assigned a high C3 coefficient for retweets to this type of content and a relatively low C2 coefficient for comments and a very low C1 coefficient for likes when calculating the LCR score.

In Table I, the three factors of tweet texts are shown, and we note that the Tweet texts not Useful and is not taken into account since the LCR Score is equal to zero.

TABLE I. Three Factors of Tweet Texts

Retweets_count	Comment_count	Likes_count	LCR Score	Tweet texts
R > 0	C >= 0	L >=0	LCR> 0	Useful tweet
R= 0	C= 0	L= 0	LCR= 0	Not Useful tweet

Data mining of the tweet texts algorithm is used to calculate the LCR score for tweet text, based on the overall number of likes L, the overall number of comments C and the overall number retweets R for a tweet texts in D (D represents the raw data). The algorithm LCR begins with an initial score equal to zero. Thus, that the Tweet text Not Useful and is not taken into account. then, the LCR score is computed step by step for a tweet text, and sequential patterns that combines likes count, comments count, and retweets count— is going to use these operators to find the raw data tweet useful.

Algorithm 1: Data mining of the tweet texts

```
Input:
D (The raw of dataset)
L (The likes count)
C (The comments count)
R (The retweets count)
LCR (Score of tweet text)
Output:
LCR (Score of tweet text)
Initialize Tweet_text = Scan(D)
    If Tweet_text contains content
      L= Likes_count
      C = Comments_count
      R = Retweets_count
      LCR = Score(L,C,R)
    End
```

V. Results and Discussion

Big data in social networks occupy a big place in our lives, and during the COVID-19 pandemic, social media have become essential to be in contact with our loved ones.

The tweets data analyzed in Table II are pre-defined data (not in real-time). We use this tweets text to test implementation of the LCR score, in order to detect useful and to classification tweets by interested degree.

TABLE II. Results Values Calculated by LCR Score For Pre-stored Tweets Texts

Tweet text	Retweets count	Comments count	Likes count	LCR Score
The EU says it may have a digital identity portfolio by 24, regardless of the challenges	106	11	89	429
WHO: COVID-19 - we all need to remain vigilant.	142	333	270	1362
Swedish company DSruptive Subdermal that specializes in microchip implants has created a way to store COVID-19 passport data. Here’s how it works.	340	70	265	1425
Andrew Tate nails the covid narrative in 90 Seconds!	1013	53	2292	5437
UNICEF vaccine ad	1245	547	1345	6174

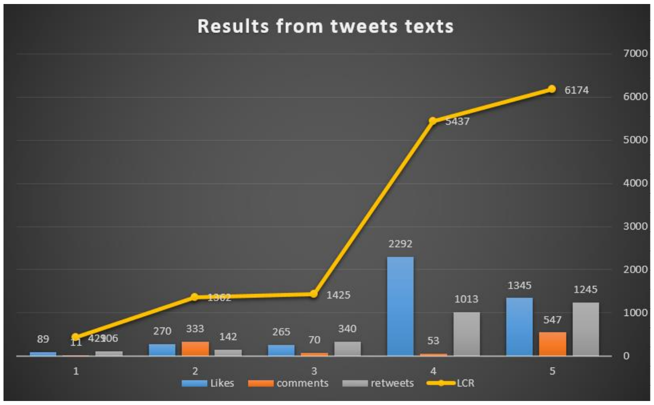


Fig. 8. Procedures a part of result from tweets texts.

The results in Fig. 8 show that an important Tweets stimulate the users to retweet, where the important information generally have a high number of retweets. The high precision of the LCR Score manifests in the form of a large number of retweets, which brings useful information. Furthermore, the results indicate that the number of likes and the user's ability to comment on the tweet are relatively low factors in detecting useful tweets. The experiments carried out allow us to know the most useful tweet from which the important information is disseminated and retweeted in a major time. To conclude, the more Retweets we have, the stronger to detect useful tweets are and the more significant the calculated LCR score is.

The work presented in this document represents our efforts towards creating a truly representative and comprehensive Big Data benchmark suite, using a high volume of data collected on Twitter by viral tweets related to COVID-19. We suggest a new approach with new measures that can be used for prediction the tweet texts (useful or not), then we make several interesting observations throughout diffusion graph and an analysis in detail using factors based on number of likes, number of comments and number of tweets on the Twitter platform.

We have compared of the two most known architectures for analyzing streamed big data in Section III. We're talking about the criteria here of choice of our architecture for analyzing Twitter text data and how well it meets our problem's requirements. The proposes of this work is about building a new Big Data architecture and advanced method capable to analysis the tweets texts and detecting useful tweets. By two solutions main challenges for analysis of social media data. First, how to detect not useful and irrelevant messages from big data analysis and second, categorization of this informative tweet into different degree of interest. By utilizing data from past detectoing, we show the performance of LCR score algorithm for multi-class classification factors. We observe the tweet texts to contains useful information always help improve the classification accuracy.

To evaluate the performance, the proposed method and obtained results show that the method achieves good results to detect useful and to classification tweets by interested degree.

VI. Conclusion

In this article, we reviewed studies on the procesures of big data mining on social media and focused on the concept of analysing. By the main stream processing and our proposed big data analytics architecture for understanding human interaction, the LCR measuring process method provides a framework for such research and is summarized in three sections: tweet texts input, tweet texts analysis (LCR score), and tweet output (tweet containing useful information or not useful). From a big data analytical framework perspective and our method, the discussions are focused on the analysis-oriented, and results-oriented of tweet texts information. It has to do with the perspective of data mining, this document gives a technologies of data mining algorithm of the tweet texts information and classification.

In the future phase of research, we're going to be implementing additional elements of our Architecture for analyzing Twitter texts data in order to analyze the batch data and real-time data. We shall test the speed layer with social networking platforms, by combining technologies such as Hadoop, Storm and Kafka.

REFERENCES

[1] Fu, Weina, Shuai Liu, and Gautam Srivastava. "Optimization of big data scheduling in social networks." Entropy 21.9 (2019): 902.

[2] Kay, Samantha, Rory Mulcahy, and Joy Parkinson. "When less is more: the impact of macro and micro social media influencers' disclosure." Journal of Marketing Management 36.3-4 (2020): 248-278.

[3] Sivarajah, Uthayasankar, et al. "Critical analysis of Big Data challenges and analytical methods." Journal of business research 70 (2017): 263-286.

[4] Loebbecke, Claudia, and Arnold Picot. "Reflections on societal and business model transformation arising from digitization and big data analytics: A research agenda." The Journal of Strategic Information Systems 24.3 (2015): 149-157.

[5] Nirmal, V. Jude, and DI George Amalarethinam. "Parallel implementation of big data pre-processing algorithms for sentiment analysis of social networking data." International journal of fuzzy mathematical archive 6.2 (2015): 149-159.

[6] FELT, Mylynn. Social media and the social sciences: How researchers employ Big Data analytics. Big Data & Society, 2016, vol. 3, no 1, p. 2053951716645828.

[7] GHANI, Norjihan Abdul, HAMID, Suraya, HASHEM, Ibrahim Abaker Targio, et al. Social media big data analytics: A survey. Computers in Human Behavior, 2019, vol. 101, p. 417-428.

[8] Bello-Orgaz, Gema, Jason J. Jung, and David Camacho. "Social big data: Recent achievements and new challenges." Information Fusion 28 (2016): 45-59.

[9] IMMONEN, Anne, PÄÄKKÖNEN, Pekka, et OVASKA, Eila. Evaluating the quality of social media data in big data architecture. Ieee Access, 2015, vol. 3, p. 2028-2043.

[10] KUMARI, Savita. Impact of big data and social media on society. Global Journal for Research Analysis, 2016, vol. 5, p. 437-438.

[11] Nguyen, Dat Tien, et al. "Applications of online deep learning for crisis response using social media information." arXiv preprint arXiv:1610.01030 (2016).

[12] Laney, Doug. "3D data management: Controlling data volume, velocity and variety." META group research note 6.70 (2001): 1.

[13] Hashem, Ibrahim Abaker Targio, et al. "The rise of "big data" on cloud computing: Review and open research issues." Information systems 47 (2015): 98-115.

[14] Ghani, Norjihan Abdul, et al. "Social media big data analytics: A survey." Computers in Human Behavior 101 (2019): 417-428.

[15] Gandomi, Amir, and Murtaza Haider. "Beyond the hype: Big data concepts, methods, and analytics." International journal of information management 35.2 (2015): 137-144.

[16] Persico, V., Pescapé, A., Picariello, A., & Sperlí, G. (2018). Benchmarking big data architectures for social networks data processing using public cloud platforms. Future Generation Computer Systems, 89, 98-109.

[17] N. Marz and J. Warren, Big Data: Principles and best practices of scalable real-time data systems. New York; Manning Publications Co., 2015.

[18] Hasani, Z., Kon-Popovska, M., & Velinov, G. (2014). Lambda architecture for real time big data analytic. ICT Innovations, 133-143.

[19] J. Kreps, "Questioning the lambda architecture," Online Artic. July, p. 205, 2014.

[20] Lin, Jimmy. "The lambda and the kappa." IEEE Internet Computing 21.05 (2017): 60-66.

[21] El Marrakchi, M., Bensaid, H., & Bellafkih, M. (2017). E-reputation prediction model in online social networks. International Journal of Intelligent Systems and Applications, 9(11), 17.

[22] Arnaboldi, Valerio, et al. "Online social networks and information diffusion: The role of ego networks." Online Social Networks and Media 1 (2017): 44-55

[23] S. Samanta, V.K. Dubey and B. Sarkar, Measure of influences in social networks. Appl. Soft Comput. 99 (2021) 106858.

[24] Essaidi, Abdessamad, Dounia Zaidouni, and Mostafa Bellafkih. "New method to measure the influence of Twitter users." 2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS). IEEE, 2020.

[25] Jesmeen, M. Z. H., et al. "A survey on cleaning dirty data using machine learning paradigm for big data analytics." Indonesian Journal of Electrical Engineering and Computer Science 10.3 (2018): 1234-1243.

Drug Repositioning for Coronavirus (COVID-19) using Different Deep Learning Architectures

Afaf A. Morad[1], Mohamed A. Aborizka[2], Fahima A. Maghraby[3]
Arab Academy for Science Technology & Maritime Transport-College of Computing and Information Technology,
Heliopolis, Cairo Governorate, Egypt[1, 2, 3]

***Abstract*—In December 2019, the COVID-19 epidemic was found in Wuhan, China, and soon hundreds of millions were infected. Therefore, several efforts were made to identify commercially available drugs to repurpose them against COVID-19. Inferring potential drug indications through computational drug repositioning is an efficient method. The drug repositioning problem is a top-K recommendation function that presents the most likely drugs for specific diseases based on drug and disease-related data. The accurate prediction of drug-target interactions (DTI) is very important for drug repositioning. Deep learning (DL) models were recently exploited for promising DTI prediction performance. To build deep learning models for DTI prediction, encoder-decoder architectures can be utilized. In this paper, a deep learning-based drug repositioning approach is proposed, which is composed of two experimental phases. Firstly, training and evaluating different deep learning encoder-decoder architecture models using the benchmark DAVIS Dataset. The trained deep learning models have been evaluated using two evaluation metrics; mean square error and the concordance index. Secondly, predicting antiviral drugs for Covid-19 using the trained deep learning models created during the first phase. In this phase, these models have been experimented to predict different antiviral drug lists, which then have been compared with a recently published antiviral drug list for Covid-19 using the concordance index metric. The overall experimental results of both phases showed that the most accurate three deep learning compound-encoder/protein-encoder architectures are Morgan/AAC, CNN/AAC, and CNN/CNN with best values for the mean square error, the first phase concordance index, and the second phase concordance index.**

Keywords—Antiviral drugs; computational drug repositioning; coronavirus; deep learning; drug-target interactions

I. Introduction

Since December 2019, Coronavirus disease (COVID-19) has become a crucial public issue across the world. There is a real need to develop antiviral drugs for COVID-19 to stop viral infections. Recent efforts have been carried out to design novel inhibitors or utilize a drug repurposing strategy to determine anti-COVID-19 drugs that can serve as promising inhibitors versus coronavirus protease [1, 2].

Drug discovery and development is a time-consuming, complicated and costly task, including the identification of candidates, synthesis, characterization, screening, assays for therapeutic efficacy, and clinical trials [3, 4, 5]. However, drug development success rates are extremely low. In clinical phrases, numerous investigational drugs have failed due to insufficient achievement, safety concerns, or commercial purposes [6]. Alternative drug development strategy drug repositioning seeks to identify novel uses for present drugs and can decrease the risk and costs associated with the development of new drugs [7, 8]. Inferring potential drug indications via computational drug repositioning is an efficient method. The drug repositioning problem is a top-K recommendation function that presents the most probable drugs for certain diseases based on drug and disease-related data. For drug repositioning, it is crucial to accurately predict drug-target interactions (DTI), which define the binding of substances to protein targets [9]. The precise identification of molecular drug targets is essential for drug discovery and development [10, 11] and is particularly important for discovering effective and safe treatments for new pathogens, such as SARS-CoV-2 [12].

Diverse issues in bioinformatics and cheminformatics applications [13], and more specifically, drug development and discovery [14], have been successfully solved using deep learning (DL) techniques [15, 16]. By comparing DL techniques to traditional machine learning (ML) algorithms, DL algorithms designed to predict drug-target binding affinities (DTBA) occasionally do better [17]. These DL-based DTBA prediction algorithms differ in two key aspects. The representation of the input data is the first aspect. The input drug features can include, for instance, extended connectivity fingerprint (ECFP), ligand maximum common substructure (LMCS), simplified molecular input line entry system (SMILES), or a combination of these features. The second aspect relates to the DL system design created using various neural network (NN) types [18]. The construction of the many NN types varies and may include a number of layers, hidden units, filter sizes, or an integrated activation function. Each variety of NN has particular advantages that make them better suited for particular applications. These variant types of NN are Feedforward Neural Networks; FNN, Radial Basis Function Neural Networks; RBNN, Multilayer Perceptron; MLP, Recurrent Neural Networks; RNN, Convolutional Neural Networks; CNN, and Modular Neural Networks; MNN, etc. [19].

Deep learning (DL) models for predicting drug-target interactions (DTI) often use encoder-decoder architectures [20]. The encoder converts ligand or protein representations into numerical vectors for training or evaluating the DL model. There are many different encoder-decoder architectures available for DTI prediction, but only a few have been explored in previous research. In this study, we propose a DL-based method for drug repositioning, which involves two experimental phases. Firstly, training and evaluating different deep network architectures of compound encoders (Morgan, CNN, CNN_RNN, MPNN, Transformer) and protein encoders (Transformer, AAC, CNN, CNN_RNN, Conjoint_triad) using the benchmark DAVIS Dataset [21]. The trained models have been evaluated using the mean square error and the concordance index metrics. Based on the experimental evaluation of this phase, the most superior five DL compound-encoder/protein-encoder architectures are Morgan/Transformer, Morgan/AAC, CNN/AAC, Morgan/CNN, and CNN/CNN. Secondly, predicting antiviral drugs for Covid-19 using the trained models. In this phase, the trained models created in the first phase have been experimented with to predict different antiviral drug lists for Covid-19. These lists have been evaluated by comparing them to a recently published antiviral drug list for Covid-19 [22] using the concordance index metric. The experimental evaluation of this phase showed that the most superior five DL compound-encoder/protein-encoder architectures are Morgan/Conjoint_triad, Morgan/AAC, CNN/AAC, CNN/CNN, and CNN/CNN_RNN. The overall experimental evaluation of both phases showed that the most accurate three DL compound-encoder/protein-encoder architectures are Morgan/AAC, CNN/AAC, and CNN/CNN with best values for the mean square error, the first phase concordance index, and the second phase concordance index. To summarize, there are two main contributions to this paper:

- Training and evaluating different twenty-one deep network architectures of compound encoders and protein encoders for drug repositioning using the benchmark DAVIS Dataset.
- Predicting antiviral drug lists for Covid-19 using the trained twenty-one models, and then comparing them to a recently published antiviral drug list for Covid-19.

The rest of this paper is organized in the following manner. While Section II provides an overview of the key scientific concepts, Section III reviews the related work. The conceptual model, system architecture, used benchmark DAVIS dataset, and reference Coronavirus antiviral medicine list are all described in Section IV along with the suggested technique. Section V describes the tools and libraries exploited in implementation and then presents the experimental evaluation using two main experiments: Evaluating the trained models using the DAVIS dataset and predicting antiviral drugs for Covid-19 using the trained models. Finally, the paper is concluded in Section VI.

II. Background

DTIs have a very important function in the drug discovery procedure. DTIs recognize the interaction sites between protein targets and drug compounds and describe the attributes of the interactions sites. DTI aims in recognizing new ligands versus specified protein targets. A large number of researches have gotten advantages from recognizing DTIs containing drug repositioning [23, 24]. Costly and time-consuming laboratory tests are required to determine the affinity value for a sizable number of drug-target combinations. Therefore, computational approaches have gotten more attention in the recent years [25].

A crucial step in predicting drug-target interactions DTI is feature extraction (also called feature encoding) from the input data. Feature extraction obtains useful, discriminating, and non-trivial information from input data to facilitate subsequent learning phases. Fig. 1 illustrates the two types of feature encoding techniques: data-driven and non-data-driven. The main distinction between these two groupings is that data-driven approaches develop characteristics for each input automatically. In strategies that are not data-driven, features are calculated in a fixed way for each input. The data-driven approaches are mainly based on deep learning methods, which is a set of machine learning algorithms that uses a model of the human visual system to create new hierarchical feature representations [26].

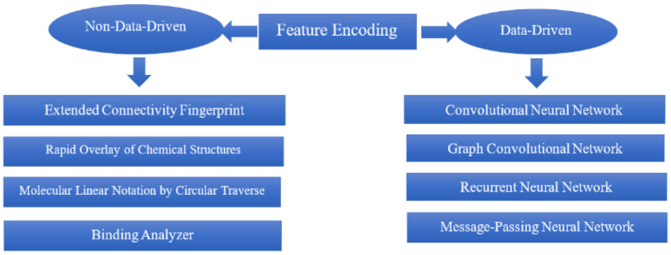

Non-Data-Driven
Feature Encoding
Data-Driven
Extended Connectivity Fingerprint
Rapid Overlay of Chemical Structures
Molecular Linear Notation by Circular Traverse
Binding Analyzer
Convolutional Neural Network
Graph Convolutional Network
Recurrent Neural Network
Message-Passing Neural Network

Fig. 1. Feature encoding methods [26].

A neural network with two or more hidden layers is deemed a deep learning. The input layer receives the input features directly, while the output layer generates predictions through a series of non-linear transformations utilizing hidden layers. Each output node corresponds to a class-based prediction task. If there is only one node in the output layer, then the network is considered a single-task deep learning. Otherwise, it is known as a multi-task deep learning [27].

A basic sequence, a derived atomic fingerprint, or a mixture of both can constitute the input feature space for a deep network. Numerous studies have used a network's input to be a raw molecular sequence. Other works convert a raw sequence to a more appropriate form, such as one-hot coding, to feed it to deep networks like CNN. In one-hot coding, each character in the sequence is represented by a binary vector with its matching bit set to one and the other bits set to zero.

There are many feature encoding architectures in DTI prediction such as Convolutional Neural Network CNN and Message-Passing Neural Network MPNN. For example, the Transformer encoders [28] are multi-layered bidirectional Transformer encoders following the initial Transformer model. The Transformer encoder is capable of modelling a sequence without the aid of a CNN or RNN. Transformer, in contrast to these earlier sequence processing layers, can efficiently encode the relationship between far-flung tokens (atoms) in a sequence. Various Transformer-based NLP models surpass earlier techniques in many benchmarks due to this effective context modelling.

Molecular descriptors must be created from symbolic representations of molecules, such as the SMILES (Simplified Molecular Input Line Entry System) format, in order to do deep learning. “Morgan Fingerprints,” a vector representation of molecular attributes, is a commonly adopted way to describe a molecule. Morgan fingerprint, also known as extended-connectivity fingerprint (ECFP) [29], is often used algorithm, or encoder specification. ECFPs are innovative category of topological fingerprints for molecular characterization. For the purpose of modelling structure-activity, ECFPs were created. ECFPs are circular fingerprints with a variety of useful properties: their ability to be calculated quickly; they are not predefined and can represent an essentially infinite number of different molecular features; and their features indicate the presence of specific substructures, which makes it simpler to interpret analysis results.

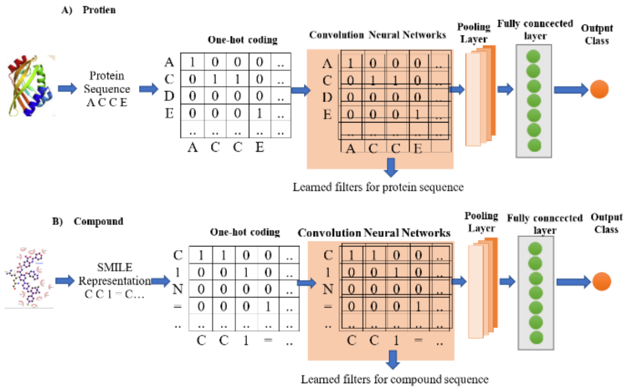


Fig. 2. One-dimension convolutional layer operation on a protein sequence and a molecule sequence [26].

The most popular deep neural network encoder is CNN, which operates on a grid data structure like digital images. As depicted in Fig. 2 [26], CNN consists of multiple convolutional and pooling layers arranged in an arbitrary order. Convolutional layers discover a series of filters that derive a group of local patterns from a specific receptive field of the layer input. In subsequent convolutional layers, the receptive field also expands. By down-sampling the layer's input, the pooling layer expands the receptive field. Moreover, the pooling layer does not define any additional parameters. The CNN input might consist of one-dimensional or two-dimensional matrices that are scanned along the sequence in only one or two directions, respectively. Until now, 1D CNNs have been used for DNA sequences for categorization, DNA-protein binding, and motif extraction, among other tasks [30, 31, 32].

In CNN, Fig. 2 depicts how to apply the one-dimension convolutional layer on a small molecule sequence or a protein sequence. In Fig. 2A, a protein sequence is depicted as an amino acids' sequence contained in a matrix where every amino acid is encoded by a single-hot code (all bits are zeros except the corresponding bit of the symbol is one). As seen, by reducing the loss function on both positive and negative task samples, the filter is moved along the sequence. Fig. 2B depicts the SMILE sequence as a string of characters. Every character represents an atom or molecule structural indicator. Every character is then encoded with a one-hot code and inserted in each matrix column. In both instances, the learned filter is displayed.

III. Related Work

Deep learning methods are now widely used in many research fields like; speech recognition [33, 34, 35] and image processing [36, 37, 38], involving bioinformatics such as genomics works [39,40] and quantitative-structure activity relationship (QSAR) researches in drug discovery [41]. The primary benefit of deep learning architectures aims to offer improved raw data representations through non-linear modifications in every layer [42]; hence facilitates the learning of the data's hidden patterns.

Several studies utilizing Deep Neural Networks (DNN) for the prediction of DTI binary class with various input models for drugs and proteins have been conducted previously [43,44,45], as well as a few studies utilizing stacked auto-encoders [46] and deep belief networks [47]. Likewise, stacked auto-encoder-based models using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were used to describe genomic and chemical structures as real-valued vectors [48, 49]. The protein-ligand interaction scoring is performed by using deep learning methods. The protein-ligand interaction scoring commonly utilizes CNNs, which learn from the three-dimensional compositions of the protein-ligand complexes [50, 51, 52, 53].

Pahikkala et al. [54] used a method called KronRLS (Kronecker Regularized Least Squares), which only requires two dimension-based chemical similarity-based representations of the medications and the Smith-Waterman similarity representation of the targets. Latterly, the SimBoost approach was suggested to forecast the scores of binding affinity with a gradient boosting machine by utilizing feature engineering to provide DTI [55]. They exploited similarity-based information of DT pairings and attributes acquired from the pairs' network-based interactions. Both kinds of research obtained similarity-based information using 2D representations of the substances and typical machine learning algorithms.

The ensemble of deep learning models (EnsembleDLM) for DTI prediction was presented [56]. EnsembleDLM utilizes a set of chemical compounds and proteins and assembles the predictions from numerous deep neural networks. It provides good achievements in cross-domain applications spanning various bio-activity types and protein classes. By using transfer learning, the EnsembleDLM obtained a good performance (Pearson correlation coefficient and concordance index).

To construct negative DTIs, a new similarity-based strategy was presented [57]. Multiple least absolute shrinkage and selection operator (LASSO) models were presented to incorporate various collections of feature sets in order to investigate the strength of prediction and forecast DTIs. In addition, LASSO Deep Neural Network (LASSO-DNN) model was developed to predict DTIs based on the features retrieved from the LASSO models with the highest achievement. LASSO-DNN was compared to LASSO, standard logistic (SLG) regression, support vector machine (SVM), and conventional DNN models. The LASSO-DNN outperformed the SLG, LASSO, SVM, and regular DNN models, as demonstrated by the results of the experiments.

Wang et al. [58] provided the three-step strategy for identifying obscure DTIs using deep learning. The first step is an illustration of drug-target pairings, where the drug compounds are encoded as fingerprint characteristics. In contrast, the protein sequence features are produced by applying Legendre Moments (LM) to a position-specific scoring matrix (PSSM), including evolutionary protein information. The second step involves compression and fusion of features. The sparse principal component analysis (SPCA) was utilized to reduce the dimension and redundancy of the features.

Eventually, the prediction task that used the deep long short-term memory (DeepLSTM) model was exploited. The experimental results proved that the suggested technique outperforms other DTI prediction methods.

DTI prediction model with 2D paired distance maps of proteins and molecular graphs as inputs for targets and medicines was presented [59]. To retrieve the interactive effects of targets and drugs, the mutual interaction neural network (MINN) by integrating two interacting transformers (Interformer, for short) with an enhanced communicative message passing neural network (CMPNN) (titled Inter-CMPNN) was proposed.

In conclusion, the majority of computational algorithms suggested forecasting DT interactions concentrated on binary classification, wheresoever the primary objective is to assess whether or not a drug-target pair interacts. Yet, the interactions of protein-ligand suppose a continuum of binding affinity values, often known as binding strength. Increased availability of affinity information in drug-target KBs (knowledge bases) enables the application of sophisticated learning approaches like deep learning architectures to predict binding affinities. The main contribution of this research is to train and evaluate twenty-one different deep network architectures of compound encoders and protein encoders for drug repositioning.

IV. Research Methodology

A. Proposed Approach

Fig. 3 shows the proposed approach conceptual model, which exploits a benchmark dataset called DAVIS [21] and a reference Coronavirus antiviral drug list [22], and then outputs a ranked list of candidate drugs for Coronavirus for each trained deep learning model. According to various compound-decoders/protein-decoders, twenty-one DL models have been trained.

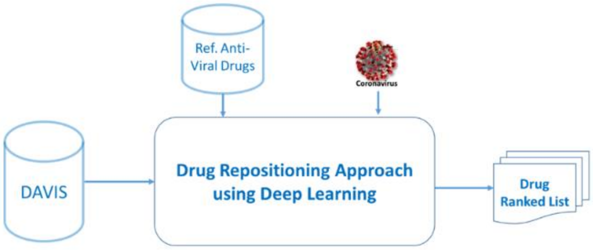


Fig. 3. The proposed approach conceptual model.

This research work has two main objectives. Firstly, training and evaluating models of different deep learning compound-encoders and protein-encoders using the benchmark DAVIS Dataset. Secondly, predicting antiviral drugs for Covid-19 using the trained models. To achieve these objectives, Fig. 4 shows the system architecture, which includes two phases: the training phase and the prediction phase. In the training phase, both drug compounds and disease target proteins of the DAVIS dataset are encoded using different neural network architectures and then their embeddings are decoded to generate the trained model. For example, as shown in Fig. 4, the drug compound is encoded using the Transformer architecture, and the disease target protein is encoded using the CNN architecture. The procedure of converting ligand or protein representations into numerical vectors utilized to train or assess a model of machine learning is known as encoding. Table I and Table II show the compound-encoders and protein-encoders exploited in this research, respectively. The cross-validation technique was applied on all DAVIS dataset instances.

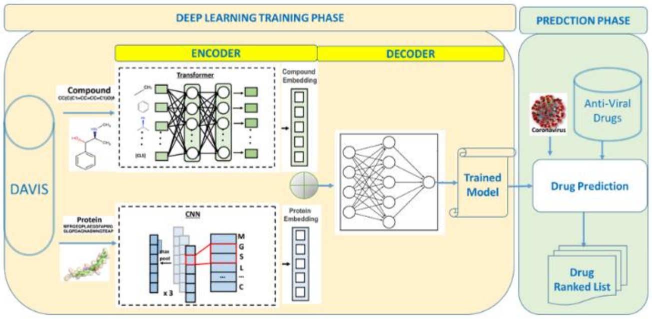


Fig. 4. The system architecture.

TABLE I. The Compound-Encoders and their Descriptions

Compound Encoder	Description
Morgan	Extended-Connectivity Fingerprints
CNN	Convolutional Neural Network on SMILES
CNN_RNN	A GRU/LSTM on top of a CNN on SMILES
MPNN	Message-passing neural network
Transformer	Transformer Encoder on ESPF

TABLE II. The Protein-Encoders and their Descriptions

Protein Encoder	Description
Morgan	Transformer Encoder on ESPF
CNN	Amino acid composition up to 3-mers
CNN_RNN	Convolutional Neural Network on target seq
MPNN	A GRU/LSTM on top of a CNN on target seq
Transformer	Conjoint triad features

In the prediction phase, the proposed solution exploits the trained model to predict a ranked list of drugs candidate for Coronavirus. The predicted ranked list of drugs is evaluated versus a reference list of antiviral drugs published recently for

Coronavirus. Fig. 5 shows how different trained models are generated and how different deep learning architectures are exploited. For example, the trained model 1 was generated by encoding drug compounds and disease target proteins using DNN and CNN, respectively. In the same way, the trained model 2 was generated by encoding drug compounds and disease target proteins using DNN and Transformer, respectively. In this research, twenty-one combinations of neural network architectures were experimented.

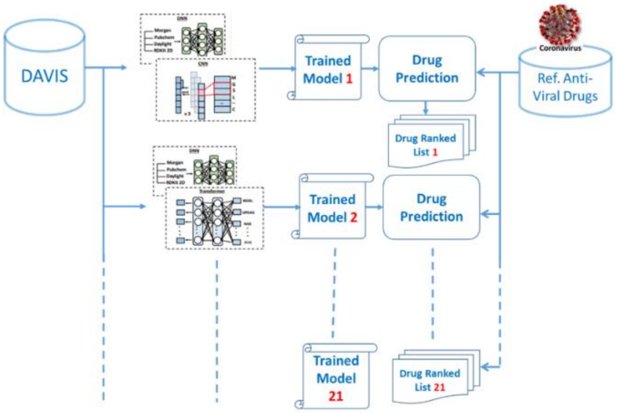


Fig. 5. Predicting twenty-one different drug-ranked lists for coronavirus.

B. The Benchmark Dataset

The selectivity assays of the set of kinase proteins and the pertinent inhibitors with their definite dissociation constant (Kd) values are included in the DAVIS dataset [21]. The DAVIS dataset compound SMILES strings were retrieved from the Pubchem compound database using their Pubchem CIDs [60]. The DAVIS dataset's protein sequences were retrieved from the UniProt protein repository using gene names and RefSeq accession codes [61]. Table III shows the number of Drug-Target Interaction Instances.

TABLE III. DAVIS DATASET SUMMARY

Proteins	Compounds	Interactions
424	68	30056

C. The Reference Coronavirus Antiviral Drug List

Recently, [22] used virtual screening and molecular docking methods to locate prospective inhibitors from existing drugs that can respond to COVID-19. Based on their binding energy (kcal/mol), the authors ranked 121 drugs as potent drugs against SARS-CoV-2 since they tightly bind to their main protease. Table IV presents a list of ranked Drugs based on their docking scores Binding energy (BE).

V. RESULTS AND DISCUSSION

A. Implementation Setup

In this research, the DeepPurpose DTI prediction tool was exploited. This tool provides a set of eight ligand encoders that can be combined with other protein representations and architectures to generate new models [62]. The DeepPurpose tool contains seven distinct protein encoders, which can be divided into two distinct categories: expert-designed algorithms and text-processing based on neural networks. Among algorithmic encoders, the ACC encoder produces a vector of 8420 elements describing the frequency of all amino acid k-mers for k values up to three [63]. In contrast, the conjoint triad encoder offers a 3-mer frequency count utilizing a restricted amino acid alphabet [64]. Neural network encoders, in contrast, apply directly on the sequence.

TABLE IV. A LIST OF RANKED DRUGS BASED ON THEIR DOCKING SCORES BINDING ENERGY (BE) [22]

Drug	BE	Drug	BE
Beclabuvir	-10.4	Pibrentasvir	-6.9
Nilotinib	-9.9	Valaciclovir	-6.9
Tirilazad	-9.6	Valganciclovir	-6.9
Paritaprevir	-9.2	Pirodavir	-6.8
Raltegravir	-9.1	Vidarabine	-6.8
Venetoclax	-9	Zanamivir	-6.8
Bictegravir	-8.9	Daclatasvir	-6.7
Danoprevir	-8.9	Alovudine	-6.6
Pimodivir	-8.8	Entecavir	-6.6
Voxilaprevir	-8.7	Famciclovir	-6.6
Faldaprevir	-8.6	Idoxuridine	-6.6
Setrobuvir	-8.6	Laninamivir	-6.6
Letermovir	-8.5	Tenofovir	-6.6
Bisantrene	-8.4	Cidofovir	-6.5
Rilpivirine	-8.4	Dasabuvir	-6.5
Indinavir	-8.2	Didanosine	-6.5
Ombitasvir	-8.2	Efavirenz	-6.5
Saquinavir	-8.2	Fiacitabine	-6.5
Simeprevir	-8.2	Lobucavir	-6.5
Digitoxin	-8.1	Penciclovir	-6.5
Dolutegravir	-8.1	Telbivudine	-6.5
Remdesivir	-8.1	Tenofovir disoproxil	-6.5
Lopinavir	-8	Umifenovir	-6.5
Maraviroc	-8	Capravirine	-6.4
Calanolide A	-7.9	Grazoprevir	-6.4
Darunavir	-7.9	Interferon alfa-2b	-6.4
Elbasvir	-7.9	Peramivir	-6.4
Etravirine	-7.9	Sorivudine	-6.4
Velpatasvir	-7.9	Zidovudine	-6.4
Vesatolimod	-7.9	Boceprevir	-6.3
Digoxin	-7.8	Emivirine	-6.3
Tipranavir	-7.8	Gancicolovir	-6.3
Amprenavir	-7.7	Ribavirin	-6.3
Maribavir	-7.7	Stavudine	-6.3

Nelfinavir	-7.7	Cenicriviroc	-6.2
Ruzasvir	-7.7	Dexelvucitabine	-6.2
Sofosbuvir	-7.7	MK-0608	-6.2
Doravirine	-7.6	Oseltamivir	-6.2
Mericitabine	-7.6	R1479	-6.2
Ritonavir	-7.6	Taribavirin	-6.1
Atazanavir	-7.5	Chloroquine	-6
Dapivirine	-7.5	Hydroxychloroquine	-6
Delavirdine	-7.5	Ledipasvir	-6
Fosamprenavir	-7.5	Elvucitabine	-5.9
Asunaprevir	-7.4	Emtricitabine	-5.9
Baloxavir marboxil	-7.4	Tromantadine	-5.9
Elvitegravir	-7.3	Acyclovir	-5.8
Sovaldi	-7.3	PSI-6130	-5.8
Tenofovir alafenamide	-7.3	Triazavirin	-5.8
Vedroprevir	-7.3	Aciclovir	-5.7
Galidesivir	-7.2	Fingolimod	-5.7
Methylprednisolone	-7.2	Lamivudine	-5.7
Clevudine	-7.1	Zalcitabine	-5.7
Telaprevir	-7.1	Ingavirin	-5.6
Thalidomide	-7.1	Rimantadine	-4.6
Abacavir	-7	Favipiravir	-4.4
Adefovir dipivoxil	-7	Amantadine	-4.3
Cobicistat	-7	Docosanol	-4.3
Nevirapine	-7	Foscarnet	-4.3
Tenofovie alafenamide	-7		

Every amino acid is transformed into a numerical value as a fixed-length one-dimension array using the DeepPurpose CNN encoder. Subsequently, it employs a convolutional neural network [65] to learn spatial information from the sequence (local amino acid neighborhoods) that may be pertinent to the DTI binding model.

Fig. 6 shows different compound-encoders and different protein-encoders developed by the DeepPurpose DTI prediction tool [60].

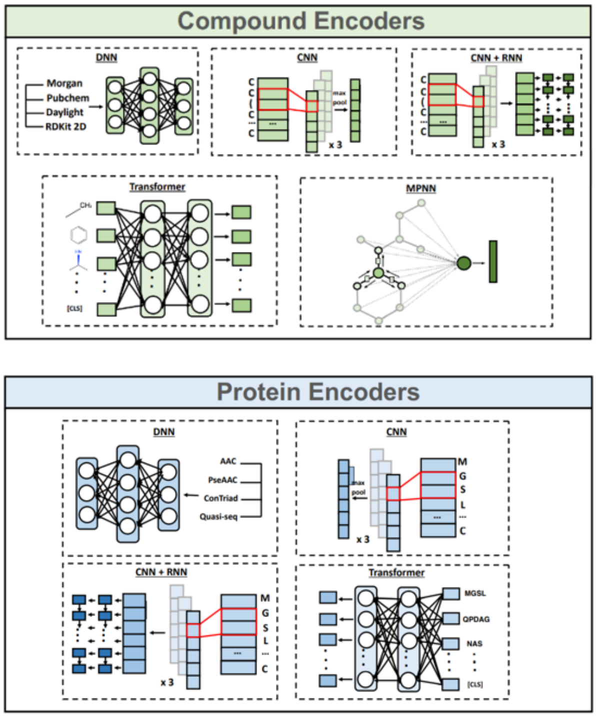


Fig. 6. Compound-encoders and protein-encoders developed by the DeepPurpose tool [62].

The DeepPurpose DTI prediction tool enables developers to set some pamperers for the model training phase. Table V shows some parameter settings for NN architectures, which had been used during the model training phase.

TABLE V. PARAMETER SETTINGS FOR NN-BASED TRAINED MODEL

Parameter	Range
Filter length (compounds)	[32,64,96]
Filter length (proteins)	[32,64,96]
Hidden neurons	1024; 1024; 512
Batch size	256
Epoch	100
Learning rate	0.001
CNN Drug Kernels	[4,8,12]
CNN Target Kernels	[4,8,12]

B. Evaluation Metrics

1) Mean squared error (MSE): During training, a learning model attempts to reduce the gap between the actual (real) value and the prediction. The mean squared error (MSE) is picked as the loss function because a regression problem is used. Equation 1 shows how the MSE is calculated.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(P_i - Y_i)^2 \quad (1)$$

Where P is the vector of predictions, Y is the vector of actual outputs, and n represents the total number of samples.

2) Concordance index (CI): The Concordance Index (CI) was used to evaluate the effectiveness of a model that outputs continuous values [17]. CI determines whether the anticipated binding affinity values of two random drug–target pairs were predicted in the same sequence as their true values were. Equation 2 shows how the CI is calculated.

$$CI = \frac{1}{Z}\sum_{\delta_i > \delta_j} h(b_i - b_j)$$

$$h(x) = \begin{cases} 1, & if\ x > 0 \\ 0.5, & if\ x = 0 \\ 0, & if\ x < 0 \end{cases} \quad (2)$$

Where b_i represents the prediction value for the greater affinity δ_i, b_j represents the prediction value for the lesser affinity δ_j, Z is a normalization constant, and h(x) is the step function.

C. Phase 1: Evaluating the Trained Models using the Davis Dataset

Based on exploiting different deep learning compound-encoders/protein-encoders, twenty-one trained models were evaluated using the benchmark DAVIS Dataset. The cross-validation technique was applied. Both Fig. 7 and Fig. 8 show the evaluation results for the different twenty-one trained models in terms of MSE (Mean Square Error) and CI (Concordance Index) evaluation metrics. It is noted that the most superior five DL compound-encoder/protein-encoder architectures are Morgan/Transformer, Morgan/AAC, CNN/AAC, Morgan/CNN, and CNN/CNN.

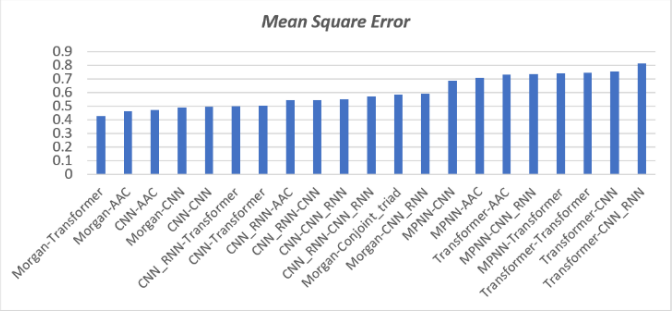


Fig. 7. The MSE evaluation results for the different twenty-one trained models.

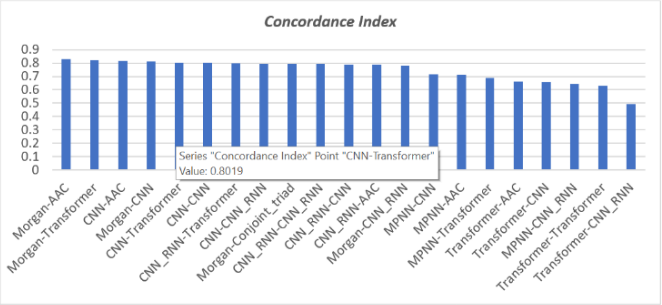
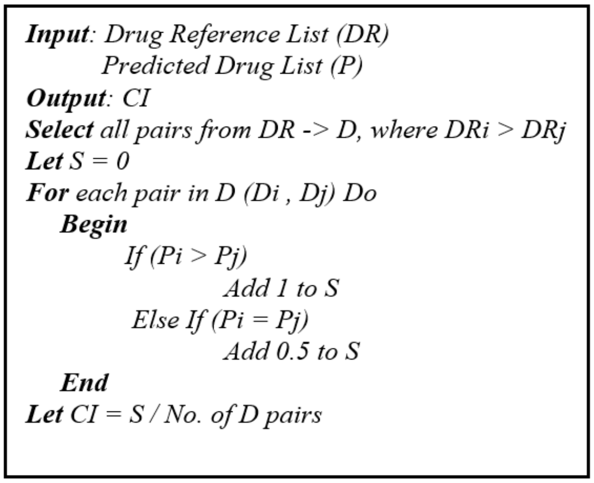


Fig. 8. The CI evaluation results for the different 21 trained models.

D. Phase 2: Predicting Antiviral Drugs for Covid-19 using the Trained Models

To evaluate the predicting accuracy for the trained models versus the 121-reference antiviral ranked drugs, the Concordance Index was used to compare each predicted ranked list versus the reference antiviral ranked drugs. Fig. 9 shows the Concordance Index calculating algorithm for a Predicted Drug List versus the Covid-19 Drug Reference List. Fig. 10 shows the Concordance Index for the twenty-one Predicted Drug Lists versus the Covid-19 Drug Reference List. As shown in Fig. 10, the most superior five DL compound-encoder/protein-encoder architectures are Morgan/Conjoint_triad, Morgan/AAC, CNN/AAC, CNN/CNN, and CNN/CNN_RNN.

The overall experimental evaluation for the two experimental phases can be summarized in Fig. 11, which shows the evaluation results using the CI metric for the different twenty-one trained models in the model testing phase using the DAVIS dataset and in the predicting phase versus the coronavirus drug reference list. As shown in Fig. 12, it is worth noting that the most accurate three deep learning compound-encoder/protein-encoder architectures are Morgan/AAC, CNN/AAC, and CNN/CNN with best values for the mean square error, the first phase concordance index, and the second phase concordance index.



```
Input: Drug Reference List (DR)
       Predicted Drug List (P)
Output: CI
Select all pairs from DR -> D, where DRi > DRj
Let S = 0
For each pair in D (Di , Dj) Do
   Begin
         If (Pi > Pj)
                   Add 1 to S
          Else If (Pi = Pj)
                   Add 0.5 to S
   End
Let CI = S / No. of D pairs
```

Fig. 9. The concordance index calculating algorithm for a predicted drug list versus the Covid-19 drug reference list.

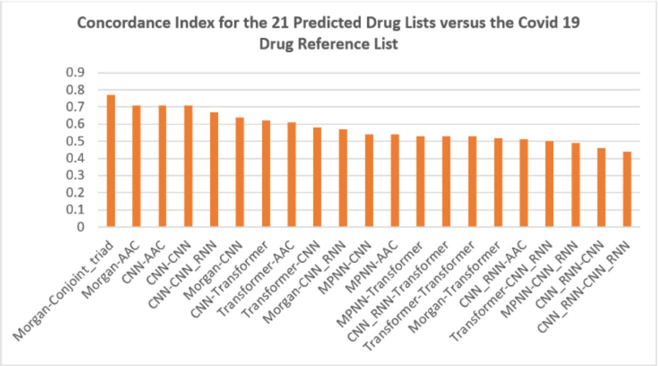


Fig. 10. The concordance index for the twenty-one predicted drug lists versus the Covid-19 drug reference list.

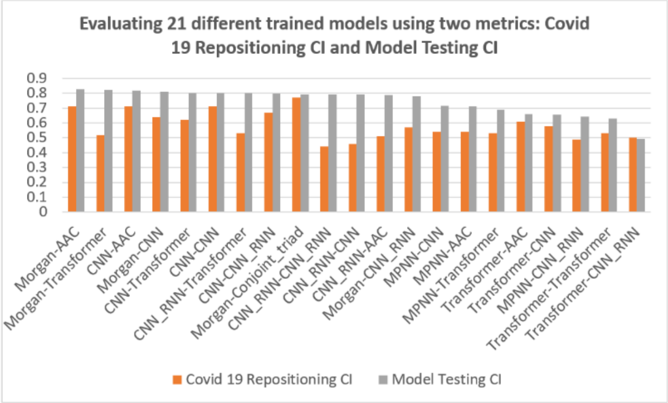


Fig. 11. Evaluating twenty-one different trained models using two metrics: Covid-19 repositioning CI and model testing CI.

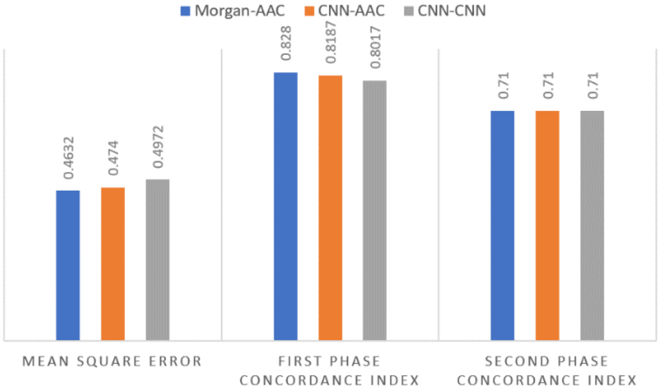


Fig. 12. The most accurate three deep learning compound-encoder / protein-encoder architectures: Morgan / AAC, CNN / AAC, and CNN / CNN.

VI. Conclusion

This research proposed a deep learning-based drug repositioning method to train and evaluate twenty-one models based on deep learning compound-encoders and protein-encoders. The trained models have been evaluated using two experiments. Firstly, testing the trained models by applying the cross-validation technique on the benchmark DAVIS Dataset. Secondly, comparing the predicted antiviral drug lists by the trained models to a recently published antiviral drug list for Covid-19. The experimental evaluation showed the most accurate three deep learning compound-encoder/protein-encoder architectures are Morgan/AAC, CNN/AAC, and CNN/CNN with best values for the mean square error, the first phase concordance index, and the second phase concordance index. As a future work, the same 21 different deep network architectures of protein and compound encoders are suggested to be trained and to be evaluated using other datasets and other viral diseases.

References

[1] R. J. Khan, R. Jha, G. M. Amera, M. Jain, E. Singh, A. Pathak, and A. Singh, “Targeting novel coronavirus 2019: A systematic drug repurposing approach to identify promising inhibitors against 3C-like proteinase and 2’-O-ribose methyltransferase”, Journal of Biomolecular Structure and Dynamics, vol. 2, pp. 1–40, 2020.

[2] P. Sarma, N. Sekhar, M. Prajapat, P. Avti, H. Kaur, S. Kumar, and B. Medhi, “In-silico homology assisted identification of inhibitor of RNA binding against 2019-nCoV N-protein (N terminal domain)”, Journal of Biomolecular Structure & Dynamics, vol. 39, no. 8, 2021.

[3] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C.C. Persinger, B.H. Munos, S.R. Lindborg, and A.L. Schacht, "How to improve R&D productivity: the pharmaceutical industry's grand challenge", Nature reviews Drug discovery, vol. 9, no. 3, pp. 203-214, 2010.

[4] N. Prakash and P. Devangi, "Drug Discovery", Journal of Antivirals & Antiretrovirals, vol. 2, no. 4, pp. 63-68, 2010.

[5] H. Xue, J. Li, H. Xie, and Y. Wang, "Review of drug repositioning approaches and resources", International journal of biological sciences, vol. 14, no. 10, pp. 1232-1244, 2018.

[6] T. J. Hwang, D. Carpenter, J. C. Lauffenburger, B. Wang, , J. M. Franklin, and A. S. Kesselheim, "Failure of investigational drugs in late-stage clinical development and publication of trial results", JAMA internal medicine, vol. 176, no. 12, pp. 1826- 1833, 2016.

[7] S. Pushpakom, F. Iorio, P.A. Eyers, K.J. Escott, S. Hopper, A. Wells, A. Doig, T. Guilliams, J. Latimer, C. McNamee, and A. Norris, “Drug repurposing: progress, challenges and recommendations”, Nature reviews Drug discovery, vol. 18, no. 1, pp.41-58, 2019.

[8] G. Jin and S. T. C. Wong, "Toward better drug repositioning: prioritizing and integrating existing methods into efficient pipelines", Drug discovery today, vol. 19, no. 5, pp. 637-644, 2014.

[9] R. Santos, O. Ursu, A. Gaulton et al., “A comprehensive map of molecular drug targets”, Nature reviews Drug discovery, vol. 16, no. 1, pp. 19–34, 2017.

[10] A. Rutkowska, D. W. Thomson, J. Vappiani et al., “A Modular Probe Strategy for Drug Localization, Target Identification and Target Occupancy Measurement on Single Cell Level”, ACS Chemical Biology, vol. 11, no. 9, pp. 2541–2550, 2016.

[11] M. Zitnik, F. Nguyen, B. Wang et al., “Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities”, Information Fusion, vol. 50, pp. 71–91, 2019.

[12] T. P. Velavan and C. G. Meyer, “The COVID19 epidemic”, Tropical Medicine & International Health, vol. 25 , no. 3, pp. 278–280, 2020.

[13] Y. Li, C. Huang, L. Ding, Z. Li, Y. Pan, and X. Gao, “Deep learning in bioinformatics: introduction, application, and perspective in the big data era”, Methods, vol. 166, pp. 4–21, 2019.

[14] M. Karimi, D. Wu, Z. Wang, and Y. Shen, “DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks”, Bioinformatics, vol. 35, no. 18, pp. 3329–3338, 2019.

[15] S.W. Naidoo, N. Naicker, S.S. Patel, and P. Govender, “Computer vision: the effectiveness of deep learning for emotion detection in marketing campaigns”, International Journal of Advanced Computer Science and Applications, Vol. 13, no. 5, 2022.

[16] J. Kalezhi, M. Chibuluma, C. Chembe, V. Chama, F. Lungo, and D. Kunda, “A Cross Platform Contact Tracing Mobile Application for COVID-19 Infections using Deep Learning”, International Journal of Advanced Computer Science and Applications, vol. 13, no. 8, 2022.

[17] H. Öztürk, A. Özgür, and E. Ozkirimli, “DeepDTA: deep drug–target binding affinity prediction”, Bioinformatics, vol. 34, no. 17, pp. i821–i829, 2018.

[18] S. Krig, "Feature learning and deep learning architecture survey," in Computer Vision Metrics (Cham: Springer), 375–514, 2016.

[19] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications”, Neurocomputing, vol. 234, pp. 11–26, 2017.

[20] K. Cho, B. v. Merrienboer, D. Bahdanau et al., “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”, In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation at ACL, pages 103–111, Doha, Qatar, 2014.

[21] M. I. Davis, J. P. Hunt, S. Herrgard et al., “Comprehensive analysis of kinase inhibitor selectivity”, Nature Biotechnology, vol. 29, no. 11, 1046–1051, 2011.

[22] C. Samir, B. Assia, A. Adnane et al., “Discovery of potent SARS-CoV-2 inhibitors from approved antiviral drugs via docking and virtual screening”, Combinatorial chemistry & high throughput screening, vol. 24, no. 3, pp.441-454, 2021.

[23] A. Masoudi, Z. Mousavian, and J. H. Bozorgmehr, “Drugtarget and disease networks: polypharmacology in the postgenomic era”, vol. 17, 2013.

[24] Y. M. Sobhanzadeh, Y. Omidi, M. Amanlou et al., “Drug databases and their contributions to drug repurposing”. Genomics, vol. 112, no. 2, pp. 1087-1095, 2020.

[25] A. Ezzat, M. Wu, X.-L. Li et al., “Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey”, Briefings in Bioinformatics, vol. 20, no. 4, pp. 1337–1357, 2019.

[26] A. Karim, R. Parvin, P. Antti et al., “Deep learning in drug target interaction prediction: current and future perspectives”. Current Medicinal Chemistry, vol. 28, no. 11, pp. 2100-2113, 2021.

[27] A. S. Rifaioglu, E. Nalbat, V. Atalay et al., “DEEPScreen: high performance drug–target interaction prediction with convolutional neural networks using 2-D structural compound representations”, Chemical Science, vol. 11, no. 9, pp. 2531-2557, 2020.

[28] B. Shin, S. Park, K. Kang et al., “Self-attention-based molecule representation for predicting drug-target interaction”, 4th Machine Learning for Healthcare Conference, PMLR 106:230-248, 2019.

[29] D. Rogers and M. Hahn, “Extended-connectivity fingerprints. Journal of chemical information and modeling”, vol. 50, no. 5, pp.742-754, 2010.

[30] H. Zeng, M. D. Edwards, G. Liu et al., “Convolutional neural network architectures for predicting DNA protein binding. Bioinformatics”, vol. 32, no. 12, i121-i127, 2016.

[31] J. Lanchantin, R. Singh, B. Wang et al., "Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks”, Biocomputing, pp. 254-265, 2017.

[32] B. Alipanahi, A. Delong, M. T. Weirauch et al., “Predicting the sequence specificities of DNA- and RNAbinding proteins by deep learning”, Nature Biotechnology, vol. 33, no. 8, pp. 831-838, 2015.

[33] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition”. IEEE Transactions on Audio, Speech, and Langua.ge Processing, vol. 20, no. 1, pp. 30-42, Jan. 2012.

[34] A. Graves; A. Mohamed; G. Hinton, “Speech recognition with deep recurrent neural networks”, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6645–6649, 2013

[35] G. Hinton, L. Deng, D. Yu et al., “Deep neural networks for acoustic modeling in speech recognition” IEEE Signal Processing Magazine, vol. 29, no. 6, 82–97, 2012.

[36] D. Ciregan, U. Meier, J. Schmidhuber, “Multi-column deep neural networks for image classification”. Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649, 2012.

[37] J. Donahue, Y. Jia, O. Vinyals et al., “Decaf: a deep convolutional activation feature for generic visual recognition”, 31st International Conference on Machine Learning, PMLR, vol. 32, no. 1, pp. 647-655, 2014.

[38] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, 3rd International Conference on Learning Representations (ICLR), Hilton San Diego Resort & Spa, May 7–9, 2015.

[39] M. K. Leung, H. Y. Xiong, L. J. Lee et al., “Deep learning of the tissue-regulated splicing code”, Bioinformatics, vol. 30, no. 12, pp. i121–i129, 2014.

[40] H. Y. Xiong, B. Alipanahi, L. J. Lee et al., “The human splicing code reveals new insights into the genetic determinants of disease”, Science, vol. 347, no. 6218, 2014.

[41] J. Ma, R. P. Sheridan, A. Liaw et al., “Deep neural nets as a method for quantitative structure–activity relationships”, Journal of Chemical Information and Modeling, vol. 55, no. 2, pp. 263–274, 2015.

[42] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-regulated splicing code”, Bioinformatics, vol. 30, no. 12, pp. i121-i129, 2014.

[43] P. W. Hu, K. C. Chan, and Z. H. You, “Large-scale prediction of drug-target interactions from eep representations”, In 2016 international joint conference on neural networks (IJCNN), IEEE, pp. 1236-1243, 2016.

[44] K. Tian, M. Shao, Y. Wang et al., “Boosting compound-protein interaction prediction by deep learning”, Methods, vol. 110, pp. 64-72. 2016.

[45] M. Hamanaka, K. Taneishi, H. Iwata et al., “CGBVS - DNN: Prediction of Compound - protein Interactions Based on Deep Learning ”, Molecular informatics, vol. 36, no. 1-2, pp.1600045, 2017.

[46] L. Wang, Z.H. You, X. Chen et al., “A computational-based method for predicting drug–target interactions by using stacked autoencoder deep neural network”, Journal of Computational Biology, vol. 25, no. 3, pp.361-373, 2018.

[47] M. Wen, Z. Zhang, S. Niu et al., “Deep-learning-based drug–target interaction prediction”, Journal of proteome research, vol. 16 no. 4, pp.1401-1409, 2017.

[48] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud et al., “Automatic chemical design using a data-driven continuous representation of molecules”, ACS central science, vol. 4, no. 2, pp. 268-276, 2018.

[49] S. Jastrzębski, D. Leśniak, and W. M. Czarnecki, “Learning to smile (s)”, arXiv preprint arXiv: 1602.06289, 2016.

[50] Gomes, J., Ramsundar, B., Feinberg, E.N., and Pande, V.S., “Atomic convolutional networks for predicting protein–ligand binding affinity”, arXiv preprint arXiv: 1703.10603, 2017.

[51] M. Ragoza, J. Hochuli, E. Idrobo et al., “Protein–ligand scoring with convolutional neural networks”, Journal of chemical information and modeling, vol. 57, no. 4, pp. 942-957, 2017.

[52] I. Wallach, M. Dzamba, and A. Heifets, “AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery”, arXiv preprint arXiv:1510.02855, 215.

[53] P. W. Rose, A. Prlić, A. Altunkaya et al., “The RCSB protein data bank: integrative view of protein, gene and 3D structural information”, Nucleic acids research, p.gkw1000, 216.

[54] T. Pahikkala, A. Airola, S. Pietilä et al., “Toward more realistic drug–target interaction predictions”, Briefings in bioinformatics, vol. 16, no. 2, pp. 325-337, 2014.

[55] T. He, M. Heidemeyer, F. Ban et al., “SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines”, Journal of cheminformatics, vol. 9, no. 1, pp. 1-14, 2017.

[56] P. Kao, S. Kao, N. Huang, and Y. Lin, “Toward Drug-Target Interaction Prediction via Ensemble Modeling and Transfer Learning”, IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, pp. 2384-2391, 2021.

[57] Y. Jiaying, D. Robert, and P. Hu, “Predicting drug-target interaction network using deep learning model”, Computational Biology and Chemistry, Vol. 80, PP. 90-101, 2019.

[58] Y. Wang, Z. You, S. Yang et al., “A deep learning-based method for drug-target interaction prediction based on long short-term memory neural network”, BMC medical informatics and decision making, vol. 20 no. 2, pp.1-9, 2020.

[59] F. Li, Z. Zhang, J. Guan, and S. Zhou, "Effective drug–target interaction prediction with mutual interaction neural network", Bioinformatics, Vol. 38, no. 14, pp. 3582–3589, 2022.

[60] E.E. Bolton, Y. Wang, P.A. Thiessen, and S.H. Bryant, "PubChem: integrated platform of small molecules and biological activities", In Annual reports in computational chemistry, vol. 4, pp. 217-241, Elsevier, 2008.

[61] Y. Zhou, Y. Hou, J. Shen, Y. Huang, W. Martin, and F. Cheng, "Network-based drug repurposing for novel coronavirus 2019-nCoV/SARS-CoV-2", Cell discovery, vol. 6, no. 1, p.14, 2020.

[62] K. Huang, T. Fu, L. M. Glass et al., "DeepPurpose: A deep learning library for drug-target interaction prediction", Bioinformatics, vol. 36, no. 22-23, pp. 5545–5547, 2020.

[63] H. Huang, and X. Gong, "A review of protein inter-residue distance prediction", Current Bioinformatics, vol. 15, no. 8, pp.821-830, 2020.

[64] M. Bagherian, E. Sabeti, K. Wang, M.A. Sartor, Z. Nikolovska-Coleska, and K. Najarian, "Machine learning approaches and databases for prediction of drug–target interaction: a survey paper", Briefings in bioinformatics, vol. 22, no. 1, pp.247-269, 2021.

[65] Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

Development of a Smart Sensor Array for Adulteration Detection in Black Pepper Seeds using Machine Learning

Sowmya Natarajan[1], Vijayakumar Ponnusamy[2]
Department of ECE, SRMIST, Chennai, India[1, 2]

Abstract—**Black pepper is an expensive commodity with a high risk of adulteration. Ground papaya seed is the main adulterant in pepper because it cannot be discriminated visually. There are few destructive methods. Since pepper is costlier, non-destructive method of adulteration is must but it is challenging one. The existing non-destructive method uses costlier equipment, bulky, involve laboratory-based testing, time consuming in the process. To overcome the above issues, this article presents the development of Non-destructive E- nose gas sensor for pepper adulteration detection. This system determines the VOC in a controlled environment. The proposed system utilizes MQ2 and MQ3 gas sensor arrays to identify Volatile Organic Compounds present in pepper seeds to discriminate adulterant and non-adulterant sample. The sensor data are utilized to perform the qualitative analysis to determine the adulteration using a support vector machine learning algorithm. The proposed sensor system with Support Vector Machine learning algorithm outperforms in comparison with existing methods with 100% classification accuracy. Conclusion: The developed gas sensor system is connected to the internet via the IoT application model to show results on the web pages and enables access by the authenticated user from anywhere. Client server model with MQTT protocol is used for developing IoT application.**

Keywords—Gas sensor system; volatile organic compounds; pepper seeds; papaya seeds; machine learning

I. Introduction

In India, adulteration in food products is highly uncontrollable, mainly the products sold in semi–urban, urban slum areas and rural areas. Due to these substandard or poor-quality products, consumers are cheated by paying more prices [1]. Among the whole spice items, pepper is a significant herb utilized for seasoning food. It is known to be the "King of Spice." Black pepper is called piper nigrum. It consists of chemical, and nutritional components, namely vitamins (K, A and C), protein, starch, polyphenols, piperine, and essential oils. Traditionally, techniques like sensory, analytical evaluation, iodide test and microscopic determination were deployed for adulteration identification in food products.

Real-time controlling, low cost, high sensitivity, and simple operated electronic nose were employed to determine the freshness of packed food items. The model [2] proposes the determination of Total Volatile Organic Compounds (TVOCs) from packed foods to find their freshness. A sensor mote comprised of low-power metal oxide gas sensors and humidity sensors are employed for this purpose, which is self-powered from the far-field Radio Frequency Energy Harvesting (RFEH). Packed fish and pork samples are utilized for examining the TVOC, which are kept in the refrigerator at ambient temperature for eight days. The acquired sensor data are trained with the 1D Convolutional Neural Network (CNN), Multilayer perceptron, and Support Vector Machine (SVM). The Multilayer perceptron model achieves a maximum of 99% classification accuracy for packed pork and 93.3% of classification accuracy for packed fish.

Spice powders are commonly adulterated by wheat, corn and rice flour. The study [3] presents a DNA barcoding methodology for identifying adulteration in spice powder. This study employs 91 commercial spice varieties brought from retail shops to detect adulteration using DNA sequences (ITS2 and psbA- trnh- Gen bank accession number). Concluded that with the help of 78- ITS2 and 77- psbA- trnh DNA sequences(barcodes), it is possible to identify that out of 91 varieties, 39 samples are pure,43 samples adulterated, 4 were uncertain and five samples were undetected.

The following literature work discusses the determination of adulteration determination through spectral information and VOCs.

A. Related Work

The appearance of papaya seeds is similar to the black pepper seeds, so adulteration is made easier, but the determination of papaya seeds becomes tedious through normal visualization. The significant flavor in black pepper is a terpene which is a VOC. Significant key odorants of black pepper are β - pinene, α –pinene, myrcene, α – phellandrene, limonene, methyl propanal, 2, 3 methyl butanol, butyric acid etc. The work [4] develops Molecular Imprinted Polymer (MIP) coated gold Nano Particle (Au NP) sensor for the determination of α-pinene VOCs in black pepper. The molar ratio of α- pinene template to monomer is investigated on the MIP film adsorption characteristics. The molar ratio of the alpha-pinene template to monomer was achieved as 1:4 on MIP absorption characteristics.

Total flavonoids, phenol, and piperine are the bioactive compounds present in the pericarp of pepper berries. Pericarp determines the color intensity and texture of the pepper. The study [5] various methods to determine the chemical components in the pericarp of pepper berries. A colorimetric assay was deployed for the identification of total flavonoids and total phenols. It is identified that the total phenol content is

1421.95± 22.35mg/100 gram, and the total flavonoid is 983.82 ± 8.19mg/100g. High-Performance Liquid Chromatography (HPLC) was utilized for piperine compound identification and found as 2352.19 ± 68.88mg/100g. Gas chromatography/Mass spectroscopy (GC/MS) identifies the characteristic peak areas of VOCs at 510nm wavelength -total flavonoids and 750nm wavelength -total phenols.

To determine the presence of adulteration in the paprika powder, the work [6] proposed a portable NIR spectrometer. The paprika powder samples are adulterated with acacia gum, potato starch, and annatto powder samples at various concentrations ranging from 0-36%. The acquired spectral data are validated through Partial Least Squares- Discriminant Analysis (PLS-DA) and Partial Least Square Regression (PLSR) algorithms. PLS-DA outperforms better classification to differentiate the adulterated samples at an accuracy of 90%. This model can predict the adulteration of annatto powder with Root Mean Square Error Prediction (RMSEP) at the rate of 1.74 and R2P of 0.87.

Research work [7] discusses the detection of papaya powder in black pepper powder by utilizing Thin Layer Chromatography (TLC) and GC/MS technique. Experimentation made for 20g of addition of papaya seeds powder in 1000g of pepper powder. It includes the analysis of Column chromatography through Gas Chromatography/Mass Spectrometry (GC/MS), which indicates the "366nm wavelength" fluorescent spot marker at Rf = 0.943. The "366nm wavelength" also detects the mixture of 2- decenal, n-nonanal and trans 2 – undecenal aldehydes present in the papaya seeds.

Terpene gas detecting sensor array is developed to determine the concentration of terpene gas in pepper sample [8,9]. Molecular Imprinted Polymer (MIP) and combination of MIP and Conductive Polymer (MMC) sensors is developed and compared the sensitivity in detecting the Terpene, limonene, and α-terpene. Among the two sensors MOC outperforms with higher sensitivity. 500ppm terpene concentration is detected using MOC sensor at the rate of 6.5 and 100 times more sensitivity for limonene and α-pinene.

This research work developed Molecular Imprinted Polymer (MIP) coated with gold nano particle (AuNP) excited Localized Surface Plasma Resonance (LSPR) [10] for the detection of α- pinene vapor. MIP powder and coating film adsorption characteristics are found by using gas chromatography and mass spectrometry. Molar ratio of α-pinene to monomer is investigated on the MIP film adsorption characteristics. Molar ratio of alpha- pinene template to monomer achieved as 1:4. less than 3 sec of time is required to determine 90% of terpene vapour transmittance. Good selectivity and sensitivity of this developed MIP coated AuNP sensor is realized at 1400 r/ min at the rate of spin coating. On field responses also recorded for limonene, α- pinene, γ-terpene and also verified to be reversible, rapid and selective.

Sichuan pepper plays a significant role as a flavoring agent due to its unique aroma and taste. Another work [11] focused on determining adulterants in the Sichuan pepper by utilizing NIR spectroscopy. Wheat bran, rosin powder, rice bran, and corn flour mixtures are added to Sichuan pepper intentionally to gain extra profit in the market. The acquired NIR spectral samples are validated through Partial Least Squares (PLS) analysis. It predicts the adulteration level in Sichuan pepper with the coefficient of prediction of 0.994 rosin powder mixture, 0.967 corn flour mixture, 0.948 rice bran mixture, and 0.969 wheat bran mixture, respectively.

Near-infrared and Fourier Transform - Infrared spectroscopy (NIR & FT-IR) [12] were used to determine adulteration in black pepper. NIR-FTIR spectroscopic data are processed with chemometric analysis to detect adulterants in black pepper. The adulterants are defatted spent black pepper, chili, papaya seeds, black pepper husk, and pinheads. The NIR& FT-IR spectral data are combined and applied to the binary classifier using Orthogonal Projections to Latent Structures Discriminant Analysis (OPLS-DA) model, which achieves 98% classification accuracy for adulterant.

NIR-Hyper Spectral Imaging (HSI) was employed to identify papaya seeds and their powder adulteration in black pepper, berries, and powder [13]. Adulterations of papaya seeds are made at different levels of concentrations with black pepper. Hyperspectral images are obtained and classified through Soft Independent Modelling Class Analogy (SIMCA) and Partial Least Squares (PLS) analysis. SIMCA model achieves better classification accuracy of 90% with less than 8% of error rate for adulteration determination in black pepper, berries, and its powder. PLS achieves a better prediction capability of $R^2P>0.8$, and the ratio of prediction deviation is higher than 2.5.

Among all the country's pepper production, Sarawak state in Malaysia produces a superior quality of pepper. It deserves a large price in the global market due to its better quality. Sarawak pepper samples are compared with the Indian ground black pepper sample for species adulteration determination. Different VOCs in the black pepper was determined by utilizing the geo–tracing module built with Metal-oxide Semiconductor (MOS) gas sensor array [14]. Totally 200 counts of black pepper samples are obtained from India and Malaysia. The sensor data are processed and analyzed with Principle Component Analysis (PCA). The PCA result produces an overall classification accuracy of 92.5% for discriminating the India and Malaysia pepper samples.

Identification of Volatile Organic Compounds (VOCs) is carried out with the E- nose model [15]. Gas chromatography Mass Spectrometry methods also carried out for the aphid attacks in the tomato plants also without infestation. Principle Component Analysis (PCA) delivers better results for the infected and healthy plants at an accuracy of 86.7%.

Table I gives a brief discussion of the methodologies employed for pepper VOCs/adulterant detection. MIP-coated sensor motes, E-nose sensor system and spectroscopic methods are discussed for the adulteration determination in the whole pepper seeds and its powder sample. Among all the methods, the NIR-FTIR model achieves a maximum of 98% classification accuracy in discriminating the adulterants in the black pepper seeds.

TABLE I. METHODOLOGIES FOR PEPPER VOLATILE ORGANIC COMPOUNDS (VOCS) AND ADULTERANT DETERMINATION

Literature work employed	Techniques& Methodology employed	VOCs/Adulterant components detected in pepper samples	Accuracy achieved
Wilde, A.S., (2019) [12]	NIR-FTIR&OLS-DA	Defatted spent, husk and pinheads	98%
Lee, H.E., (2020) [14]	Geo–tracing with Metal-oxide Semiconductor (MOS) gas sensor array&PCA	Sarawak pepper	92.5%
Iqbal, N., (2010) [9]	E-nose sensor system& pattern analysis	Terpenes (VOCs)	<20%
Paradkar, M.M (2014) [7]	Gas Chromatography/ Mass spectrometry& Pattern recognition	Papaya powder	94.3%

Spectroscopic, E-nose sensor array system, MIP coated sensor mote and geo-tracing techniques were employed to determine the adulteration in pepper samples. Literature works focus on determining the Volatile Organic Compounds of pepper samples and the presence of other adulterants in pepper seeds. Moreover, these experimental methods are carried out in a lab environment; samples are analyzed in whole and powdered form, and it takes time delay to produce the results. Real-time data analysis enables the system to be more effective and predict the results on the field spot. To overcome these drawbacks, this research work contributed as follows:

- Design a low-cost, portable gas sensor array system for rapid data collection in the field.
- Machine Learning algorithm-based adulteration detection mechanism is developed to handle the variation factor such as placing the sensor inside a measurement chamber, aged pepper sample, sensor placement during data collection and able to achieve an accuracy of 100% adulteration determination.
- The designed sensor system is IoT enabled to visualize and update the results on the webpage, which can be accessed by an authenticated end user anywhere.

The article is organized as follows: Section II describes the materials and methods employed for adulteration determination. Cost comparison analysis. Section III discusses the results of SVM algorithm for classification of adulteration. Section IV conclude the research work.

II. MATERIALS AND METHODS

This section discusses the working principle of the designed systems. The Artificial Intelligence (AI) enabled sensor system design, sample preparation, data collection, and processing of them using machine learning algorithms were discussed.

A. *Design of Gas Sensor Array System for Pepper Adulteration Detection*

The designed sensor array with an Artificial Intelligence-enabled system is shown in Fig. 1 which composes of MQ series MQ2 and MQ3 gas sensors for volatile organic compounds, gas detection from pepper samples, Arduino – UNO board, Raspberry – pi module is used for data acquisition processing and classification of gas sensor data.

Finally, the output of the test results of the adulterant detected is communicated to the web server and can be accessed by the authenticated users through the Graphical User Interface (GUI) of the web page.

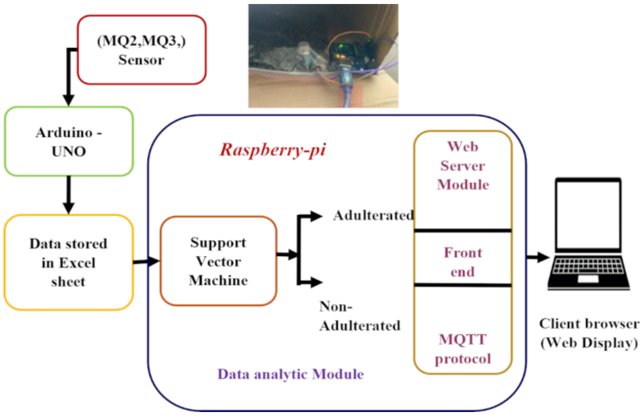


Fig. 1. System model of artificial intelligence (AI) enabled pepper adulteration determination.

Fig. 1 shows the general block diagram for the pepper adulteration detection system. The samples and sensors are kept inside the zip lock bag. Arduino UNO acquires the gas sensor data and stores it in an excel sheet. The stored information is transferred to the Raspberry Pi module through a serial port for data processing. The machine learning algorithm of the Support vector machine is utilized to classify the adulterated and unadulterated black pepper samples. Message Queuing Telemetry Transport (MQTT) protocol is utilized for communicating the result of classifiers to the webpage running in the web server module. Graphical User Interface designed to display the acquired sensor data results on the web page.

The following section explains the individual blocks of the sensor system array.

1) MQ2 sensor: MQ2 sensor module detects gas compounds such as Propane, hydrogen, Liquefied Petroleum Gas (LPG), propane, methane, smoke, carbon monoxide, and alcohol. The resistance of the sensor is directly proportional to the concentration of the volatile organic compound. The resistance measurement of the sensor detects the presence of the piperine compound. The concentration of piperine in the black pepper sample is about 2,531-8,073mg/100g. Resistance of the MQ2 sensor produces effective gas detection at approximately 200-1000 Parts Per Million (ppm) (i.e., 200mg/L-10,000mg/L) in the gas.

2) MQ3 sensor: MQ3 detects an alcohol, terpene, methane, hexane, LPG, carbon monoxide, and benzene detector. Resistance of the sensor varies according to the changes of volatile organic compounds in the measuring environment. This sensor works well for detecting gas components from 0.05-10ppm (i.e., 0.05mg/L-10mg/L). The terpene compound volatile concentration in black pepper determined ranges from 5.1%-28.7% per 100 grams.

3) Arduino Uno: Arduino UNO acquires the analog signal from the sensor system and converts it into digital values through the analog-to-digital converter. Raspberry Pi processes the received data from the Arduino UNO board through the serial port.

4) Raspberry-pi (data analytic module): Raspberry-pi receives the Arduino data and analyses the data for classification of adulteration. The sensor data are pre-processed to remove the outlier data. Using the machine learning algorithm of the SVM model, the data are classified as adulterant or non-adulterant. The results are communicated to the web display through the MQTT protocol.

5) Web display: The web page is designed to display the results of the detected adulterants with password authentication in the client browser. The user has to provide the login credentials to view the results of the tested samples. Only authenticated users can access the results of the tested samples. MQTT protocols transmit/receive the information between the web server module and raspberry-pi.

B. Sample Preparation and Data Collection

The pepper and papaya seed samples are brought from the local market and fruit shops for testing. The papaya and pepper samples are mixed at different proportions to make adulterated samples.

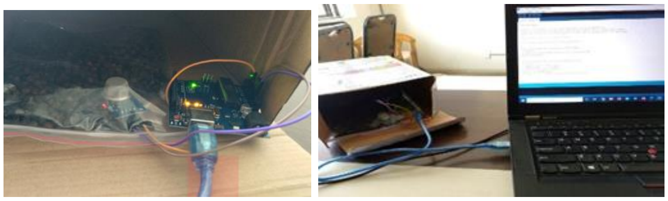


Fig. 2. Experimental setup for data acquisition.

Sample preparation steps are discussed and listed in Table II. The first set of experiments was carried out for 50 grams of pepper adulterated with 5 grams of papaya seeds (10%). Likewise, the adulterated samples are prepared at various concentrations in the ratio of 5:50, 10:50, 15:50 and 20:50, which means that 5 grams of papaya in 50 grams of pepper on top of it 5 grams of papaya in 50 grams of pepper. Similarly, the procedure was repeated for other concentration levels of pepper adulteration mixture.

TABLE II. SAMPLE PREPARATION

S.No	Pepper sample (in grams)	Papaya seeds (in grams)	% of the concentration of adulterant (papaya mixture in pepper sample)
1.	50 grams	5gms	10%
		10gms	20%
		15gms	30%
		20gms	40%
2.	100 grams	10gms	10%
		20gms	20%
		35gms	35%
3.	150 grams	15 gms	10%
		30gms	20%
		45gms	30%
4.	400grams	40gms	10%
		80gms	20%
		120gms	30%
		160gms	40%
5.	500 grams	100 gms	20%
		150gms	30%
		200gms	40%
		250gms	50%
6.	600grams	60 gms	10%
		120gms	20%

*gms- grams

Fig. 2 shows the placement of sensors on the sample inside a zip lock air-tight bag and the sensor data acquisition using the data streamer of an excel sheet through Arduino Uno. The number of sensor data acquired from the pure pepper and papaya seeds adulterated samples at various concentrations of the mixture is illustrated in Table III.

TABLE III. DATA ACQUIRED FOR VARIOUS LEVELS OF ADULTERATION

S. no	Pure and adulterant Cases	Total no of samples acquired
1.	50(Pure)+[5+5+5+5] grams	1500
2.	100(Pure)+[10+10+15] grams	1300
3.	150 (Pure)+ [15+15+15] grams	1200
4.	400(Pure)+[40+40+40+40]grams	1500
5.	500(Pure)+[100+50+50+50] grams	1400
6.	600(Pure)+[60+60]	900
Total no of samples		7800

Table III shows the number of sensor data obtained for each case of adulteration. Total number of 7800 spectral sensor communicated through MQTT protocol to Raspberry Pi for machine learning algorithm classification.

In Table III, “50 Pure+ [5+5+5+5]" describes that in 50 grams of pure pepper, 10% addition of adulterant (i.e., 5 grams of papaya seeds). Followed by adding 5 grams of papaya seeds three times and sensor data acquired for each adulteration addition. Likewise, the papaya seeds are added at various concentration levels in 100 grams, 150 grams, 400 grams, 500 grams and 600 grams of pure pepper samples.

C. *Cost Analysis*

This section compares the cost of the proposed sensor system design with the existing sensor module for pepper adulteration determination and is presented in Table IV. Our proposed work mainly focuses on designing low-cost, portable, rapid adulteration detection.

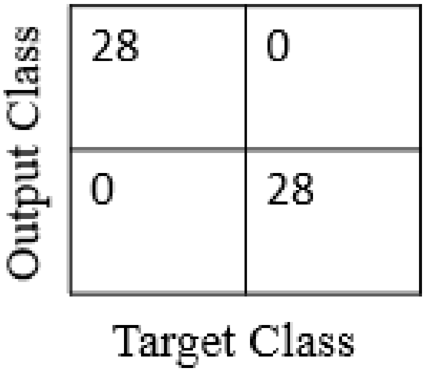
Table IV list the equipment utilized for pepper adulteration detection with its cost details. The proposed gas sensor system consists of a Raspberry –pi module of ₹ 4900/-, Gas sensors (MQ2, MQ3) cost around ₹ 500/- and Arduino UNO of ₹ 1850/-. The further jumper cable and USB cable are utilized. The designed sensor system cost is near to ₹ 7500/-.

TABLE IV. COST DETAILS

Equipment name	Adulterant determination	Cost
NIR-FTIR [wilde.A.S., 2019] [12]	defatted spent, husk and pinheads	₹ 5,86,528/-
GC-MS[Paradkar, M.M (2014)] [7]	Papaya powder	₹10,00,000/-
MOS- gas sensor array Lee, H.E,. (2020) [14]	Species/country origin variation (pepper)	₹15,000/-
Proposed gas sensor system	Papaya seeds	₹ 7500/-

III. RESULT AND DISCUSSION

This section delineates pepper adulteration identification using a support vector machine algorithm. The acquired gas sensor samples are collected and arranged so that 15% of the whole data set is assigned for the validation set, 70% is applied for the training set, and the remaining 15% is allotted for the testing phase.



Output Class \ Target Class		
	28	0
	0	28

Fig. 3. Confusion matrix binary Classifier.

SVM-based binary classification model could classify whether an adulterant is present or not. The support vector machine algorithm is trained using various level adulterant mixed data as in Table III.

Fig. 3 shows the confusion matrix for the binary class classification model. An accuracy of 100% is achieved for adulterant discrimination.

Fig. 4 shows the scatter plot MQ2 and MQ3 sensor data between pure pepper and adulterated samples. The response was recorded for the MQ2 and MQ3 sensor datasets. In the scatter plot, data lies on the diagonal lines with a different cluster that comes with points of 130(green points) of MQ2& 465 (green points) of MQ3. From the scatter plot, the adulterant's presence can be classified successfully.

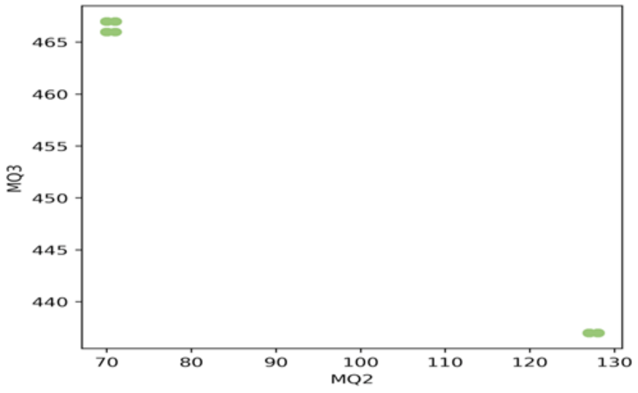


Fig. 4. Scatter plot of MQ2 and MQ3 sensor.

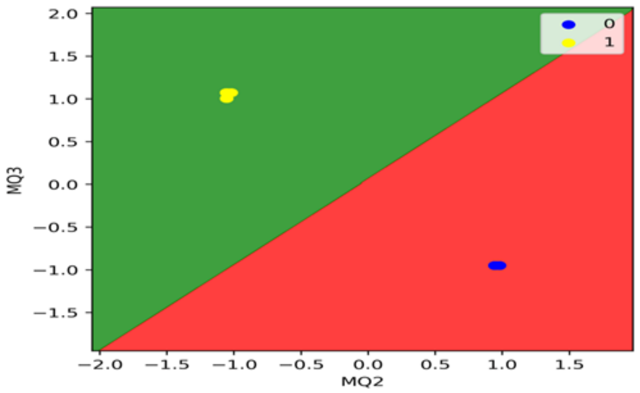


Fig. 5. Cluster plot of MQ2 and MQ3 sensors

In Fig. 5, the SVM algorithm shows that the data are fitted correctly according to their respective classes. The classification accuracy of the SVM model achieves 100%. The diagonal line indicated the separation of data and mapped in two regions (-1, 1) and (1,-1) successfully. It infers that the data are classified into their respective classes.

A. *Web Interface*

Our gas sensor model is designed with an application to give a real-time web interface for visualization of the detected adulterants. This web interface enables users to access the adulterant result anytime from anywhere.

TABLE V. COMPARISON CHART

Literature work employed	Techniques & data analysis methods	Compounds identified in the black pepper/powder sample	Accuracy achieved
Wilde, A.S., (2019) [12]	NIR-FTIR&OPLS-DA	defatted spent black pepper, black pepper husk and pinheads	98%
Lee, H.E., (2020)[14]	Geo–tracking with Metal-oxide Semiconductor (MOS) gas sensor array&PCA	50 various Sarawak pepper	92.5%
Paradkar, M.M (2001) [7]	Gas Chromatography/Mass Spectrometry& TLC pattern	Papaya powder	94.3%
Proposed method	Gas Sensor system& Binary SVM classifier	Papaya seeds	100%

Table V provides some of the existing works employed for pepper adulteration identification using different methodologies and achieves maximum classification accuracy of 98%. In Table V. the research work discussed deploying laboratory evaluation methods, which are expensive, consume time for determination, and are not portable. Our proposed system is rapid, portable, and low-cost for the determination of papaya adulteration in black pepper and achieves 100% of accuracy.

IV. CONCLUSION

Nowadays, adulteration in food is one of the serious problems faced by consumers. New creative, rapid, non-destructive, portable techniques need to develop for adulteration detection. Many electronic nose sensor systems can be able to detect adulteration in pepper, but those techniques would consume time to deliver the results with moderate accuracy. A low-cost gas sensor array system was developed to detect adulterants rapidly. The proposed system detects adulterants in pepper with gas sensors of MQ2 and MQ3. The sensor readings are processed with a support vector machine algorithm for adulterant classification, which achieves 100% accuracy of detection. The proposed gas sensor array module output performance can be viewed and accessed by the authenticated user through the GUI of the web interface from anywhere. In the near future, work can be taken forward to determine the various concentration levels of adulterants mixed in pepper seed samples by utilizing other gas sensor modules.

ACKNOWLEDGMENT

Sowmya Natarajan involves in collecting the pepper and papaya samples and acquired the gas sensor data by testing the samples with various levels of adulterated and unadulterated samples. The data processing and classification and the research direction contributed by Vijayakumar Ponnusamy.

REFERENCES

[1] Pardeshi, S., 2019. Food adulteration: its implications and control approaches in India. International Journal of Trend in Scientific Research and Development, 3(5), pp.1117-1123.

[2] Lam, M.B., Nguyen, T.H. and Chung, W.Y., 2020. Deep learning-based food quality estimation using radio frequency-powered sensor mote. IEEE Access, 8, pp.88360-88371.

[3] Lam, M., Shi, Y., Sun, W., Wu, L., Xiong, C., Zhu, Z., Zhao, H., Zhang, B., Wang, C. and Liu, X., 2019. An efficient DNA barcoding based method for the authentication and adulteration detection of the powdered natural spices. Food Control, 106, p.106745.

[4] Chen, B., Liu, C. and Hayashi, K., 2014. Selective terpene vapour detection using molecularly imprinted polymer coated Au nanoparticle LSPR sensor. IEEE Sensors Journal, 14(10), pp.3458-3464.

[5] Lee, J.G., Chae, Y., Shin, Y. and Kim, Y.J., 2020. Chemical composition and antioxidant capacity of black pepper pericarp. Applied Biological Chemistry, 63(1), pp.1-9.

[6] Oliveira, M.M., Cruz-Tirado, J.P., Roque, J.V., Teófilo, R.F. and Barbin, D.F., 2020. Portable near-infrared spectroscopy for rapid authentication of adulterated paprika powder. Journal of Food Composition and Analysis, 87, p.103403.

[7] Paradkar, M.M., Singhal, R.S. and Kulkarni, P.R., 2001. A new TLC method to detect the presence of ground papaya seed in ground black pepper. Journal of the Science of Food and Agriculture, 81(14), pp.1322-1325.

[8] Lee, S.P., 2015, June. Terpene sensor array with bridge-type resistors by CMOS technology. In IOP Conference Series: Materials Science and Engineering (Vol. 87, No. 1, p. 012065). IOP Publishing.

[9] Iqbal, N., Mustafa, G., Rehman, A., Biedermann, A., Najafi, B., Lieberzeit, P.A. and Dickert, F.L., 2010. QCM-arrays for sensing terpenes in fresh and dried herbs via bio-mimetic MIP layers. Sensors, 10(7), pp.6361-6376.

[10] Chen, B., Liu, C. and Hayashi, K., 2014. Selective terpene vapor detection using molecularly imprinted polymer coated Au nanoparticle LSPR sensor. IEEE Sensors Journal, 14(10), pp.3458-3464.

[11] Wu, X.Y., Zhu, S.P., Huang, H. and Xu, D., 2017. Quantitative identification of adulterated Sichuan pepper powder by near-infrared spectroscopy coupled with chemometrics. Journal of Food Quality, 2017.

[12] Wilde, A.S., Haughey, S.A., Galvin-King, P. and Elliott, C.T., 2019. The feasibility of applying NIR and FT-IR fingerprinting to detect adulteration in black pepper. Food control, 100, pp.1-7.

[13] Orrillo, I., Cruz-Tirado, J.P., Cardenas, A., Oruna, M., Carnero, A., Barbin, D.F. and Siche, R., 2019. Hyperspectral imaging as a powerful tool for identification of papaya seeds in black pepper. Food Control, 101, pp.45-52.

[14] Lee, H.E., Mercer, Z.J., Ng, S.M., Shafiei, M. and Chua, H.S., 2020. Geo-Tracing of Black Pepper Using Metal Oxide Semiconductor (MOS) Gas Sensors Array. IEEE Sensors Journal, 20(14), pp.8039-8045.

[15] Cui, S., Inocente, E.A.A., Acosta, N., Keener, H., Zhu, H. and Ling, P.P., 2019. Development of fast e-nose system for early-stage diagnosis of aphid-stressed tomato plants. Sensors, 19(16), p.3480.

Digital Signature Algorithm: A Hybrid Approach

Prajwal Hegde N[1], Dr. Veena Devi Shastrimath V[2]
Dept. of Electronics and Communication Engineering
Nitte (Deemed to be University), NMAM Institute of Technology, Nitte
Karnataka, India[1, 2]

***Abstract*—Security is one of the most important issues in layout of a Digital System. Communication these days is digital. Consequently, utmost care must be taken to secure the information. This paper specializes in techniques used to defend the facts from thefts and hacks the use of quit-to-cease encryption and decryption. Cryptography is the important thing technique related to Encrypting and Decrypting messages. We use Digital Signature preferred (DSS) and the Digital Signature Algorithm (DSA). The code for this algorithm is written in MATLAB. The DSA Algorithm is commonly used in cryptographic applications to provide services such as entity authentication, key transit, and key agreement in an authenticated environment. This structure is related with steady Hash Function and cryptographic set of rules the government groups in USA as it is taken into consideration to be one of the safest approaches of protection system. This fashion- able could have a top notch effect on all of the Government Agencies and Banks for protective the facts.**

Keywords*—*DSA; digital signature algorithm; hash function; public key; private key; RSA

I. Introduction

Cryptography is the system of conversion among undeniable text to cipher textual content and to straightforward text. Fig. 1 shows how Cryptographic manner is achieved:

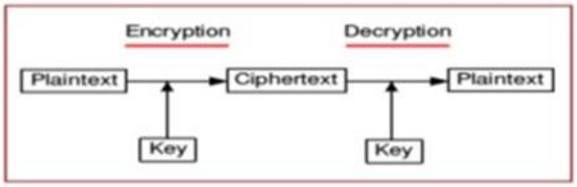


Fig. 1. Cryptographic operations.

- The sender converts the obvious textual content to its cipher shape with a key. This manner is called as Encryption.
- The cipher textual content is received through the receiver.
- The Received cipher text is transformed to a readable shape with a key. This technique is referred to as Decryption.

A. *Types of Cryptography*

As shown in the Fig. 2, the entirety of the cryptographic algorithms can be classified in to three main categories

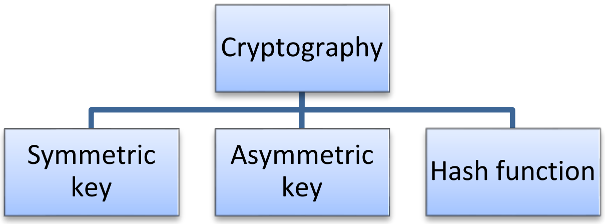


Fig. 2. Types of cryptography.

1) Private key cryptography: Private Key cryptography, also known as symmetric key cryptography, employs the same key for both message sign encryption and decryption. Symmetric key cryptography is tremendously effective since there may be no time put off for Encryption and Decryption of messages [1].

2) Asymmetric key: Asymmetric keys are the foundation of Public Key Infrastructure (PKI), a cryptographic scheme that requires two different keys, one to lock or encrypt the plaintext and one to unlock or decrypt the cipher text. It uses public key for Encrypting messages and Private key for the Decryption of messages. Public key can be shared to everybody but private key's kept secret [2].

3) Hash function: A hash function is a mathematical algorithm that takes an input, or message, and produces a fixed-size output, also known as a hash or digest. The output is deterministic, meaning that the same input always results in the same output, making it challenging to find two inputs that produce the same hash. This property makes hash functions useful in various applications, including data integrity verification and digital signature generation. Examples of widely-used hash functions include SHA-256, SHA-3, and MD5.The following picture illustrated hash function.

II. Related Work

Cryptography is a branch of cryptology that deals with designing algorithms for encryption and decryption to ensure the confidentiality and/or authenticity of messages. In 1991, the DSA was proposed by the U.S. [3]. The growing use of services like e-commerce and open network communications has highlighted the crucial role of public key cryptosystems as security solutions [4]. In public key cryptographic system, Digital Signature provides vital sort of authentication [2]. A digital signature generated as a checksum by making use of the text/data to which it will be later appended to and will look like a whole text [4]. Since the generation of this digital signature is dependent on the transmitted message along with the secrete key, one cannot easily understand the transmitted data or can reproduce the signature nor can they be able to tamper the transmitted data without getting noticed by the transmitter/receiver. A virtual signature is computed using a set of regulations and a fixed of parameters so as to conceal the identity of the original signatory and also to prove the integrity of the records [4].

Digital signatures have been in use since the early days of digital data transmission, due to the discovery of one-way functions. Many digital signature schemes have been shown to be secure under certain theoretical assumptions. A recent advancement has been the development of an offline signature verification system based on a displacement extraction technique, where a questionable signature is compared to a valid one. The proposed digital signature uses a set of rules and generates dynamic values through a new hash feature.

A signature scheme is a technique for signing an electronic communication that has been saved. A signed communication can therefore be sent across a computer network. Prior to studying various signature methods in this section, let's first talk about some key distinctions between traditional and digital signatures. The first concern is file signing. A traditional signature is one where the document being signed really contains the signer's signature. However, a virtual signature is now not connected bodily to the message this is signed, so the set of rules this is used need to by some means bind the signature to the message. A virtual signature, however, is no longer physically attached to the message it is signed, therefore the set of rules that are utilized must somehow link the signature to the message. The verification issue comes next. By comparing it to other real signatures, a conventional signature is made legitimate.

For instance, when a customer conducts a credit card purchase, the salesperson is required to check the signature on the income slip against the signature on the bottom back of the credit card to confirm it. Of direction, this approach isn't always reliable because it's quite easy to fake another person's signature. In contrast, a publicly acknowledged set of verification procedures may be used to produce digital signatures. Thus, all and sundry can affirm a virtual signature [5].

Chaum is credited with originating the idea of blind signatures [9]. This technology enables a user to have a message signed by another user without divulging any details about the message. Blind signatures have a wide range of practical uses, including electronic cash, untraceable electronic mail, electronic voting systems, time-stamping and anonymous access control.

To overcome several security issues identified in RSA algorithm [1], Gupta et al. suggested an hybrid approach for encryption and decryption algorithms by making use of RSA and Diffie-Hellman algorithm[DH], Named after the founder by Whitfield Diffie and Martin Hellman. A DH algorithm is an extensive algorithm which is majorly used in several internet connectivity protocols. Several example protocols like SSL, IPsec, SSH [1,9,11]. The DH is based on two key principles, a public and a private key. The transmitter and the receiver exchange the secret value among them by making use of these available key values [10]. In the proposed algorithm [1], the authors have combined both algorithm to get a secure and efficient cryptographic environment by exploiting the benefit of security from public key system and the reduction of computing time from secret key system.

The primary goal of Gupta et.al, proposal's combination of these two algorithms is to create a better and more secure cryptosystem by using the speed and security of the secret key system and the public key system, respectively. Users often find it simple to interact securely across open networks using this hybrid technique, especially when sending confidential messages or information. For improved algorithm functioning, efficiency can be altered in terms of time complexity. Additionally, the size of the keys used for encryption and decryption can be further decreased [12].

In contrast to the original RSA technique, which relies on 1024-bit prime numbers, the Iswari et al approach uses 256-bit prime numbers to reduce the computing time needed for key creation. Due to problems with factorization and discrete logarithm calculations, RSA and ElGamal are combined to retain security factors and complexities even when little bit prime numbers are utilised [13].

Patidar and Bhartiya introduced an innovative concept to enhance the conventional RSA method during information transmission between two parties through a network. They utilized a third prime number to form a modulus n that is harder to decipher by outsiders. This improvement combines a refined version of RSA with a unique design approach. Although the technique speeds up communication encryption and decryption, maintaining secure key storage remains crucial as it safeguards against potential attacks [14].

Shikha et al. presents a modified approach to the traditional RSA algorithm that enhances its security by incorporating exponential powers, multiple public keys, and K-NN algorithm. The modified approach also provides verification at both sender and receiver sides, which ensures authenticity of a message. This approach reduces encryption and decryption time for encrypting and decrypting input messages while making it difficult for intruders to hack the information being transmitted [15].

Jaju et al. proposed an updated version of the RSA technique that utilizes three irreducible numbers instead of two prime random numbers for calculating the common modulus, and passing the value of X in both the public and private keys instead of n. This modification is believed to enhance security and improve speed compared to the original RSA technique. The three prime integers, p, q, and r, must be factored to determine the common modulus n, which is a computationally intensive task, making the system more secure. In case of a factorization attack, it becomes difficult to uncover the value of n as X is included in the public key, rather than n. While the new method offers improved security and faster key generation, the encryption and decryption processes take longer than in the original RSA approach [16].

By removing the distribution of n, a big number whose factor, if uncovered, weakens the RSA method, Minni et al. presented a different safe technique. The updated algorithm's slower key generation time as compared to the RSA technique is one of its drawbacks [17].

Thangavel and colleagues proposed a modified version of the RSA public-key cryptography system, utilizing four prime numbers. The four primes are used to calculate the value of N and determine the values of E and D. The computation of E is more complex as it requires finding the values of e1 and e2 before calculating E1. This added step makes the system more secure and longer to attack. The only value made public is n, so an attacker with knowledge of n cannot determine the other primes and uncover the values of N and D. The system's complexity is further increased by the addition of the E1 parameter. The authors demonstrated the superior security of this approach, making it a safer alternative to the conventional RSA. [18].

Islam et al. proposed a modified RSA (MRSA) scheme to address some weaknesses in RSA algorithm computation. The MRSA approach uses 'n' unique prime numbers for key creation, with three components in both the private and public key. The component N is the product of four large prime integers (w, x, y, and z) selected randomly. The public key consists of three elements, e, f, and N, where e and f are selected randomly from a group of three. In their implementation, the authors noted that the MRSA has different crucial parameters that impact the security and speed of the algorithm. [19].

III. Proposed Algorithm

A. RSA Algorithm

RSA is named behalf of its inventors [6]. RSA is an era for Encryption and Decryption of messages. RSA is primarily based on Asymmetric Key Cryptography which means that it makes use of two keys, one for Encryption and Other for Decryption. One key is kept Secret and the other is kept Public [7]. The methods of cryptography may be demonstrated as steady until it's cracked. Since RSA uses very massive numbers it is difficult to think out which is the variety taken. For Example, if one hundred digit numbers are taken for p, q. The end result 'n' can be around 200-digit quantity. The recognized Factoring Algorithm will take a piece time for an attacker to crack the records. Any Cryptographic technique which cannot be cracked without difficulty are referred to as steady, as of now RSA algorithm is secured.

In the Standard RSA algorithm, two additional prime numbers are used in the Enhanced RSA (ERSA) [10] algorithm. This concept was inspired by the High Speed and Security RSA algorithm [11], which used two random numbers for key generation.

B. DSA Algorithm

Digital signature widespread (DSS) is used to verify the originality of the Digital Messages or the documents dispatched [8]. Digital signature is a cryptographic cost, which is observed the usage of most effective from the message signals and the private key owned via the non-public key holder. By virtual signature we can offer security for privacy, Authentication, Integrity, and Non-repudiation. There are three main virtual signature techniques below Digital Signature Standard. They are the DSA Algorithm, the RSA Algorithm.

DSA algorithm works on the premise of public key cryptography [4].DSA set of rules is used by the receiver of a message to verify whether the message is changed or is it in its unique form. DSA uses Public key to verify the sender's message, but verifying is complicated compared to RSA.DSA set of rules works on three steps like,

- Key generation
- Signing
- Verification

Fig. 3 shows how the Encryption of message sign is executed; the message sign is sent through Hash characteristic to generate a hash code [3].

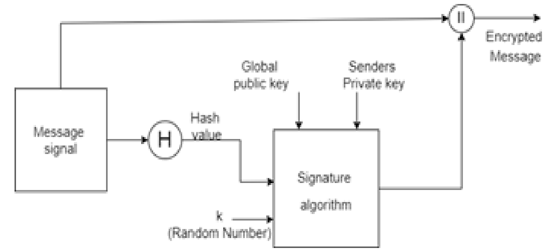


Fig. 3. DSA encryption process.

Then the hash code and random variety 'okay' is given as an input for signature algorithm in conjunction with worldwide public key and sender's private key. Then the message signal and signature could be appended to get an encrypted message.

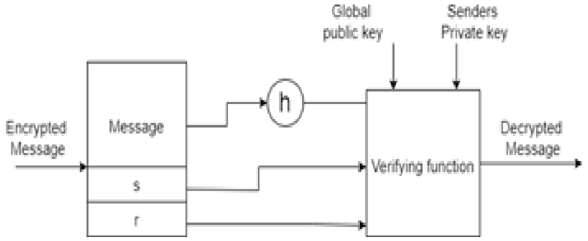


Fig. 4. DSA decryption process.

Fig. 4 shows how the Decryption of message sign is finished, once the Encrypted message is obtained with the resource of the receiver, he desires to decrypt the message to get decrease returned the unique message signal [4]. The Encrypted message signal will encompass the proper message sign, signature parameters like s and r. The message signal is given to the verifying characteristic, at the side of it global public key and sender's personal secret is given to it. And we get the decrypted fee that is not anything however the parameter "v".

C. Hash Functions

It is essentially meaningful for Hash features to compress data such that the output is comparatively shorter than the input, and possess the traits of a great hash function. There are three most important characteristics:

- The information that is being hashed needs to be fully determined via the hash fee.
- The hash characteristic makes use of all the enter records. The complete input statistics must be used by the hash characteristic.
- The hash function uniformly distributes the information across the entire set of possible hash values.

A Hash characteristic is a mathematical feature that converts an input fee into a compressed numerical value – a hash or hash fee. The period of the output always relies upon on the hashing set of rules. The maximum popular hashing algorithms may have a hash period ranging from 160 to 512.

IV. Result and Discussion

The following section discuss about the performance evaluation of proposed algorithm in comparison with standard RSA algorithm.

A. Execution Time Analysis

Table I depicts the overall time taken for the process of encryption and decryption using various algorithms. In the proposed algorithm the Block cipher symmetric algorithm is used for key selecting technique. It can be observed that even though Enhanced RSA has less computation time. In spite of using the block cipher approach for encryption and decryption the proposed algorithm still gives a better time results when compared to either the RSA algorithm or DSA algorithm alone.

TABLE I. Key Selection Procedure and Execution Timing

Algorithm	Key Selection Procedure	Execution Timing
RSA	Any two significant primes	5.9 Seconds
Enhanced RSA	Any two significant primes	2.9 Seconds
DSA	Using Block cipher symmetric algorithm	5.9 Seconds
Proposed DSS with E-RSA	Using Block cipher symmetric algorithm	3.9 Seconds

B. Visual Analysis of Digital Signature

The proposed approach of Digital Signature Scheme based totally on the linear block cipher RSA basically symmetric key set of rules. However, in this case, we employed an asymmetric key set of rules to create the symmetric key technique and applied it in a digital signature scheme. In addition to acknowledging informed agreement and acceptance by a signatory, digital signatures can offer added guarantees of the proof of provenance, identity, and standing of a digital document. The suggested digital signature scheme's actual structure is shown in Fig. 5 to Fig. 9 depicts the assessment performance of recent Digital Signature Scheme. Algorithms are simulated using MATLAB Tool.

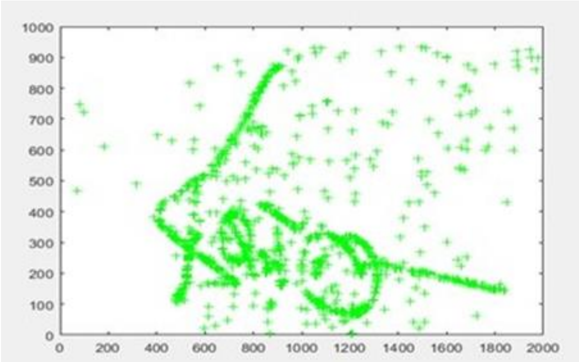


Fig. 5. Signing of sample signature.

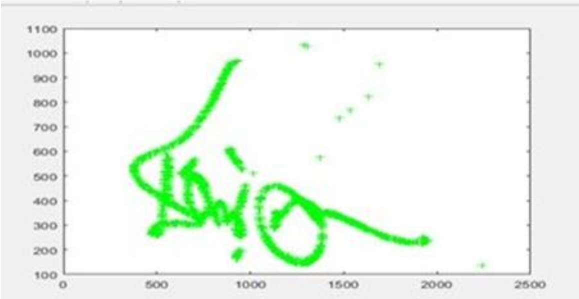


Fig. 6. Identification process of sample signature.

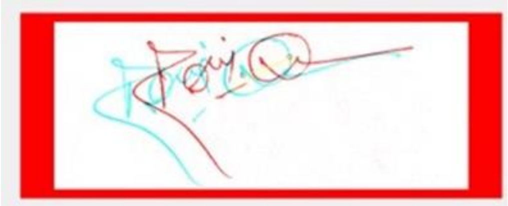


Fig. 7. Signature verification 1.

Fig. 8. Signature verification 2.

Fig. 9. Verified signature.

V. Conclusion and Future Scope

In this research article a technique has been implemented to evaluate the performance of DSA with RSA. In order to further improve the efficiency in terms of execution time, we have modified the existing signature scheme by incorporating Lightweight hash function. The proposed technique is validated by performing the comparative investigations with other existing techniques. The outcomes achieved validated that we have achieved the better performance than other techniques. Even though the research work accomplishes the primary objective of attaining the improved time efficiency, there is a scope of improvement in transaction of data integrity. Hence in future the proposed system can be extended to focus on the integrity of data by improving the Hash function that is suitable for the Digital signature scheme.

References

[1] A. Nist, “proposed federal information processing standard for digital signature standard (dss),” Federal Register, vol. 56, no. 1692, pp. 42980–42982, 1991.

[2] W. C. Cheng, C.-F. Chou, and L. Golubchik, “Performance of batch-based digital signa- tures,” in Proceedings. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, pp. 291–299, IEEE, 2002.

[3] S. Singh, M. S. Iqbal, and A. Jaiswal, “Survey on techniques developed using digital signa- ture: public key cryptography,” International Journal of Computer Applications, vol. 117, no. 16, 2015.

[4] P. Kitsos, N. Sklavos, and O. Koufopavlou, “An efficient implementation of the digital signature algorithm,” in 9th International Conference on Electronics, Circuits and Systems, vol. 3, pp. 1151–1154, IEEE, 2002.

[5] R. Kasodhan and N. Gupta, “A new approach of digital signature verification based on biogamal algorithm,” in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pp. 10–15, IEEE, 2019.

[6] A. Khalique, K. Singh, and S. Sood, “Implementation of elliptic curve digital signature algorithm,” International journal of computer applications, vol. 2, no. 2, pp. 21–27, 2010.

[7] R. Soram and E. S. Meitei, “On the performance of rsa in virtual banking,” in 2015 Inter- national Symposium on Advanced Computing and Communication (ISACC), pp. 352–359, IEEE, 2015.

[8] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, Handbook of applied cryptography. CRC press, 2018.

[9] G. Wang, Bibliography on Blind Signatures [Online]. Available: http://www.i2r.a-star.edu.sg/icsd/staff/ guilin/bible/blind-sign.htm [ONLINE], Available

[10] Amalarethinam, DI George, and H. M. Leena. "Enhanced RSA algorithm for data security in cloud." International Journal of Control Theory and Applications 9 (2016): 147-152.

[11] Sarthak R Patel, Khushbu Shah, “Security Enhancement and Speed Monitoring of RSA Algorithm”, “International Journal of Engineering Development and Research”, vol. 2, 2057-2063, 2014.

[12] Gupta, Shilpi, and Jaya Sharma. "A hybrid encryption algorithm based on RSA and Diffie-Hellman." 2012 IEEE International Conference on Computational Intelligence and Computing Research. IEEE, 2012.

[13] Iswari, N.M.S., "Key generation algorithm design combination of RSA and ElGamal algorithm." 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, 2016.

[14] Patidar, Ritu, and Rupali Bhartiya. "Modified RSA cryptosystem based on offline storage and prime number." 2013 IEEE International Conference on Computational Intelligence and Computing Research. IEEE, 2013.

[15] Mathur, Shikha, et al. "Analysis and design of enhanced RSA algorithm to improve the security." 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT). IEEE, 2017.

[16] Jaju, Sangita A., and Santosh S. Chowhan. "A Modified RSA algorithm to enhance security for digital signature." 2015 international conference and workshop on computing and communication (IEMCON). IEEE, 2015.

[17] Minni, R., Sultania, K., Mishra, S., and Vincent, D. R. "An algorithm to enhance security in RSA." 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). IEEE, 2013.

[18] Thangavel, M., Varalakshmi, P., Murrali, M., & Nithya, K. (2015). An enhanced and secured RSA key generation scheme (ESRKGS). Journal of information security and applications, 20, 3-10.

[19] Islam, M. A., Islam, M. A., Islam, N., & Shabnam, B. (2018). A modified and secured RSA public key cryptosystem based on “n” prime numbers. Journal of Computer and Communications, 6(03), 78.

Auto JSON: An Automatic Transformation Model for Converting Relational Database to Non-relational Documents

K. Revathi[1], T. Tamilselvi[2], Batini Dhanwanth[3], M. Dhivya[4]
Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India[1, 2, 4]
Department of Computer Science and Engineering, Panimalar Institute of Technology, Chennai, India[3]

Abstract—**In recent days, the demand for dealing large set of distributed data obsoletes the relational database and its structured query language (SQL) solutions in practice and paves the way for novel solutions in the name of non-relational database as not-only SQL (NoSQL). The NoSQL offers dynamic, flexible, scalable, highly available, greater performance and near real-time access to the distributed nature of voluminous data used for current industrial applications. Apart from these giant features of NoSQL, the SQL is still found to be in operation because of its popularity and standard. This paper projected an algorithm to convert the relational documents of MySQL into any document oriented NoSQL databases automatically without destructing the existing relational database setup and installing the NoSQL from scratch in the core machines. Java Script Object Notation (JSON) is a human readable data interchange format, being used in web development. The characteristics of JSON widened its use cases from web development to database storage. The Mongo DB, one of the most popular document oriented NoSQL adapts JSON format for its storage. The proposed algorithm is built based on its schema definition and the performance is captured through evaluating it against a sample database from hospital management system. The findings are discussed with great interest of addressing the challenges and revealing the scope for improvement.**

Keywords—Distributed data; document oriented NOSQL; hospital management system; Mongo DB; my SQL

I. Introduction

Databases Management System was created as a result of an increase in demand from industries for preserving their customer, stock, and account-related data and the process of obtaining relevant information from the data (DBMS). Systematic maintenance, as well as the effective storage and retrieval of data, are made possible by DBMS. Data are initially maintained using file systems, which store information directly in files without any connection to one another. Several models were presented as a result of the constraints in data access patterns that were found. International Business Machines (IBM) created a hierarchical model in 1960 that enables data organization in a tree structure by altering the parent-child relationship.

The network model, put forth by Charles Bachman in 1969, allows for the arrangement of data in a graph-like structure, with nodes serving as records and arcs as the connections between them. E. F. Codd created the relational model in 1970, which organizes data as tables and is currently regarded as a special model employed in important industries due of its features [1].

When dealing with situations in the real world, the data is contained in a single container referred to as an object. This begins the object-oriented database that has been in use since 1985 [2]. In order to create a new database known as an object relational database, the capabilities of relational databases and object-oriented databases were combined. By adding more dimensions to the data by displaying it as a cube, online analytical processing capability (OLAP) is offered in place of transaction processing.

Big data has replaced the traditional data that used to occur over a longer period of time in sectors that deal with data created every day. Even today's data volume is measured in petabytes or zettabytes, and 50% of the data are unstructured. The performance of the relational database, which exclusively processes structured data, is good for reasonable workloads but degrades as it is scaled up. Big data and analytical processing have made Relational Database Management Systems (RDBMS), which use Structured Query Language (SQL) to operate, ineffective [3]. Carlo Strozzi was the first to suggest Not Just SQL (NoSQL). This RDBMS was file-based and lacked a SQL interface. NoSQL, often known as non-relational databases, was first introduced in 2009 by Eric Evans [4, 5]. It is recognized as a promising database to handle massive data.

NoSQL is an alternative to RDBMS, facilitate mechanisms to store and retrieve enormous data in a distributed platform. The features on NoSQL over RDBMS are listed below.

- Horizontal Scalability – new nodes can be added to dynamic accommodate the storage requirements / requests
- Sharding – balances the workload distribution over the clusters in the distributed environment
- High Availability – due to its distributed nature, there is no single point of failure. Replication promises the high availability
- Better Throughput – offer better throughput to even high volume data than RDBMS
- Faster Performance – facilitate faster performance for big data than RDBMS

NoSQL is a popular language due to its enormous storage capacity as well as the following characteristics that set it apart from SQL.

- ACID Free: The acronym ACID stands for atomicity, consistency, isolation, and durability and supports the SQL transaction notion [6, 7]. NoSQL, a distributed database, provides improved data storage by relying on consistency but does not guarantee ACID properties.
- BASE: BASE stands for fundamentally available, soft state (data may vary over time), and finally consistency. It assures a high degree of availability through replication (no requirement to have identical copies in all nodes for all the time). In order to prioritize availability above consistency, eBay proposed this database design behavior, and NoSQL adopts it [6-9].
- CAP: Eric Brewer proposed the CAP theorem at a symposium on the fundamentals of distributed computing in 2000. It states that in network shared data systems, there is a trade-off between consistency, availability, and partition tolerance. BASE, which is a reverse notion of ACID and it is derived from the CAP theorem, is not feasible to hold up in distributed database, any two only achievable at time [6, 8] and depicted in Fig. 1.

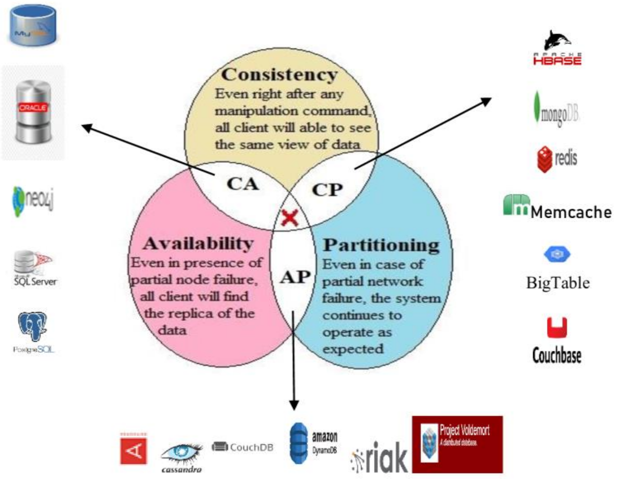

Consistency
Even right after any manipulation command, all client will able to see the same view of data
CA
CP
Availability
Even in presence of partial node failure, all client will find the replica of the data
AP
Partitioning
Even in case of partial network failure, the system continues to operate as expected
HBASE
mongoDB
redis
Memcache
BigTable
Couchbase
neo4j
SQL Server
PostgreSQL
ORACLE
cassandra
CouchDB
amazon DynamoDB
riak
Project Voldemort
A distributed database

Fig. 1. Visual representation of CAP theorem.

The rest of the paper is organized as follows: In Section II the significant past efforts made are analyzed and the basic detail about migration is discussed. The Section III discusses the migration algorithm in high interest. Section IV discloses the results obtained out of the device and Section V is intended to conclude the work.

II. Literature Review

A. Motivations for Migration Model

The latest and well-known initiatives to convert relational databases to NoSQL databases in order to address the emerging demand of modern applications have been thoroughly examined and are given in this section.

For the effective conversion of relational data into non-relational ones, the researchers offered a variety of methodologies, including data model-based, cloud-based, layer-based, web-based, and cross query engine based ones [10]. According to a research article by Liana Stanescu et al. [11], a base algorithmic view, the necessity of converting MySQL, a relational database, to a NoSQL database, nurturing the features of Mongo DB through addressing the fundamental principles to be followed in the transformation process, was elaborately captured.

Also, a framework was created using the NET platform, and an algorithm was created. The effectiveness of the algorithm was assessed based on the execution time of Create, Read, Update, and Delete (CRUD) operations over the databases provided with different workloads [12, 13]. The tuples are projected as documents, the columns are shown as fields in Mongo DB, and each MySQL database is represented as a collection. Using either embedding or referencing techniques, the key relationships in MySQL should be translated into Mongo DB.

Mahamood [14] described a method and created an interface using VB. Net that automatically converts Microsoft SQL Server tables to Mongo DB collections, in a manner comparable to the work mentioned with Liana Stanescu et al. [13].

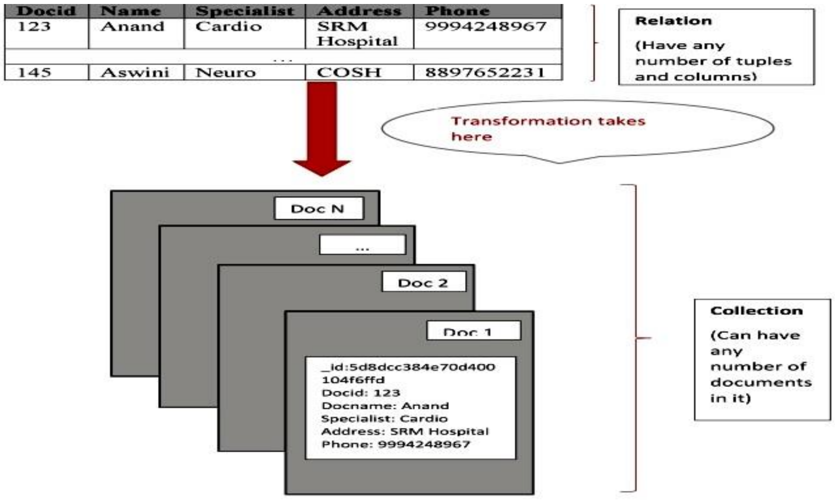
Researchers Gyorodi, Kumar, Krishnan and Nair also conducted performance evaluations of MySQL and Mongo DB [15-18]. The results of the experiment demonstrated Mongo DB's superior efficiency. Additionally, the effectiveness is confirmed using the Yahoo Cloud Server Benchmarking (YCSB) tool by Kumar and Chakraborttii [19-20]. Saber et al. [21] discussed the efficiency of Mongo DB for handling Internet of Things (IoT) data in comparison to relational databases.

Singh [22] created a data conversion method that converts relational databases into Mongo DB collections and was discovered to be a successful pattern for cloud storage. In addition to the data transformation paradigm, Bajwa et al. [23] suggested data cleaning methods. A query-based transformation module to move from relational to non-relational data was developed by Al-Mahruqi et al. [24] and is designed to be used with apps.

B. Elemental Facts on Migration Model

The tables in MySQL databases are transformed into a collection of documents like in Mongo DB throughout the migration process, which is illustrated in Fig. 2 as a general process flow.

The mentioned transformation by Liana Stanestcu et al. [11-13] uses the metadata information of relational data bases and influences the Entity Relationship (ER) model, maps the key relations (1:1, 1: N, and M: N) found in RDBMS that are framed using primary and foreign keys against the relationship models in Mongo DB as embedding and referenced or linking.



Docid	Name	Specialist	Address	Phone
123	Anand	Cardio	SRM Hospital	9994248967
...				
145	Aswini	Neuro	COSH	8897652231

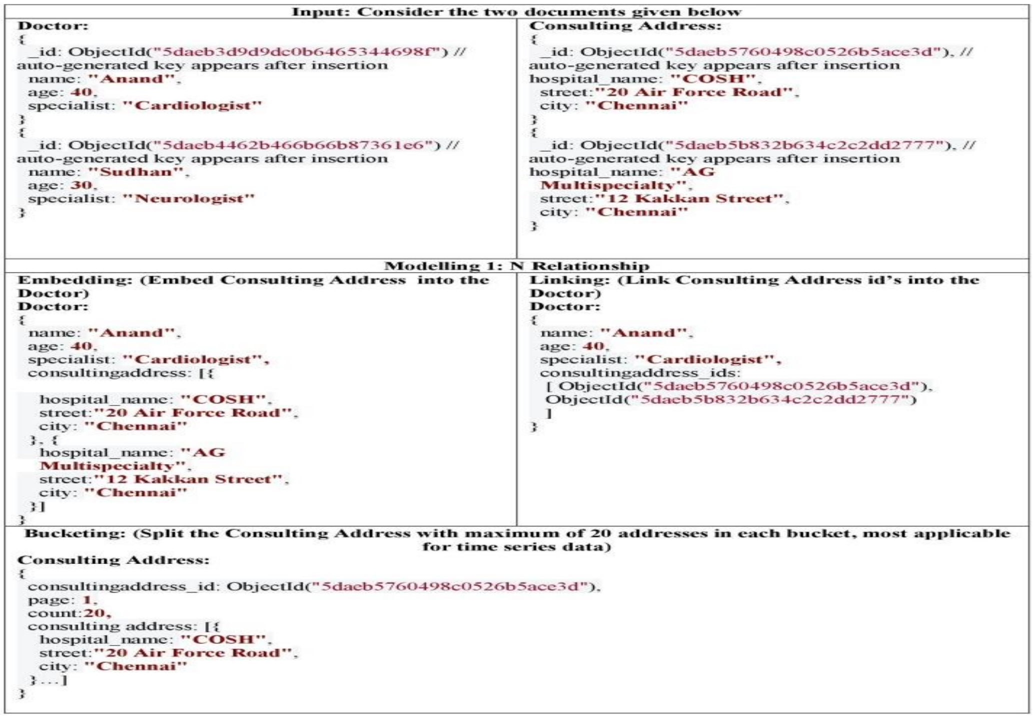
Relation
(Have any number of tuples and columns)
Transformation takes here
Doc N
...
Doc 2
Doc 1
_id:5d8dcc384e70d400 104f6ffd
Docid: 123
Docname: Anand
Specialist: Cardio
Address: SRM Hospital
Phone: 9994248967
Collection
(Can have any number of documents in it)

Fig. 2. The generic flow of migration process.

The following discusses in depth the various Mongo DB schema designs that might be used to reproduce the established RDBMS relationships and their recommended method of implementation [25, 26] by Gopinath et al. 2017, Yassine & Awad 2018.

In Mongo DB, there are only two modeling options for one-to-one (1:1) relationships: embedding or linking. When a document is embedded using arrays inside another document, all associated documents are shown as a single document. The embedding strategy increases data size, affects write performance, but makes retrieval simpler because it is combined with a single read. Although though documents are maintained separately, linking involves referencing one document's id through a field in another document using an automatically created key since relational databases store the data using foreign keys. In contrast to embedding, it shrinks the data yet affects read performance. Embedding is the best option because it offers effective retrieval.

There are three different techniques to model the one-to-many (1: N) relationship: embedding, linking, and bucketing. The first two still function. While bucketing is a third strategy that conforms to efficient retrieval while combining the advantages of embedding and connecting through slicing data into buckets with set data limits. Time series data applications benefit from bucketing. The methods for modeling the 1: N connection that was covered is depicted as a code snippet in Fig. 3.



Input: Consider the two documents given below

Doctor:
```
{
 _id: ObjectId("5daeb3d9d9dc0b6465344698f") //
auto-generated key appears after insertion
 name: "Anand",
 age: 40,
 specialist: "Cardiologist"
}
{
 _id: ObjectId("5daeb4462b466b66b87361e6") //
auto-generated key appears after insertion
 name: "Sudhan",
 age: 30,
 specialist: "Neurologist"
}
```

Consulting Address:
```
{
 _id: ObjectId("5daeb5760498c0526b5ace3d"), //
auto-generated key appears after insertion
hospital_name: "COSH",
 street:"20 Air Force Road",
 city: "Chennai"
}
{
 _id: ObjectId("5daeb5b832b634c2c2dd2777"), //
auto-generated key appears after insertion
hospital_name: "AG
 Multispecialty",
 street:"12 Kakkan Street",
 city: "Chennai"
}
```

Modelling 1: N Relationship

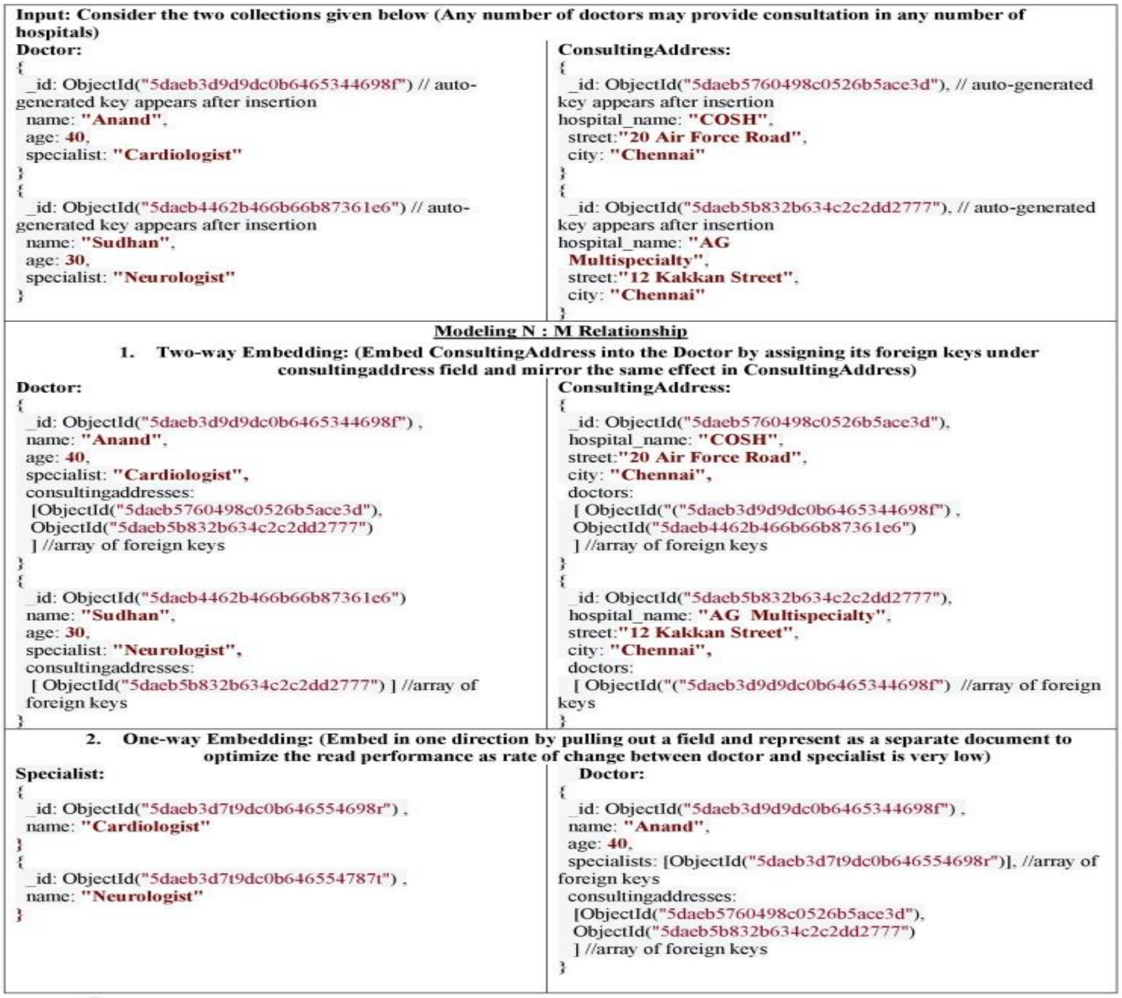
Embedding: (Embed Consulting Address into the Doctor)
Doctor:
```
{
 name: "Anand",
 age: 40,
 specialist: "Cardiologist",
 consultingaddress: [{

   hospital_name: "COSH",
   street:"20 Air Force Road",
   city: "Chennai"
  }, {
   hospital_name: "AG
   Multispecialty",
   street:"12 Kakkan Street",
   city: "Chennai"
  }]
}
```

Linking: (Link Consulting Address id's into the Doctor)
Doctor:
```
{
 name: "Anand",
 age: 40,
 specialist: "Cardiologist",
 consultingaddress_ids:
  [ ObjectId("5daeb5760498c0526b5ace3d"),
  ObjectId("5daeb5b832b634c2c2dd2777")
   ]
}
```

Bucketing: (Split the Consulting Address with maximum of 20 addresses in each bucket, most applicable for time series data)
Consulting Address:
```
{
 consultingaddress_id: ObjectId("5daeb5760498c0526b5ace3d"),
 page: 1,
 count:20,
 consulting address: [{
   hospital_name: "COSH",
   street:"20 Air Force Road",
   city: "Chennai"
  }...]
}
```

Fig. 3. Strategies to map 1: N relationship in Mongo DB.

The Fig. 4 illustrates two modeling approaches that are commonly used to represent many-to-many (M: N) relationships: two-way embedding and one-way embedding. In two-way embedding, one can insert one document into another by mirroring the foreign keys of the two documents in each field. When the size of the one-way embedding is extremely imbalanced, it is found to be useful as an optimal solution because it only enables embedding in one direction.



Input: Consider the two collections given below (Any number of doctors may provide consultation in any number of hospitals)

Doctor:

```
{
 _id: ObjectId("5daeb3d9d9dc0b6465344698f") // auto-
generated key appears after insertion
 name: "Anand",
 age: 40,
 specialist: "Cardiologist"
}
{
 _id: ObjectId("5daeb4462b466b66b87361e6") // auto-
generated key appears after insertion
 name: "Sudhan",
 age: 30,
 specialist: "Neurologist"
}
```

ConsultingAddress:

```
{
 _id: ObjectId("5daeb5760498c0526b5ace3d"), // auto-generated
key appears after insertion
hospital_name: "COSH",
 street:"20 Air Force Road",
 city: "Chennai"
}
{
 _id: ObjectId("5daeb5b832b634c2c2dd2777"), // auto-generated
key appears after insertion
hospital_name: "AG
 Multispecialty",
 street:"12 Kakkan Street",
 city: "Chennai"
}
```

Modeling N : M Relationship

1. Two-way Embedding: (Embed ConsultingAddress into the Doctor by assigning its foreign keys under consultingaddress field and mirror the same effect in ConsultingAddress)

Doctor:

```
{
 _id: ObjectId("5daeb3d9d9dc0b6465344698f") ,
 name: "Anand",
 age: 40,
 specialist: "Cardiologist",
 consultingaddresses:
  [ObjectId("5daeb5760498c0526b5ace3d"),
  ObjectId("5daeb5b832b634c2c2dd2777")
  ] //array of foreign keys
}
{
 _id: ObjectId("5daeb4462b466b66b87361e6")
 name: "Sudhan",
 age: 30,
 specialist: "Neurologist",
 consultingaddresses:
  [ ObjectId("5daeb5b832b634c2c2dd2777") ] //array of
 foreign keys
}
```

ConsultingAddress:

```
{
 _id: ObjectId("5daeb5760498c0526b5ace3d"),
 hospital_name: "COSH",
 street:"20 Air Force Road",
 city: "Chennai",
 doctors:
  [ ObjectId("("5daeb3d9d9dc0b6465344698f") ,
  ObjectId("5daeb4462b466b66b87361e6")
  ] //array of foreign keys
}
{
 _id: ObjectId("5daeb5b832b634c2c2dd2777"),
 hospital_name: "AG  Multispecialty",
 street:"12 Kakkan Street",
 city: "Chennai",
 doctors:
  [ ObjectId("("5daeb3d9d9dc0b6465344698f")  //array of foreign
keys
}
```

2. One-way Embedding: (Embed in one direction by pulling out a field and represent as a separate document to optimize the read performance as rate of change between doctor and specialist is very low)

Specialist:

```
{
 _id: ObjectId("5daeb3d7t9dc0b646554698r") ,
 name: "Cardiologist"
}
{
 _id: ObjectId("5daeb3d7t9dc0b646554787t") ,
 name: "Neurologist"
}
```

Doctor:

```
{
 _id: ObjectId("5daeb3d9d9dc0b6465344698f") ,
 name: "Anand",
 age: 40,
 specialists: [ObjectId("5daeb3d7t9dc0b646554698r")], //array of
foreign keys
 consultingaddresses:
  [ObjectId("5daeb5760498c0526b5ace3d"),
  ObjectId("5daeb5b832b634c2c2dd2777")
  ] //array of foreign keys
}
```

Fig. 4. Methodologies to map M: N relationship in Mongo DB.

C. Identified Research Gaps

The major efforts are for converting MySQL to Mongo DB was realized in [10-14, 22-26]. In most of the attempts, transformation mechanism was rooted from its schema design. The data type and key relationship exists between the tables plays an significant role in deciding the transformation strategy. Moreover the conversion is automated through an interface developed. In addition to key-relationship, table volume can be utilized in order to take precise decision about opting for embedding or linking. No existing work focused on time series data. By accounting current data type utilized in modern applications, the proposed work provided an automated database conversion solution for time series data.

III. AUTO JSON: A MIGRATION MODEL

The proposed work employs MySQL as a source database and selects Mongo DB as the destination NOSQL database based on inspiration drawn from related contributions listed in the preceding section. The proposed algorithm that automates the transformation of tables in source to collections in desired database is discussed in this section elaborating the data with the execution environment used for it.

A. Hospital Management System: A Source Database

A software program known as a hospital management system controls medical setup operations without the need of paper. The HMS incorporates all pertinent data, including those on doctors, patients, and related services. A hypothetical hospital administration scenario is used as the input database, and Fig. 5 depicts the appropriate relationships between the tables in the database.

Five tables make up the database, and connectors are used to create the relationship between the tables. Each class's primary key attribute is represented by the first entry in that class. It is beyond of scope to go into depth about class functions.

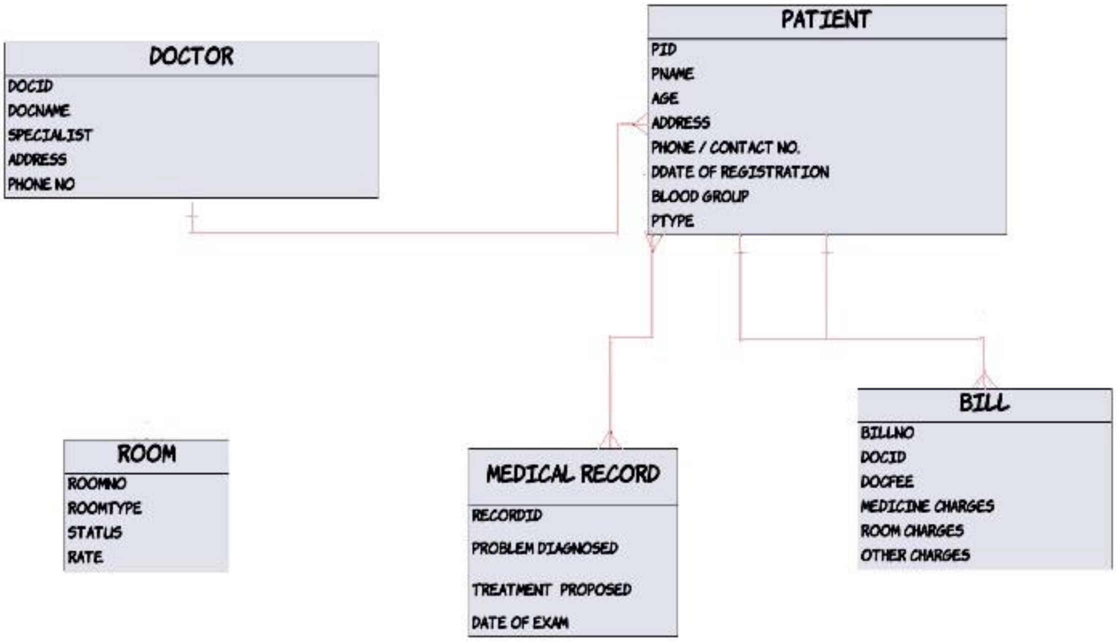


Fig. 5. A class diagram of typical HMS.

B. Migration Algorithm

Tables can be converted into collections using the suggested method's options for bucketing, embedding, linking, one-way embedding, and two-way embedding.

TABLE I. ALGORITHMIC STEPS FOR AUTO MIGRATION ALGORITHM

```
Algorithm 1: Auto JSON (SQL DB, Relationship _Mode,
Table_Volume, Data _Type)
Input: Relational Table (MySQL)
Output: JSON Documents (Mongo DB)
Extract meta data of  SQL DB
Observe the data pattern and relationship exists among the tables
  | if(Relationship_Mode == 1:1 && Table_Volume==High) then
  |     Perform Linking
  | else if (Relationship_Mode == 1: N && Data_type==
  |          Time_series) then
  |     Perform Bucketing
  | else if (Relationship_Mode == 1: N && Data_type!=
  |          Time_series) then
  |     Perform Linking
  | else if (Relationship_Mode == M: N  &&
  |          Table_Volume==High) then
  |     Perform Two Way Embedding
  | else if (Relationship_Mode == M: N  &&
  |          Table_Volume==Low) then
  |     Perform One Way Embedding
  | else
  |     Perform Embedding
Return the Collections in Mongo DB
```

The decision is significantly impacted by a number of factors, including table size, data type, and relationship method. These parameters are referenced about using meta-data that is given in the source database's information schema. Previous attempts focused on the sort of relation that already existed with keys between tables and left out any consideration for the data's nature or table volume. The proposed research optimizes the selection of transformation by taking these aspects into account, as shown in Table I.

The Extract, Transform, and Load (ETL) principle underlies the migration algorithm's operation. In order to convert tables into JSON collections, the meta-data of the table schema must first be extracted. Careful consideration of the schema information, such as field type, table volume, and relationship type, enables selection of options like linking, embedding, bucketing, and one-way or two-way embedding. Lastly, the collections are uploaded to a cloud storage system for further data analytics. Without installing the underlying database packages, the migration method automates the conversion of relational model tables to collections of non-relational model tables.

IV. RESULTS AND DISCUSSIONS

This section provides the details on the experimentation environment, evaluation methodologies and metrics to demonstrate the performance of the proposed migration strategy.

A. Experimetation Setup

The Auto JSON algorithm is coded as python program and executed on the Mongo DB ATLAS platform. Mongo DB ATLAS is a cloud database as a service (DBaaS) contributed by the Mongo DB. It has various provisions listed as follows.

- Provides all the features of Mongo DB
- Simplifies the automation process without overlooking the infrastructure, configuration of database, backups and so on.

- Ensures security and privacy.
- Facilitates choice of deploying the generated database in one of the platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP) or Microsoft Azure.

B. Dataset and Queries for Evaluation

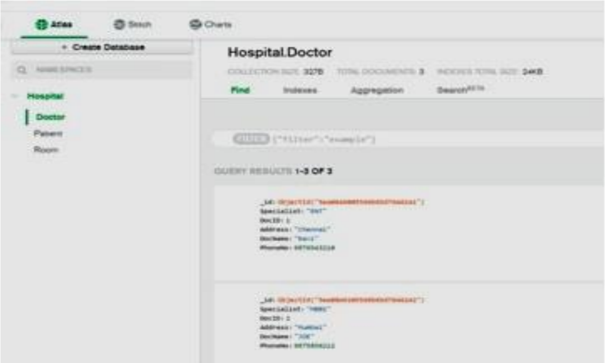
Evaluation is done using data that is in line with the hospital management system described in the preceding section. Compared to creation and read/retrieve, the scope for updating and deleting a record is minimal. The common users are restricted with the read-only and other security privileges. Hence, the following fundamental actions were picked to demonstrate the efficacy of the migration model that converts SQLDB to Mongo DB:

- Create with Insert
- Select

Among various DB functions, create function only allocates the space for data whereas the insertion command only populates the data into the space preserved by create. The delete command removes the entry from the space. Likewise the update command attempts to accommodate few changes on the data. The transformation task has its major focus on creating and presenting the same as JSON collections of Mongo DB. Here create function means a allocating a space provided with the data by means of insertion. Thus the comparative analysis of migration algorithm from MySQL and Mongo DB is captured by means of projecting the efficiency with respect to the basic operations named Create and Select.

The steps taken to record the performance of the suggested migration model is listed as follows:

- Register for a Mongo DB ATLAS user account.
- Put in the required libraries to assist migration.
  - pip install jsonmerge
  - pip install pymongo
  - pip install sqlalchemy
- Use Python to create the SQLDB code .
- Transform to Mongo DB Collection by running the migration algorithm in python as illustrated in Fig. 6.



Hospital.Doctor

COLLECTION SIZE: 327B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 24KB

{"filter":"example"}

QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("[illegible]")
Specialist: "ENT"
DocID: 1
Address: "Chennai"
DocName: "[illegible]"
PhoneNo: [illegible]
```

```
_id: ObjectId("[illegible]")
Specialist: "[illegible]"
DocID: 2
Address: "Mumbai"
DocName: "[illegible]"
PhoneNo: [illegible]
```

Fig. 6. Screenshot of completion of migration.

C. Performance Evaluation

The primary objective of the proposed work is to improve the efficiency in terms of execution time and memory utilization. In general execution or response time with respect to the database operations is calculated by accounting the time of observing the result set from the query initiation time. As aligned with this, the formula to evaluate response time is given in Eq. (1).

$$R_T = Q_C - Q_I \quad (1)$$

Where R_T is the response time, Q_C is for query completion time and Q_I is known as query initiation time. In this case, the query is simply initiated by selecting a transformation operation, and it is completed by collecting the appropriate result set as collections. Since the source database contains non-time series data, bucketing is no longer within the purview of this project. The Fig. 7 displays the execution time plot of transformation operations.

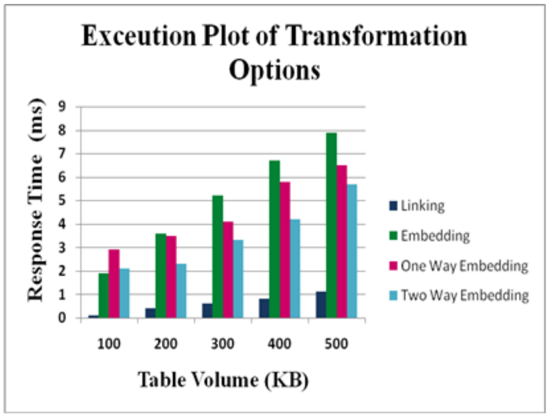


Fig. 7. Execution plot of transformation operations.

The next important metrics is demonstrating the effective memory utilization of the proposed algorithm. This is derived by invoking a comparison over consumption of memory against source and destination database. The effective memory utilization of proposed algorithm is formulated in Eq. (2) and pictorial representation is illustrated in Fig. 8.

$$MEU = (MC / MR) * 100 \quad (2)$$

Where M_{EU} stands for effective memory utilization, M_C denotes memory consumptions observed in Mongo DB collections and M_R gives memory consumptions observed in MySQL relations.

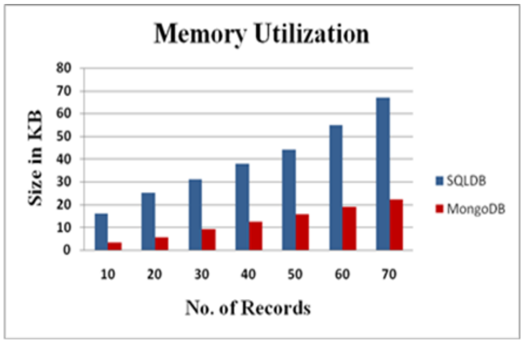


Fig. 8. Comparative analysis of memory usage.

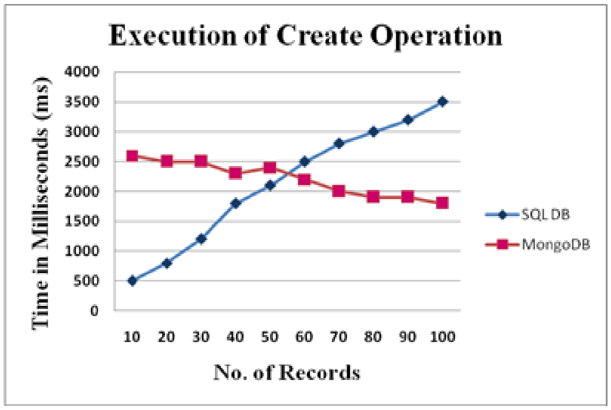


Fig. 9. Evaluated result of create operation.

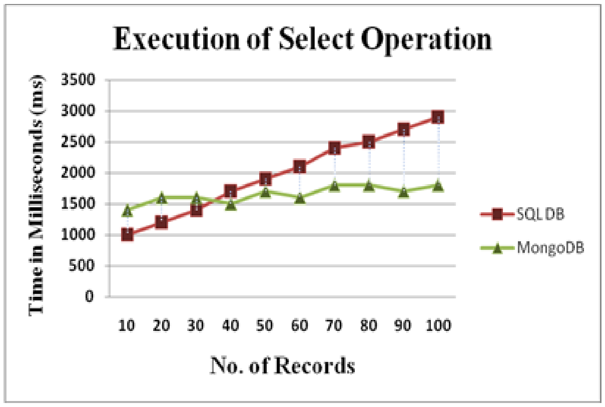


Fig. 10. Assessment map of select operation.

With respect to memory usage, Mongo DB preserves approximately 30% of memory as average when compared to MySQL. The efficiency mappings in association with basic queries into consideration are portrayed as Fig. 9 and 10, respectively.

V. Conclusion

It comes as no surprise that cloud and data storage provides enormous data storage that is compatible with ZB as well as mobile phones that provide storage in addition to 256 gigabytes (GB). A reliable, available, intact, quick, and secure database is necessary to extract intelligent information from this vast amount of data. The ideal substitute for handling massive data in multiple phases, such as transmission, storage, and analysis, is Mongo DB. Based on its schema information, an effort is performed here to convert the columnar SQL data to Mongo DB collections. The earlier methods mainly focused on relationship mode obtained by key and ignored the data volume and nature. Table type and size play important roles in selecting an appropriate transformation choice in this method.

The developed migration mechanism is tested using Python programming in the Mongo DB ATLAS environment. It captures the responsiveness in terms of transformation and query execution efficiency. A faster reaction is made possible by linking, which is 5.9% faster than two-way embedding, 7.6% faster than one-way embedding, and 8.4% faster than embedding. As comparison to the average relational table, the create operation performs 10% faster, and the select operation executes 13% faster. With relational databases taken into account, the memory utilization ratio of the migration procedure is estimated to be 30% on average.

In future the effectiveness of proposed migration model can be evaluated with complex queries including Update and Delete involving time-series data.

Authors' Contribution

Author 1 implemented the concept and drafted the article with assistance of authors 3 and 4, respectively. The author 2 reviewed the article.

Conflict of Interest

The authors declare that have no competing interest.

References

[1] S. Praveen, U. Chandra and A. A. Wani, "A Literature Review on Evolving Database", International Journal of Computer Applications, vol. 162, no. 9, pp. 35-41, 2017.

[2] H. Alzahrani,, "Evolution of Object-Oriented Database Systems", Global Journal of Computer Science and Technology, vol. 16, no. 3, pp. 33-36, 2016.

[3] M. A. Ali, M. R. Ahmed, M. A. Khatun and K. Sundaraj, "A literature review on NoSQL database for big data processing", International Journal of Engineering & Technology, vol. 7, no. 2, pp. 902-906, 2018.

[4] J. R. Lourenco, B. Cabral, P. Carrerio, M. Vieira and J. Bernardino, "Choosing the right NoSQL database for the job: a quality attribute evaluation", Journal of Big Data, vol. 2, no. 18, pp. 1-26, 2015.

[5] Priyanka and Amit Pal, "A Review of NoSQL Databases, Types and Comparison with Relational Database", International Journal of Engineering Science and Computing, vol. 6, no. 5, pp. 4963-4966, 2016.

[6] V. Sharma and M. Dave, "SQL and NoSQL Databases", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 8, pp. 20-27, 2012.

[7] R. T. Mason, "NoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Database", In Proc. of the Informing Science & IT Education Conference, pp. 259-268, 2015.

[8] D. G. Chandra, "BASE analysis of NoSQL database", Future Generation Computer Systems, vol. 52, pp. 13-21, 2015.

[9] K. Sahatqija, J. Ajdari, X. Zenuni, B. Raufi and F. Ismaili, "Comparison between relational and NOSQL databases", In Proc. of the International Convention MIPRO, pp. 216-221, 2018.

[10] S. Ghotiya, J. Mandal and S. Kandasamy, "Migration from relational to NoSQL database", In Proc. of IOP Conf. Series: Materials Science and Engineering, vol. 263, pp. 1-8, 2017.

[11] L. Stanescu, M. Brezovan and D. D. Burdescu, "An Algorithm for Mapping the Relational Databases To MongoDB – A Case Study", International Journal of Computer Science and Applications, vol. 14, no. 1, pp. 65-79, 2017.

[12] L. Stanescu, M. Brezovan and D. D. Burdescu, "Automatic Mapping of MySQL Databases to NoSQL MongoDB", In Proc. of the Federated Conference on Computer Science and Information Systems, vol. 8, pp. 837–840, 2016.

[13] L. Stanescu, M. Brezovan, C. A. Spahiu and D. D. Burdescu, "'A Framework for Mapping the MySQL Databases To MongoDB – Algorithm, Implementation and Experiments", International Journal of Computer Science and Applications, vol. 15, no. 1, pp. 65-82, 2018.

[14] A. A. Mahmood, "Automated Algorithm for Data Migration from Relational to NoSQL Databases", Al-Nahrain Journal for Engineering Sciences, vol. 21, no. 1, pp. 60-65, 2018.

[15] C. Gyorodi, R. Gyorodi, G. Pecherle and A. Olah, "A Comparative Study: MongoDB vs. MySQL", In Proc. of 13th International Conference on Engineering of Modern Electric Systems, pp. 1-6, 2015.

[16] L. Kumar, S. Rajawat and K. Joshi, “Comparative analysis of NoSQL (MongoDB) with MySQL Database”, International Journal of Modern Trends in Engineering and Research, vol. 2, no. 5, pp. 120-127, 2015.

[17] A. Krishan and R. Wagh, “A Study of Performance NoSQL Databases”, International Journal of Innovative Research in Advanced Engineering, vol. 4, no. 4, pp. 32-36, 2017.

[18] S. M. Nair, R. Roy and S. M. Varghese, “Performance Evaluation of MongoDB and CouchDB Databases”, International Journal of Scientific Research in Science and Technology, vol. 3, no. 7, pp. 21-24, 2017.

[19] R. S. Kumar and R. R. Mary, “Comparative Performance Analysis of various NoSQL Databases: MongoDB, Cassandra and HBase on Yahoo Cloud Server”, Imperial Journal of Interdisciplinary Research, vol. 3, no. 4, pp. 265-269, 2017.

[20] C. Chakraborttii, “Performance Evaluation of NoSQL Systems Using Yahoo Cloud Serving Benchmarking Tool”, In Proc. of UCSC Research Symposium, pp. 1-9, 2015.

[21] W. Saber, M. M. Eyada, M. M., El Genidy and F. Amer, “Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environment”, IEEE Access, vol. 8, pp. 110656-110668, 2020.

[22] A. Singh, “Data Migration from Relational Database to MongoDB”, Global Journal of Computer Science and Technology, vol. 19, no. 2, pp. 17-21, 2019.

[23] I. S. Bajwa, S. Ramzan, B. Ramzan and W. Anwar, “Intelligent Data Engineering for Migration to NoSQL Based Secure Environments”, IEEE Access, vol. 7, pp. 69042-69057, 2019.

[24] R. S. Al-Mahruqi, M. H. Alalf and T. R. Dean, “A Semi-automated Framework for Migrating Web applications from SQL to Document Oriented NoSQL Database”, In Proc. of the 29th Annual International Conference on Computer Science and Software Engineering, pp. 1-10, 2019.

[25] M. P. Gopinath, G. S. Tamilzharasi, S. L. Aarthy and R. Mohanasundram, “An Analysis and Performance Evaluation of NOSQL Databases for Efficient Data Management in E-Health Clouds”, International Journal of Pure and Applied Mathematics, vol. 117, no. 21, pp. 177-197, 2017.

[26] F. Yassine and M. A. Awad, “Migrating from SQL to NOSQL Database: Practices and Analysis”, In Proc. of 13th International Conference on Innovations in Information Technology, pp. 58-62, 2018.

An Energy-Aware Technique for Task Allocation in the Internet of Things using Krill Herd Algorithm

Dejun Miao[1], Rongyan Xu[2], Jiusong Chen[3], Yizong Dai[4]
School of Electrical and Automotive Engineering, Yangzhou Polytechnic College
Yangzhou, Jiangsu, 225009, P.R. China[1, 3, 4]
School of Tourism, Yangzhou Polytechnic College Yangzhou, Jiangsu, 225009, P.R.China[2]

***Abstract*—The Internet of Things (IoT) is an innovative technology that connects the digital and physical worlds as well as allows physical devices capable of different capacities to share resources to accomplish tasks. Most IoT objects have limited battery life and are heterogeneous. Assignment of these objects is, therefore, extremely challenging. Energy consumption and reliability are the primary objectives of task allocation algorithms. We present an optimization solution to the IoT task allocation problem based on the krill herd algorithm. The algorithm increases the energy efficiency and stability of the network while providing a reliable task allocation solution. An extensive test of the proposed algorithm has been conducted using the MATLAB simulator. Compared to the most relevant method in the literature, our algorithm provides a higher level of energy efficiency.**

Keywords—Internet of things; resource allocation; task scheduling; energy efficiency

I. INTRODUCTION

In recent years, the rapid development of wireless and emerging technologies, particularly the Internet of Things (IoT), Blockchain [1], 5G connectivity [2], cloud computing, smart grids [3], optical networks [4, 5], machine learning [6], and artificial intelligence [7, 8] has provided many advantages. The IoT has recently been praised as a major technological advancement. As the name implies, it is a system specifically designed to connect computers, electronic and mechanical devices, and items equipped with unique identifiers capable of transmitting data over computer-to-human or human-to-human relationships [9]. Its primary goal is to mobilize network components for active cooperation and to make their resources available for certain applications. Developing the IoT sample requires the coordination and execution of smart items and the ability to interoperate significantly [10]. The concept of interoperability describes how nodes collaborate in a collaborative application using resources, including storage, processing, electrical energy, and object potential, to accomplish a task [11]. A crucial issue in IoT is task allocation, in which tasks can be executed within certain boundaries [12]. The task assignment aims to increase network performance [13]. IoT presents many challenges due to its heterogeneity, scale, properties, continuous processing, and shared sense [14]. The task allocation process cannot be completed in a centralized manner, with frequent scenarios being adjusted [15]. The IoT consists of ubiquitous networking involving many nodes spread across a wide area [16]. The quick changes in the network make it impossible for nodes to determine network topology. Therefore, integrating nodes and reconfiguring them are challenging. IoT task allocation is a common NP-hard problem. Solving this problem requires a meta-heuristic algorithm [17].

Customer service quality is adversely affected by IoT task allocation. Performance and quality of responses will be greatly impacted by how these tasks are performed, or in other words, how resources are allocated to requests. Based on evolutionary algorithms, various methods for allocating tasks in the IoT exist. Among the best ways to improve IoT device performance is to implement better resource allocation algorithms, which would decrease operational costs and reduce communication delays. According to [18], a better task allocation solution can lead to IoT devices providing users or information systems with enhanced services. Finding a final solution to an NP-hard problem takes substantial computation time. By employing an intelligent guessing strategy, meta-heuristic algorithms can quickly find optimal solutions to task allocation. Meta-heuristic algorithms are much faster than algorithms that evaluate all possible solutions sequentially. Meta-heuristic algorithms usually produce better results than greedy algorithms.

IoT paradigm realization requires the development of cooperative intelligent objects with key interoperability capabilities. Among these interoperability characteristics is the coordination among nodes in order to develop and deploy applications collaboratively, taking into account the limited resources available for a given task, such as electrical energy, memory, processing, and object capability [19]. IoT task allocation is crucial in finding a perfect match between scheduled tasks of an application and Edge-based devices to ensure instant response and efficient utilization of resources. In order to avoid problems concerning energy consumption and response time, we need a protocol for optimizing the process of allocating processing devices to tasks. This problem is considered an NP-hard problem.

In recent years, some studies have solved the task allocation problem using different meta-heuristic algorithms [20, 21]. Although meta-heuristic algorithms usually produce superior results than conventional methods, many areas remain to be improved. Our primary objective is to propose an improved meta-heuristic algorithm for solving the IoT task allocation issue. Our contributions include the following:

- When searching for the best nodes to execute tasks, two criteria are considered: the optimal path length (the number of hops between the first and last node of a task execution) and the capabilities of nodes correspond or near to the task capabilities.

- The krill herd algorithm is used to verify both parameters and determine the best path nodes.
- Assign task allocation capabilities to the nodes on the best path.

The remaining sections of the paper are organized in the following manner. The Section II outlines IoT challenges and reviews previous IoT task allocation methods. A detailed description of the proposed method appears in Section III. The results of the experiments are presented in Section IV. The paper is concluded in Section V.

II. Background

A. IoT Challenges

IoT presents many technical and management challenges, despite its ability to create a billion-device network. Most of these challenges can be summed up as follows.

- Security and privacy: As the IoT evolves new security and privacy challenges arise due to applications implemented and objects communicating over lossy networks. Due to IoT devices' limited power and processing capacity, it is not easy to apply encryption algorithms [22].
- Management: In IoT, devices must be deployed, provisioned, configured, controlled, monitored, diagnosed, updated, and regularly maintained [23].
- Availability: As new devices, services, and connected things emerge, accessing the desired node presents a challenge [24].
- Scalability: The architecture of an IoT network can change at any time as new nodes are added. The scalability of nodes needs to be verified by forming a global architecture. The solution should address different aspects of the problem [25].
- Mobility: Device mobility in IoT brings challenges, and IoT must ensure continuous global connectivity without disrupting communications [26].
- Reliability: IoT platforms, applications, and systems must be fully operational, aggregate data from multiple sources, and rely on reliable communication protocols. Constraint nodes should be able to respond quickly to an emergency at any time [27].
- Interoperability: As IoT devices have a wide range of platforms, protocols, and systems, the challenge is interoperating these heterogeneous platforms, protocols, and systems. Interoperability refers to the ability of nodes to exchange information [28].
- Low power energy communication: In IoT devices, low power consumption leads to long life cycles, low leakage currents, low charge-discharge rates, and quick startup times. Utilizing other energy resources or using lightweight algorithms can reduce overall power consumption [29].

B. IoT Task Allocation

Colistra, Pilloni [30] discussed the resource allocation issue for distributed IoT applications and middleware architectures that could provide solutions. Additionally, a consensus protocol was proposed for distributing the execution load among network objects to allocate resources efficiently. A homogeneous allocation of resources among nodes is demonstrated using the proposed protocol. Compared to the optimal allocation obtained with a centralized method, the algorithm converges with about a 6% error based on simulations and real-world experiments.

Khalil, Ozdemir [31] formulated the task allocation problem as an optimization problem with a single objective of reducing energy consumption. They proposed two approaches based on different objectives. These protocols are then injected with heterogeneity-aware heuristics that account for the different energy levels of network entities. To maximize the stability periods, they developed two more protocols, which reduce the interval before each virtual object dies. Different measurements and benchmarking parameters are used to evaluate the proposed protocols, and they are compared with the most relevant algorithms available. Regarding stability and energy consumption, the proposed protocols outperform existing work in the literature.

Low-latency IoT services are provided by fog-aided IoT networks, which offload computationally intensive, time-sensitive tasks to fog nodes located in the local area. IoT networks feature enhanced flexibility thanks to battery-limited mobile IoT devices. The QoS of mobile IoT may be degraded due to varying wireless channel conditions. A task allocation algorithm analyzed by Yao and Ansari [32] reduces average task completion latency by distributing tasks to different fog nodes and adapting to different mobile environments constrained by QoS requirements and the power of mobile IoT devices. Integer linear programming is then used to formulate the problem. Collecting mobility patterns and information from the user takes much work. A task allocation algorithm based on online learning was proposed and demonstrated through extensive simulations.

Sensitive delays, efficient resource allocation, and reliable transactions challenge IoT. To address these issues, Xiao, Gao [33] Introduced a blockchain-based IoT architecture that ensures the integrity of resource transactions and service providers' profits. In addition, they proposed the Task Offloading and Resource Allocation (TO-RA) algorithm, which is implemented as smart contracts on the blockchain. Based on blockchain's advantages, the proposed architecture optimizes resource allocation in IoT. To complete task offloading, they proposed a subtask-virtual machine mapping strategy. Then, based on the task offloading strategy, stack cache supplementation was proposed to complete resource allocation and address possible load imbalances in the system. The TO-RA algorithm outperforms conventional approaches regarding efficiency, reliability, and consistency.

The computational model developed by Mebrek and Yassine [34] considers energy consumption and transmission latency when offloading tasks in IoT applications. The competition was modeled as a game in which IoT devices decide how to distribute their tasks based on energy consumption and latency. The authors propose a decentralized task distribution algorithm in which players update their strategies based on the actions of other players. Nash equilibrium is proven to be achieved by the proposed algorithm. The computational model is then evaluated extensively and compared with existing studies.

Najafizadeh, Salajegheh [35] proposed a multi-objective simulated annealing algorithm to secure task allocation on cloud and fog nodes. A goal-programming approach is applied in order to discover a compromise solution. Additionally, a new goal referred to as "client-driven access level and schedule" is created regarding the distribution of tasks between fog and cloud nodes. Compared to moth-flame optimization, tabu search, and PSO algorithms, the proposed algorithm showed improved efficiency regarding deadlines by 50%, accuracy by 88%, and service delivery by 10%.

Hussain and Begh [36] implemented Hybrid Flamingo Search with a Genetic Algorithm (HFSGA) to optimize task scheduling for cost minimization. On seven essential benchmark optimization functions, HFSGA is compared with other well-known optimization algorithms. Furthermore, Friedman Rank Tests are conducted to ascertain the results' significance. It produces better results in terms of task completion percentages, makespan, and costs when implemented. As compared to existing algorithms such as round-robin, genetic, PSO, and ACO, this work shows better results.

Weikert, Steup [37] proposed a multi-objective task assignment method in dynamic IoT networks prone to mobility and failure issues. They combined and improved upon task allocation optimization algorithms for error-prone and mobile networks by predicting future node positions and archiving data to enhance diversity. An elitism mechanism preserves satisfactory outcomes. Consequently, this algorithm allocates IoT tasks despite node failures and considers future node positions simultaneously. Various error rates and mobile node counts are used in the evaluation. The investigation revealed enhanced network lifetime, latency, and availability metrics.

III. Proposed Method

This section proposes an algorithm for solving the problem of task allocation. Two parameters determine the assignment of tasks: the ability to execute subtasks by a given node and the path distance required to execute those subtasks. The issue declaration is defined in the first subsection, and the proposed method is described in detail in the second subsection.

A. Problem Statement

G = (V, E) represents task allocation as a graph, where nodes denote tasks, while edges indicate routes between tasks. A user's applications are divided into numerous tasks with sequential maintenance connections, and tasks are attached using a directed acyclic graph (DAG). The edges are all of the same form (i→j) and i, j∈N. Nodes represent tasks, and edges represent communications between them. *J* cannot be executed until the implementation of i is finished, and once all of its parents are complete, j becomes ready. In an edge from i to j, i is the parent node of j, and j is its child. Each DAG node has a weight corresponding to its computation cost, and each edge has a weight corresponding to its connection cost. An example of a DAG is shown in Fig. 1.

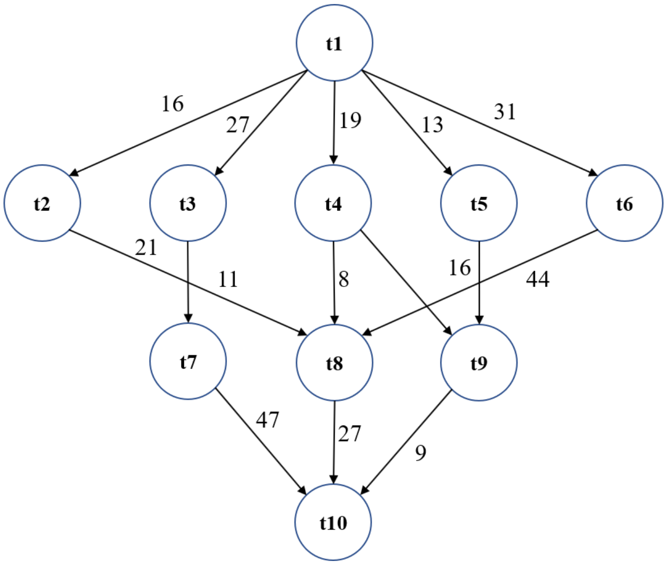


Fig. 1. A simplified DAG for IoT task allocation.

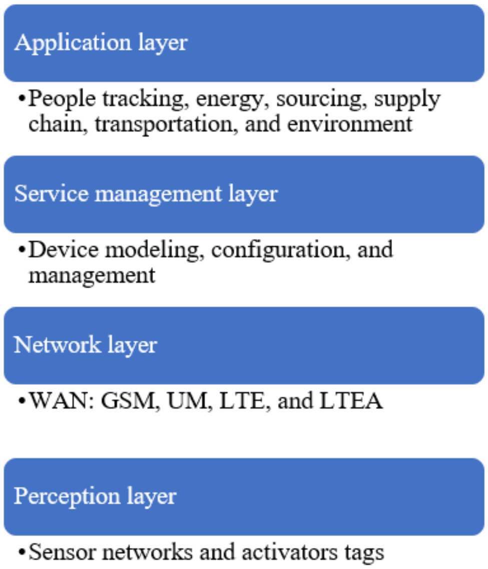

Application layer
•People tracking, energy, sourcing, supply chain, transportation, and environment
Service management layer
•Device modeling, configuration, and management
Network layer
•WAN: GSM, UM, LTE, and LTEA
Perception layer
•Sensor networks and activators tags

Fig. 2. The global IoT architecture.

B. Network Model

As shown in Fig. 2, the adopted IoT architecture for this study includes four layers: perception, network, service management, and application. Low-level tags, such as RFID and bar codes, make up the perception layer. In this layer, sensors and smart devices collect and process information in real time. In the network layer, several technologies are used to provide the necessary networking features. Sensors and smart devices create a high volume of Internet data objects that the embedded features in this layer should support. In the service management layer, information is analyzed, security is controlled, processes are modeled, and devices are managed. Sensor data can be alternated or non-intermittently sent. Task allocation is also part of its responsibilities. The application layer classifies programs based on real-time requirements, business models, convergence, and network type.

C. Krill Herd Algorithm for IoT Task Allocation

The krill herd algorithm solves continuous optimization problems using swarm intelligence. According to some studies, it is more efficient than some existing algorithmic approaches. It is simple to implement, robust, and only requires one control parameter (time interval) compared to other swarm-intelligence methods. The krill herd algorithm searches for food by traversing a multidimensional search space, where krills traverse various variables to make decisions. The distance between rich food and krills corresponds to the cost function. Three operational processes determine the location of a krill individual over time, including motion-induced processes, foraging movements, and random physical diffusion. Krill movements are visualized based on two distance measurements: the minimum distance between a krill and food sources and the least distance between each krill and the biggest herd. An n-dimensional decision space can be expressed in Eq. (1), in which *i*=1 to *nk* represents the number of krills, *Di* represents indiscriminate diffusion, *Fi* represents foraging activity, and *Ni* represents movement caused by other krill individuals. Krill movement can be expressed as Eq. (2). The variables in Eq. (2) are defined in Table I.

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{1}$$

$$N_i^{new} = \begin{bmatrix} N^{max}\left\{\sum_{j=1}^{NN}\left[\frac{K_i - K_j}{K_{worst} - K_{best}}\right]\left[\frac{X_j - X_i}{X_j - X_i + \varepsilon}\right]\right\} \\ \left\{2\left(rand + \frac{I}{I_{max}}\right)\overline{K_1, bestX_1, best}\right\} \end{bmatrix} \tag{2}$$

The forager's enjoyment of the food area determines the foraging movement using Eq. (3). Also, Eq. (4) is used to calculate the equation for sensing distances. In these equations, F_i^{old} refers to the last foraging motion, ω_f is inertia weight ranging between 0 and 1, V_f is foraging speed, and N is movement induced by every krill. Krill physical diffusion can be calculated by Eq. (5) as a random process, in which δ is a random directional vector [−1 &1], and D^{max} is the maximum diffusion speed.

$$F_i = V_f\left\{2\left(1 - \frac{I}{I_{max}}\right)\overline{K_1, foodX_1, food} + \overline{K_1, bestX_1, best}\right\} + \omega f F_i^{old} \tag{3}$$

$$d_{s,i} = \frac{1}{5N}\sum_{j=1}^{N}\left\|X_i - X_j\right\| \tag{4}$$

$$D_i = D^{max}\left(1 - \frac{I}{I_{max}}\right)\delta \tag{5}$$

Eq. (6) calculates the position vector, where *UB* and *LB* are upper and lower bounds, C_t is constant between 0 and 2, and *NV* is the variable amount.

$$X_i(t + \Delta t) = X_i(t) + \Delta t\frac{dx_i}{dt} \tag{6}$$

Good results will be obtained when the estimation vector values appear to be optimal. Eq. (7) and (8) provide genetic and mutation operators.

$$X_{i,m} = \begin{cases} X_{r,m} & rand_{i,m} < C_r \\ X_{i,m} & else \end{cases} \tag{7}$$

$$C_r = 0.2\overline{K_{1,best}} \quad r \in \{1,2,\dots,i-1,i+1,\dots,N\}$$

$$X_{i,m}\begin{cases} X_{gbest,m} + \mu\left(X_{p,m} - X_{q,m}\right) & rand_{i,m} < Mu \\ X_{i,m} & else \quad Mu = 0.05/\overline{K_{1,best}} \end{cases} \tag{8}$$

$$p, q \in \{1,2,\dots,i-1,i+1,\dots,K\}\ and\ \mu\ is\ between\ 0\ and\ 1$$

The krill herd algorithm solves the IoT task allocation problem by mapping the solution to the location of the krill group. Assuming T = (T_1, T_2, ..., T_m) and R = (R_1, R_2, ..., R_n), allocation results can be described by matrix x = [x_{ij}], i, j = (1, 2, ..., n), when T_i is allocated on R_j, xij = 1. Alternatively, if the x_{ij} value is 0, they can be used to determine a krill group's position. It simulates induced movement and feeding to determine the optimal location of krill groups so that IoT tasks can be allocated optimally. IoT task allocation using the krill herd algorithm involves the following steps:

- Set parameters for krill herd, such as the maximum number of iterations and the number of krill groups.
- The initial krill group position is generated based on the location of IoT resources. Each position of the krill corresponds to a scheme for allocating IoT resources.
- Obtaining the fitness function for each krill and evaluating the merits and demerits of each krill.
- The speed and position of krills are updated randomly to generate a new population of krills.
- If the termination condition of the algorithm is not satisfied, then proceed to step (4).
- The optimal allocation of IoT tasks is determined by the optimal location of the krill population.

IV. Experimental Results

In this section, a comparative analysis of the suggested task allocation method is presented based on the application DAG as well as connection costs (Fig. 1). The use of MATLAB is attributed to its ability to calculate matrix and calculation functions, perform practical functions, and exchange data with different programming languages. It focuses on computational methods and provides high-level programming capabilities. Mathematical issues in joint academic mathematics are presented in MATLAB. Therefore, MATLAB R2012a is used to simulate the proposed method.

We used the dataset from [38] as a basis for this study. A task allocation policy in the IoT directly impacts application performance and the best use of resources. Identifying the optimal way to assign tasks in the IoT is a significant challenge. The procedure for assigning tasks in the IoT environment is as follows. At first, tasks and resources are mapped based on this task and resource data. Consequently, by mapping resources among tasks, applicators can ensure task yield and QoS. In the end, the submitting user receives an overview of the consequences.

The convergence diagram for the method is shown in Fig. 3. Results show that about 180 generations of the proposed algorithm led to the correct answer. To prove that the proposed algorithm meets the repeatability criteria, it was repeated 20 times, 40 times, 60 times, 80 times, and 100 times. The stability diagram of the proposed algorithm is shown in Fig. 4. It is clear that successive iterations do not produce significantly different results. Consequently, the proposed algorithm is repeatable and robust.

Energy consumption is directly affected by IoT task allocation. Data centers in IoT structures can reduce their energy consumption by moving low- or idle-load servers to other servers, shutting them down, and shifting idle workloads to other servers. Fig. 5 shows the energy consumption of different algorithms for different task counts. According to the results, the suggested algorithm offers greater energy efficiency than others by increasing the number of tasks.

TABLE I. VARIABLES IN EQ. (2)

Variable	Definition
rand	A random number from 0 to 1
N_i^{old}	Final inertial weight
ω_n	Inertial weight in the range of (0,1)
$x_{1,best}$	The corresponding position of $x_{1,best}$ of i^{th} krill
$K_{1,best}$	Finest fitness rate of i^{th} krill
I_{max}	Maximum iteration
N_{max}	Large induced speed in ms^{-1}
ε	Minimum positive number
X	Exact location
K_{best}, K_{worst}	Best case and the worst case of an individual krill
K_j	Value of fitness of the acquaintance (j=1 to NN)

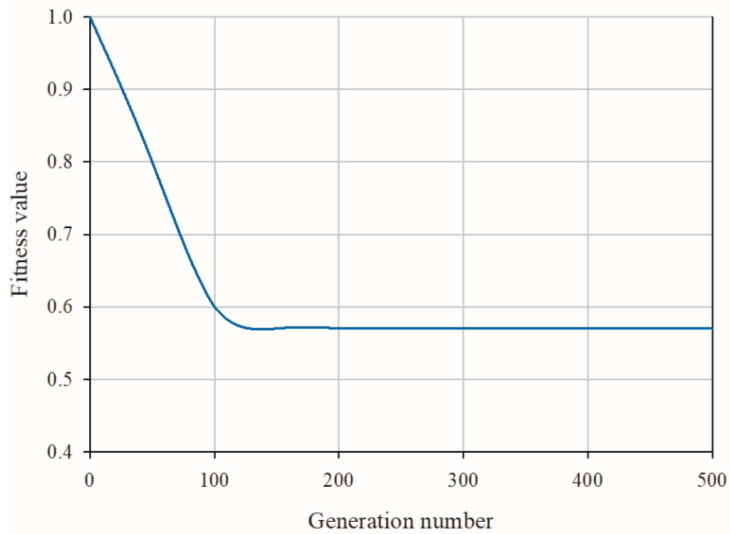


Fig. 3. Convergence diagram of the proposed method.

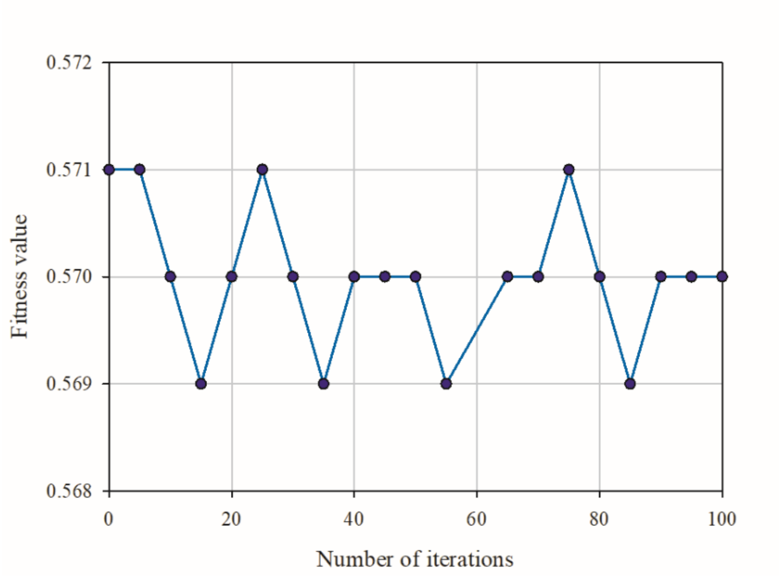


Fig. 4. Stability diagram of the proposed method.

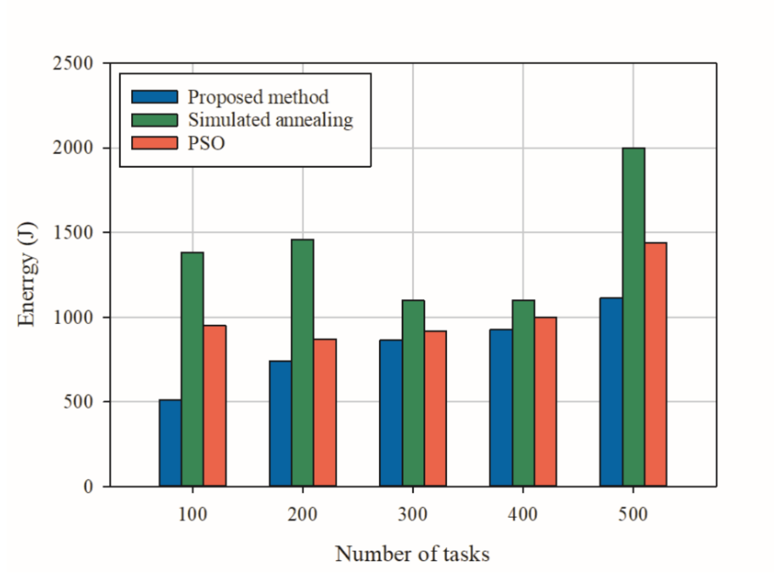


Fig. 5. Energy consumption comparison.

V. Conclusion

The primary objective of most IoT networks is to enable devices with varying capabilities to share their resources and cooperate in order to accomplish a desired function. The majority of IoT objects are heterogeneous in nature and have limited computational and energy capabilities. Consequently, assigning tasks to these objects presents a significant challenge. Most of the existing works in the literature use heuristic optimization techniques to address different aspects of the task allocation problem without taking into account the heterogeneity of the objects or the limits of their resources. This paper aimed to demonstrate the feasibility of developing meta-heuristic algorithms to assign and schedule IoT tasks. Based on the krill herd algorithm, a new method is proposed for scheduling tasks that minimize network latency and ensure energy efficiency. Our solution outperformed other algorithms, such as PSO and simulated annealing, when choosing optimal assignment decision policies for task assignments and scheduling. Further, our approach ensured low-latency communication and energy efficiency by optimizing both local and global processes. We intend to examine the similarity between the task capabilities in future work in order to predict the execution path of the tasks. To achieve even better results, we plan to explore how the task capabilities can be better coordinated by comparing them against one another.

Acknowledgment

This work was supported in part by the Jiangsu Overseas Visiting Scholar Program for University Prominent Young & Middle-aged Teachers and Presidents Foundation, in part by the Enterprise-University Research Cooperation Projects of Jiangsu Province, in part by the Enterprise-University Research

Projects of Yangzhou Qingyuan Electrical Equipment Co., Ltd and Yangzhou Kenike Mechanical Technology Co., Ltd

REFERENCES

[1] Mehbodniya, A., et al., Modified Lamport Merkle Digital Signature blockchain framework for authentication of internet of things healthcare data. Expert Systems, 2022. 39(10): p. e12978.

[2] He, P., et al., Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review. Sustainable Computing: Informatics and Systems, 2022. 36: p. 100822.

[3] Haghshenas, S.H., M.A. Hasnat, and M. Naeini, A Temporal Graph Neural Network for Cyber Attack Detection and Localization in Smart Grids. arXiv preprint arXiv:2212.03390, 2022.

[4] Khosravi, F., et al., Improving the performance of three level code division multiplexing using the optimization of signal level spacing. Optik, 2014. 125(18): p. 5037-5040.

[5] Khosravi, F., et al. Implementation of an Elastic Reconfigurable Optical Add/Drop Multiplexer based on Subcarriers for Application in Optical Multichannel Networks. in 2022 International Conference on Electronics, Information, and Communication (ICEIC). 2022. IEEE.

[6] Akhavan, J. and S. Manoochehri. Sensory data fusion using machine learning methods for in-situ defect registration in additive manufacturing: a review. in 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). 2022. IEEE.

[7] Saeidi, S.A., et al. A novel neuromorphic processors realization of spiking deep reinforcement learning for portfolio management. in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022. IEEE.

[8] Vahedifard, F., et al., Artificial intelligence for radiomics; diagnostic biomarkers for neuro-oncology. World Journal of Advanced Research and Reviews, 2022. 14(3): p. 304-310.

[9] Pourghebleh, B. and N.J. Navimipour, Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research. Journal of Network and Computer Applications, 2017. 97: p. 23-34.

[10] Mohseni, M., F. Amirghafouri, and B. Pourghebleh, CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic. Peer-to-Peer Networking and Applications, 2022: p. 1-21.

[11] Pourghebleh, B., et al., A roadmap towards energy-efficient data fusion methods in the Internet of Things. Concurrency and Computation: Practice and Experience, 2022: p. e6959.

[12] Pourghebleh, B., V. Hayyolalam, and A.A. Anvigh, Service discovery in the Internet of Things: review of current trends and research challenges. Wireless Networks, 2020. 26(7): p. 5371-5391.

[13] Attiya, I., et al., An improved hybrid swarm intelligence for scheduling iot application tasks in the cloud. IEEE Transactions on Industrial Informatics, 2022.

[14] Pourghebleh, B., K. Wakil, and N.J. Navimipour, A comprehensive study on the trust management techniques in the Internet of Things. IEEE Internet of Things Journal, 2019. 6(6): p. 9326-9337.

[15] Sellami, B., et al., Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network. Computer Networks, 2022. 210: p. 108957.

[16] Pourghebleh, B. and V. Hayyolalam, A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things. Cluster Computing, 2019: p. 1-21.

[17] Wang, B., P. Wu, and M. Arefzaeh, A new method for task scheduling in fog-based medical healthcare systems using a hybrid nature-inspired algorithm. Concurrency and Computation: Practice and Experience, 2022: p. e7155.

[18] Angelakis, V., et al., Allocation of heterogeneous resources of an IoT device to flexible services. IEEE Internet of Things Journal, 2016. 3(5): p. 691-700.

[19] Kamalov, F., et al., Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective. Sustainability, 2023. 15(4): p. 3317.

[20] Kim, M. and I.-Y. Ko. An efficient resource allocation approach based on a genetic algorithm for composite services in IoT environments. in 2015 IEEE international conference on web services. 2015. IEEE.

[21] Aerts, J.C. and G.B. Heuvelink, Using simulated annealing for resource allocation. International Journal of Geographical Information Science, 2002. 16(6): p. 571-587.

[22] Ammar, M., G. Russello, and B. Crispo, Internet of Things: A survey on the security of IoT frameworks. Journal of Information Security and Applications, 2018. 38: p. 8-27.

[23] Kumar, A., et al., Smart power consumption management and alert system using IoT on big data. Sustainable Energy Technologies and Assessments, 2022: p. 102555.

[24] Rahmani, A.M., Z. Babaei, and A. Souri, Event-driven IoT architecture for data analysis of reliable healthcare application using complex event processing. Cluster Computing, 2021. 24(2): p. 1347-1360.

[25] Seyfollahi, A., T. Taami, and A. Ghaffari, Towards developing a machine learning-metaheuristic-enhanced energy-sensitive routing framework for the internet of things. Microprocessors and Microsystems, 2023. 96: p. 104747.

[26] Srinadh, V. and P.N. Rao. Implementation of Dynamic Resource Allocation using Adaptive Fuzzy Multi-Objective Genetic Algorithm for IoT based Cloud System. in 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT). 2022. IEEE.

[27] Sekaran, R., et al., Ant colony resource optimization for Industrial IoT and CPS. International Journal of Intelligent Systems, 2021.

[28] Taami, T., S. Azizi, and R. Yarinezhad, An efficient route selection mechanism based on network topology in battery-powered internet of things networks. Peer-to-Peer Networking and Applications, 2022: p. 1-16.

[29] Meisami, S., M. Beheshti-Atashgah, and M.R. Aref, Using Blockchain to Achieve Decentralized Privacy In IoT Healthcare. arXiv preprint arXiv:2109.14812, 2021.

[30] Colistra, G., V. Pilloni, and L. Atzori, The problem of task allocation in the Internet of Things and the consensus-based approach. Computer Networks, 2014. 73: p. 98-111.

[31] Khalil, E.A., S. Ozdemir, and S. Tosun, Evolutionary task allocation in Internet of Things-based application domains. Future Generation Computer Systems, 2018. 86: p. 121-133.

[32] Yao, J. and N. Ansari. Energy-aware task allocation for mobile IoT by online reinforcement learning. in ICC 2019-2019 IEEE International Conference on Communications (ICC). 2019. IEEE.

[33] Xiao, K., et al., EdgeABC: An architecture for task offloading and resource allocation in the Internet of Things. Future Generation Computer Systems, 2020. 107: p. 498-508.

[34] Mebrek, A. and A. Yassine, Intelligent resource allocation and task offloading model for IoT applications in fog networks: a game-theoretic approach. IEEE Transactions on Emerging Topics in Computational Intelligence, 2021.

[35] Najafizadeh, A., et al., Multi-objective Task Scheduling in cloud-fog computing using goal programming approach. Cluster Computing, 2022. 25(1): p. 141-165.

[36] Hussain, S.M. and G.R. Begh, Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog–cloud environment. Journal of Computational Science, 2022. 64: p. 101828.

[37] Weikert, D., C. Steup, and S. Mostaghim. Multi-Objective Task Allocation for Dynamic IoT Networks. in 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS). 2022. IEEE.

[38] Abdelkader, D.M. and F. Omara, Dynamic task scheduling algorithm with load balancing for heterogeneous computing system. Egyptian Informatics Journal, 2012. 13(2): p. 135-145.

Marigold Flower Blooming Stage Detection in Complex Scene Environment using Faster RCNN with Data Augmentation

Sanskruti Patel
Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Charotar University of Science and Technology
Changa, India

***Abstract*—In recent years, flower growing has developed into a lucrative agricultural sector that provides employment and business opportunities for small and marginal growers in both urban and rural locations in India. One of the most often cultivated flowers for landscaping design is the Marigold flower. It is also widely used to create garlands for ceremonial and social occasions using loose flowers. Understanding the appropriate stage of harvesting for each plant species is essential to ensuring the quality of the flowers after they have been picked. It has been demonstrated that human assessors consistently used a category scoring system to evaluate various flowering stages. Deep learning and convolutional neural networks have the potential to revolutionize agriculture by enabling efficient analysis of large-scale data. In order to address the problem of Marigold flower stages detection and classification in complex real-time field scenarios, this study proposes a fine-tuned Faster RCNN with ResNet50 network coupled with data augmentation. Faster RCNN is a popular deep learning framework for object detection that uses a region proposal network to efficiently identify object locations and features in an image. The Marigold flower dataset was collected from three different Marigold fields in the Anand District of Gujarat State, India. The collection includes of photos that were taken outdoors in natural light at various heights, angles, and distances. We have developed and fine-tuned a Faster RCNN detection and classification model to be particularly sensitive to Marigold flowers, and we have compared the generated method's performance to that of other cutting-edge models to determine its accuracy and effectiveness.**

Keywords—Deep learning; convolutional neural networks; object detection; marigold flower blooming stage detection

I. Introduction

One of the main economic pillars in India is agriculture. For roughly 58% of Indians, agriculture is their main source of income. The field of horticulture known as "flower farming," also referred to as "floriculture," deals with the study of cultivating and selling flowers and foliage plants. It primarily focuses on growing ornamental plants, cultivated greens, potted flowering plants, tubers, rooted cuttings, cut flowers, and other floriculture products. In recent years, flower farming has become a successful agriculture industry that offers employment and entrepreneurship prospects in both urban and rural areas, as well as for small and marginal farmers [1]. One of the most often cultivated flowers for garden ornamentation is the Marigold, which is also widely used as loose flowers to create garlands for ceremonial and social occasions. The Marigold is one of the most popularly grown flowers for landscape decoration. It is also frequently used as loose flowers to make garlands for ceremonial and social occasions. Marigold is mostly used to treat various skin disorders, such as varicose veins, contusions, and bruises. Additionally, inflammation and minor skin wounds can be successfully addressed. Marigold cream aids in the healing of sunburns and eczema wounds. Marigold farming is a profitable activity that requires little maintenance and effort. Marigold cultivation is a profitable activity as it requires less investment and gives better harvest with a high profit [2].

To guarantee the quality of the flowers after harvest, it is crucial to understand the ideal stage of harvesting for each plant type. The flower's life is considerably decreased when it is harvested too early or too late. A flower normally becomes larger as it progresses from bud to bloom. A flower like a daisy, Marigolds can only be picked when completely opened [3].

Identifying the plant flowering status have traditionally needed human evaluators to manually inspect flower fields and report flowering status. It has been shown that human assessors regularly assessed different flowering stages using a category scoring system. For instance, you could want to know when 30% of the flowering plants in a field have blooms that are fully open. This made it possible for researchers to compute the time between various blooming phases [4]. Deep learning advancements and innovations make it possible to quickly characterize the flowering patterns of field-grown plants. It is frequently necessary to regularly spot and count newly opening blooms on plants when cultivating flowers like Marigold.

Marigold farming is a profitable activity as it requires less investment and gives better harvest with a high profit. To guarantee the quality of the flowers after harvest, it is important to understand the ideal stage of harvesting for each plant type. Deep learning advancements and innovations make it possible to quickly characterize the flowering patterns of field-grown plants. Using a cutting-edge object detector called Faster Region-based Convolutional Neural Network, we propose an efficient method to detect and classify Marigold flowers of various stages in diverse field conditions. The proposed method is inspired by successful studies using deep

Convolutional Neural Networks (CNNs) in difficult computer vision and object detection tasks.

This study suggests a Faster RCNN network coupled with image augmentation to address the challenge of Marigold flower stages identification and classification in complicated real-time field situations. In order to detect and classify emerging blooms of Marigold flower plants, the objectives of the study are: (1) To acquire and pre-process images of Marigold flowers in challenging real-time environments; (2) To develop and tune a Faster RCNN detection and classification model to become particularly sensitive to Marigold flowers; and (3) To assess the accuracy and efficacy of the developed approach by comparing its performance to that of other cutting-edge models.

II. Related Work

D. Thi Phuong Chung and D. Van Tai [5] represent a deep learning based technique for fruit detection. They have e EfficientNet architecture that recognized fruit objects from the Fruit 360 dataset and achieved 95% accuracy. A. Rocha et al. [6] introduced a novel method for classification of fruit and vegetables from images. A multi-class fruit-and-vegetable categorization task in a semi-controlled setting, like a distribution centre or the supermarket checkout line, is used to validate the newly presented fusion approach. According to the findings, the solution can lower the classification error by up to 15% compared to the baseline. I. Sa et al. [7] present a novel approach to fruit detection using deep convolutional neural networks. The goal is to develop a fruit detection system as it is a critical component of an autonomous agricultural robotic platform and is essential for estimating fruit production and automating harvesting. They have proposed a multi-modal Faster RCNN model that, when compared to earlier work, delivers state-of-the-art results, with F1 score performance for the detection of sweet pepper increasing from 0.807 to 0.838.

Moreover, T. Abbas et al. [8] mentioned different smartphone applications like LeafSnap [9], and Pl@ntNet [10], that can be used to identify flowers rapidly. I. Patel and S. Patel [11] proposed an optimized deep learning model that detects the flower species. For that, they have integrated Faster RCNN with Neural Network Search and Feature Pyramid. The mAP score obtained on the standard Oxford flower species dataset is 87.6%. D. Wu et al. [12] proposed a methodology for detecting Camellia oleifera Fruit using YOLOv7 object detection model. For the research, they have collected the dataset from the complex scene and applied different evaluation metrics. The values derived for mAP is 96.03%, Precision is 94.76%, Recall is 95.54%, and F1 score is 95.15%. S. Nuanmeesri et al. [13] proposed a novel method that predicts disease from Marigold flower images. The outcome demonstrated that the model created using the watershed dataset is the most effective. The model's validation accuracy was 88.03%, validation loss was 4.21%, and model testing accuracy was 91.67%.

From the literature survey, we have found that deep learning algorithms and models can be applied for detecting objects from real-time environment and it has a great significance and theoretical value. Moreover, there is a need for an automated model that provides improved accuracy and generalization across different growing conditions and environmental factors. There are challenges existed in developing models that can handle the variability in marigold flower appearance across different growing conditions, such as varying lighting, background, or growth stages. This research aimed to develop and propose a more sophisticated model that can handle these challenges and improve the accuracy and generalization of marigold flower blooming stage identification and classification.

III. Materials and Methods

A. Acquisition and Pre-Processing of Marigold Flower Images in Complex Scene Environment

One Canon DSLR Dual Lens Camera and two smartphones were used to capture images of Marigold flowers in the Marigold cultivation agricultural fields under natural daylight illumination. Three distinct agricultural Marigold fields in the Anand District of Gujarat, India, were chosen for the study. Three regular harvesting times in the winter November and December months have chosen for the capturing of images. The collection consists of 550 photos in total, each with a resolution of 4000 by 2250 pixels and taken at various heights, angles, and random distances in natural light conditions. The dataset was captured in two stages. The acquired images are having different conditions like top angled, side angled, heavily occluded, lightly occluded, overlapped, etc. are represented in the Fig. 1.

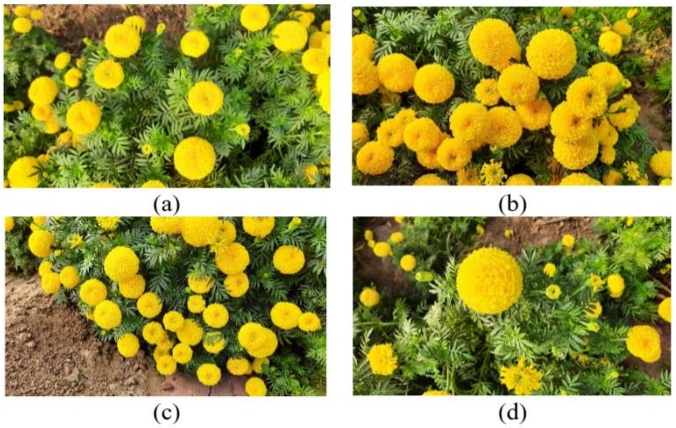


Fig. 1. Examples of Marigold images taken in various settings (a) Occlusion in image, (b) Overlapped marigold flowers and buds, (c) Image captured from side angle (d) Image captured from top angle.

Because they provide the supervised learning algorithm with the training data, image annotations play a key role in computer vision algorithms. Using the graphical image annotation tool namely, LabelImg [14], which is built on Python, the complete dataset was annotated and saved in XML documents. The images are annotated in two classes; bud and flower which are represents and differentiates their growing stages.

B. Data Augmentation

Data augmentation refers to the process of artificially expanding the size of a training dataset by creating modified versions of images in the dataset [15]. This can be useful in object detection, especially for Faster RCNN, as it helps prevent overfitting and can improve the model's ability to generalize to new, unseen data. The most popular technique of fundamental augmentation is geometric transformation [16]. The transformation's parameters may be preset or chosen at random. In this research, the common techniques for data augmentation used are flipping, scaling and rotating. Flipping is the process used to rotate an image either along its vertical axis or along its horizontal axis. In contrast to vertical flipping, which flips the image on the vertical axis, horizontal flipping flips the image on the horizontal axis [17]. An image can be rotated by adding a rotational angle. The image is rotated in a random direction to produce enhanced images. Here, the left and right axes of rotation are chosen at random. The dataset includes an image that can be zoomed in or out. One of the most popular data augmentation techniques is zooming. It is possible to conduct a zoom between 0.5% and 1.0%.

After the data augmentation applied, the final augmented training set consists of 1583 images that help to improve the generalization ability of the detection model and avoid the overfitting of the detection model.

IV. Proposed Model for Marigold Flower Blooming Stage Detection in Complex Scene Environment using Faster RCNN with Data Augmentation

The Fig. 2 represents the stepwise architecture of the proposed model for Marigold flower stages identification and classification.

There are two main types of object detection models: one-stage models and two-stage models. One-stage object detection models, also known as single-shot detectors, are designed to detect objects in a single pass over an input image [18]. These models use a single neural network to simultaneously predict object locations and class labels. Two-stage object detection models, on the other hand, are designed to detect objects in a two-step process. The first step involves generating a set of potential object locations, known as region proposals, using a separate network called a region proposal network (RPN). The second step involves classifying the region proposals and refining their locations using another network. Two-stage models typically achieve higher accuracy than one-stage models, but at the cost of slower inference speed. One-stage models, on the other hand, are faster but can be less accurate, especially for small objects or objects with high aspect ratios [19] [20]. This research is mainly focuses on identifying two stages of Marigold flower growth; fully grown flower and a bud. Bud is a small flower object that is to be detected by the proposed model. Therefore, in this research we have proposed a two-stage object detection model i.e. Faster RCNN.

We are primarily interested in object detection in our study because it is the first step in determining whether a flower is a bud or a fully blown blossom. So, using Faster RCNN, we simulate a particular generic detector. In order to create an effective technique for looking for instances of flowers and buds in a flower image, we make use of the object suggestions trained by an RPN and their associated features derived from a ResNet50 CNN architecture. By combining the convolutional strengths of RPN and Fast RCNN utilising the present neural network formulation, we further combine RPN and Fast RCNN into a coherent model. The feature network, RPN, and detection network are the three deep networks that make up the suggested methodology. A boxing approach used by faster RCNN enables the operator to specify the potential regions that will be introduced into the RPN. With the suggested approach, we begin by performing a CNN model using our dataset of Marigold flowers. After examining the input image, a selective search procedure is then used to extract a region of interest (RoI). The candidates between the closest raster frames are then refined using the prepared deep model to classify the extracted ROIs into candidates.

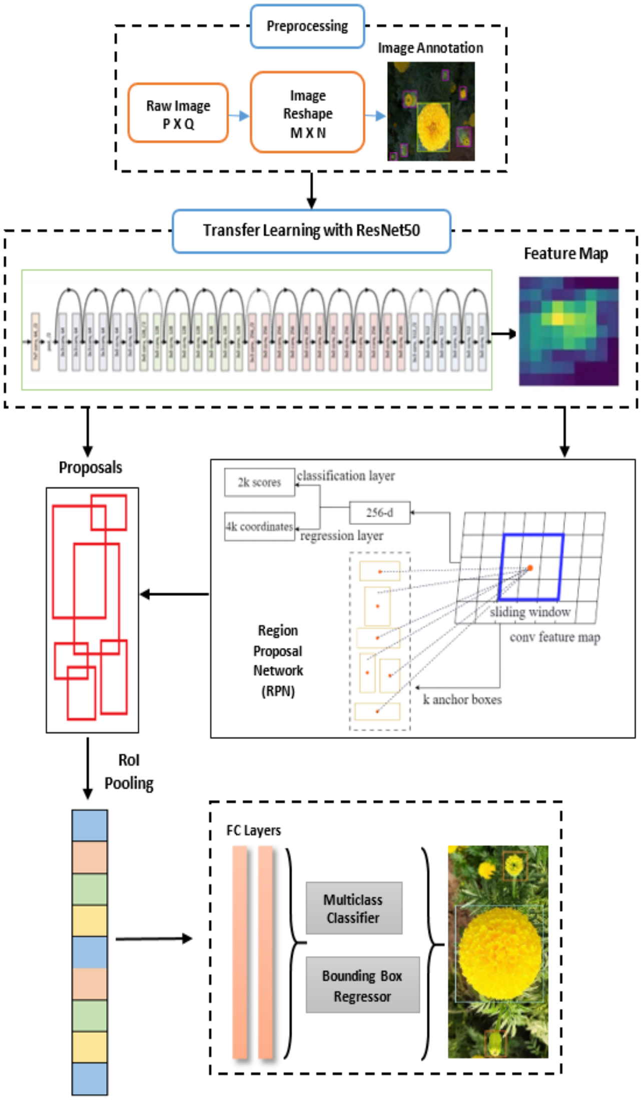


Fig. 2. A proposed two-stage object detection model for marigold flower blooming stage detection in complex scene environment using faster RCNN with data augmentation.

A. Image Pre-Processing and Annotation

In object detection, pre-processing refers to the steps taken to prepare the input data for the object detection model. This may include tasks such as resizing the image, normalizing pixel values, converting the image to grayscale, etc. [21]. In this research, image is reshaped and annotated with two labels i.e. flower and bud. The dataset is splitted into training and validation sets by having a ratio of 90:10. No repeated images among the training, validation and test sets were ensured to prevent overfitting of the model [22] [23].

Image annotation in object detection refers to the process of labeling objects within an image to train a machine learning model for object detection. The annotated data is used to train the model to detect and classify objects within new, unseen images [24]. Image annotation involves drawing bounding boxes around objects of interest and assigning a label to each bounding box. The goal of image annotation is to provide the model with enough data to learn the features and characteristics of different objects, so it can accurately detect them in new images [25]. LabelImg is a graphical image annotation tool that is used to label images for object detection in machine learning. The tool provides an interface for drawing bounding boxes around objects in an image and assigning class labels to the objects. The resulting annotations are saved in an XML file that can be used as input for training machine learning models. The Fig. 1 illustrates the image annotation performed using LabelImg tool. As mentioned earlier, the images are annotated in two classes; bud and flower which are represents and differentiates their growing stages.

Here, a 2D bounding boxes annotation for flower species detection is applied, as illustrated in Fig. 3. The 2D bounding boxes are applied by drawing rectangles or cuboids around flower objects in an image, and then, labels of respective flower classes are applied to them.

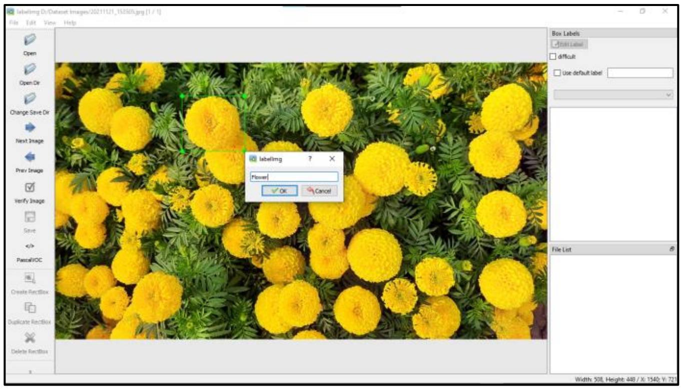


Fig. 3. Image annotation using lableImg.

B. Faster RCNN with ResNet50

Faster RCNN is a popular object detection architecture that is used for Marigold flower stages detection. Faster RCNN is a state-of-the-art object detection algorithm that combines the two-stage object detection framework with deep convolutional neural networks [26]. The first stage is a region proposal network (RPN) that generates a set of candidate object regions. These regions are then fed into the second stage, which is a Fast RCNN network that classifies the regions and refines the bounding box locations. The two stages work together to efficiently detect objects in an image by first proposing a large number of potential regions and then using the Fast RCNN network to accurately classify and locate the objects. The key advantage of Faster RCNN is its end-to-end training, which enables it to learn to detect objects directly from image data without relying on heuristics or manually-designed features [27].

The proposed Faster RCNN model with ResNet50 for flower object detection can be divided into the following modules:

1) ResNet50 backbone: This module consists of the pre-trained ResNet50 network that serves as the feature extractor. It takes an image as input and outputs a feature map [28]. In Faster RCNN, the ResNet-50 backbone is used as the feature extractor network to produce a compact feature representation of the input image [29]; the input image being fed into the backbone convolutional neural network. For that, the input image is first resized by considering the shortest px with the longer side not exceeding 1000px. The output of the backbone network is a feature map. These feature maps are then fed into the Fast RCNN network for classification and bounding box regression. The use of a pre-trained ResNet50 network as the feature extractor allows Faster RCNN to leverage the information learned from the large-scale image classification task, improving its object detection performance. Additionally, the use of ResNet50 as a backbone allows for transfer learning, where the feature extractor can be fine-tuned for the specific object detection task using a smaller dataset [30].

With the primary goal of resolving the vanishing/exploding gradient issue, ResNet architecture established the Residual Network concept. The network employs a method known as "skip connection" for that. The ResNet architecture is known for its use of "skip connections" or "shortcut connections" [31]. These skip connections help alleviate the problem of vanishing gradients in very deep neural networks. Skip connections in ResNet work by allowing the network to bypass one or more layers, effectively allowing the gradients to be backpropagated directly to earlier layers as illustrated in Fig. 4. This helps to preserve the information from the original input, making it easier for the network to learn and improve.

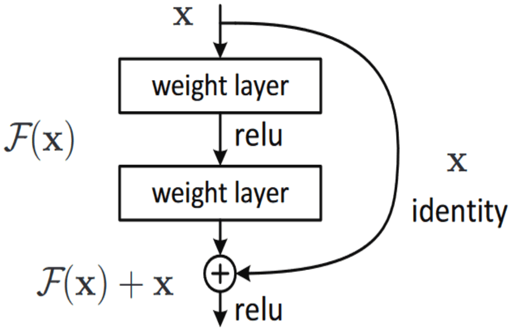

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 4. An illustration of skip connection [32].

Without using skip connection, input 'x', multiplied by the layer's weights, followed by adding a bias term:

$$H(x) = f(wx + b) \tag{1}$$

or

$$H(x) = f(x) \tag{2}$$

With the introduction of skip connection, the output of the layer changes to

$$H(x) = f(x) + x \tag{3}$$

The loss function used in ResNet50, like most of deep learning models, is typically a categorical cross-entropy loss. This loss measures the dissimilarity between the predicted class probabilities and the true class label, and is commonly used for multi-class classification problems. The following is an equation to calculate the categorical cross-entropy [33].

$$Cross - entropy = -\frac{1}{N}\sum_{i=1}^{N} logP_{model}\left[y_i \in C_{y_i}\right] \quad (4)$$

2) Region Proposal Network (RPN): The Region Proposal Network (RPN) is a crucial component of the Faster RCNN object detection framework. Its primary responsibility is to generate a set of candidate object regions in the input image, called region proposals. A region proposal network generates several regional proposals [34]. These proposals submit to the identification network's detection. The three components of RPN are the anchor window, loss function, and set of region proposals [35]. RPN adopts the sliding window methodology because a small sub-network is evaluated on a dense 3x3 sliding window in the RPN design. The IoU ratios and the ground-truth bounding boxes can thus be used by the RPN to produce numerous anchors [36].

The RPN uses anchor boxes, which are predefined bounding box shapes, to guide the generation of the region proposals. The network outputs are then combined with the anchor boxes to produce the final set of region proposals. The step-wise process is described as follows [37]: (i) RPN utilize a sliding window for each region over the feature map. (ii) To generate region proposals, k (k=9) anchor boxes are employed for each site, with 3 scales of 128, 256, and 512 and 3 aspect ratios of 1:1, 1:2, and 2:1. (iii) Whether an object is present or not, a CLS layer produces 2k scores for k boxes. (iv) For the box centre coordinates, width, and height of k boxes, a reg layer outputs 4k. (v) There are WHk anchors overall with a WH feature map size.

The total loss of the RPN is calculated by the multitask loss function. The calculation formula is [38].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\,(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}\,(t_i,t_i^*) \quad (5)$$

Where N_{cls} represents the number of batch training data, N_{reg} represents the number of anchors, λ represents the balance weight.

$L_{cls}\,(p_i,p_i^*)$ is the logarithmic loss function defined as;

$$L_{cls}\,(p_i,p_i^*) = -\log[\,p_i^* p_i + (1 - p_i^*)(1 - p_i)\,] \quad (6)$$

$p_i^* L_{reg}\,(t_i,t_i^*)$ is the regression loss calculated by the following Smooth L1 function:

$$L_{reg}\,(t_i,t_i^*) = \begin{cases} 0.5\,(t_i - t_i^*)^{\,2}\,, & |\,\mathrm{x}\,| < 1 \\ |t_i - t_i^*| - 0.5, & otherwise \end{cases} \quad (7)$$

Where p_i is the probability of the anchor being predicted as the target, and p_i^* is the truth value of the prediction result: if the anchor is predicted as a positive sample, the value of tag p_i^* is 1; otherwise, the value is 0; $t_i = \{t_x,\ t_y, t_w, t_h\}$ is the location of the predicted detection box; and t_i^* is the ground truth coordinate.

RPN network must therefore check in advance which location contains the object. The detection network will then receive the relevant locations and bounding boxes and use them to identify the object class and deliver the object's bounding box.

3) RoI pooling layer: RoI (Region of Interest) pooling is a technique used in Faster RCNN for processing the region proposals generated by the RPN [39]. RoI pooling is a layer in the Fast RCNN network that takes as input the feature map produced by the ResNet-50 backbone and a set of region proposals. The RoI pooling layer resizes each region proposal to a fixed size, regardless of its original size or aspect ratio, and aggregates the features within each region into a compact feature representation [40]. This enables the Fast RCNN network to perform classification and regression on the objects in the image, regardless of their size and aspect ratio. RoI pooling is critical for Faster RCNN's ability to accurately detect and classify objects of different sizes and aspect ratios in an image. The RoI pooling layer allows the Fast RCNN network to have a fixed input size, making it easier to train and optimize, while still allowing it to handle objects of varying sizes in the image.

4) Fast RCNN classifier and bounding box regressor: In Faster RCNN, after the RoI pooling layer, the features of the region proposals are fed into the classifier and bounding box regressor [41]. The classifier is a fully connected layer that performs object classification by predicting the probability of each region proposal belonging to each of the predefined object classes. The classifier outputs a score for each region proposal and class, indicating the likelihood of the presence of an object of that class in the region. The bounding box regressor is another fully connected layer that performs bounding box regression [42]. It takes as input the feature representation of the region proposals and outputs the adjustments to the locations of the region proposals, refining their locations to better fit the objects in the image.

Together, the classifier and bounding box regressor form the Fast RCNN network, which accurately detects and classifies objects in the image by combining the information from the region proposals, the classifier scores, and the refined bounding box locations.

V. Network Training Platform and Parameter Settings

The experiment is carried out using a machine having NVIDIA Tesla V100 32GB PCIe based GPU card equipped with 64GB RAM and Intel Xeon 6226R processor. It has 4 units of 6TB SATA 7200 RPM 3.5" HDD and it is running with Ubuntu 21.04 operating system. We have used TensorFlow 2 Object Detection API with CUDA 11.2, CuDNN 8.1.0 and Python 3.8 virtual environment.

An illustration of a training pipeline developed for experiment with numerous separate activities is shown in Fig. 5. The image annotation tool is used to build the labelled flower datasets (i.e. LabelImg). All of the datasets for tagged flowers are saved as .csv files, which are then transformed into .record files and used as inputs by the networks to forecast bounding boxes and confidences for objects. TensorFlow's object identification model needs a Label Map that converts each of the applied labels into an integer value. Both the training and evaluation processes use this Label Map. Files with the ending ".pbtxt" are label map files. We have used ResNet50 as a pre-trained CNN and modified the Faster RCNN detection model that is trained using COCO 2017 dataset and made available by Tensorflow Object Detection API - Model Zoo [43]. Finally, loss functions are used to measure the accuracy of the training process and an inference graph are generated at the end of the training pipeline.

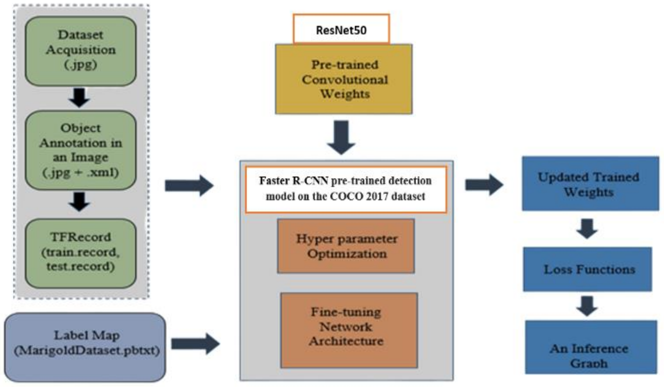

Dataset Acquisition (.jpg)
Object Annotation in an Image (.jpg + .xml)
TFRecord (train.record, test.record)
Label Map (MarigoldDataset.pbtxt)
ResNet50
Pre-trained Convolutional Weights
Faster R-CNN pre-trained detection model on the COCO 2017 dataset
Hyper parameter Optimization
Fine-tuning Network Architecture
Updated Trained Weights
Loss Functions
An Inference Graph

Fig. 5. A training pipeline for experiment.

A. Hyperparameters

The model parameter values that a learning algorithm ultimately learns are defined by hyperparameters, which are variables whose values have an impact on the learning process. The selection of hyperparameters that aid in an object identification model's optimum accuracy has an impact on the model's accuracy as well. Therefore, figuring out the best values for these factors is a challenging task [44]. Hyperparameter tuning, often known as optimization, is the process of selecting the best set of parameters for a model's learning procedure [45]. In this research, we have setup multiple hyperparameters such as learning rate, batch size, number of steps, activation function and dropout rate. The learning rate for the proposed model sets to 0.002, the batch size chose was 16. The input size was set to 640 × 640. The training Epoch was set to 1000. During the training process, Tensorboard visualization tool was used to record data and observe loss, and save the model weight of every epoch.

B. Evaluation Indicators of Model

In this study, the model's performance was accurately and impartially assessed using Precision, Recall, Mean Average Precision (mAP), and F1 score. The number of correct targets divided by the total number of targets is known as the precision evaluation index [46]. The detection impact will generally be better the higher the Precision. Precision is a highly logical evaluation metric, however occasionally a high Precision score does not mean everything. Thus, mAP, Recall, and F1 score were developed for thorough examination.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \; X\ 100 \qquad (8)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \; X\ 100\% \quad (9)$$

$$\text{Average Precision (AP)} = \int_1^0 P(r)\, dr \quad (10)$$

$$\text{Mean Average Precision (mAP)} = \frac{1}{n}\sum_{i=1}^{n} AP_I \qquad (11)$$

Other than above evaluation indicators, IoU is also used. The amount of overlap between the predicted and ground truth bounding boxes is indicated by the IoU value, which ranges from 0 to 1 as described in Fig. 6 [47].

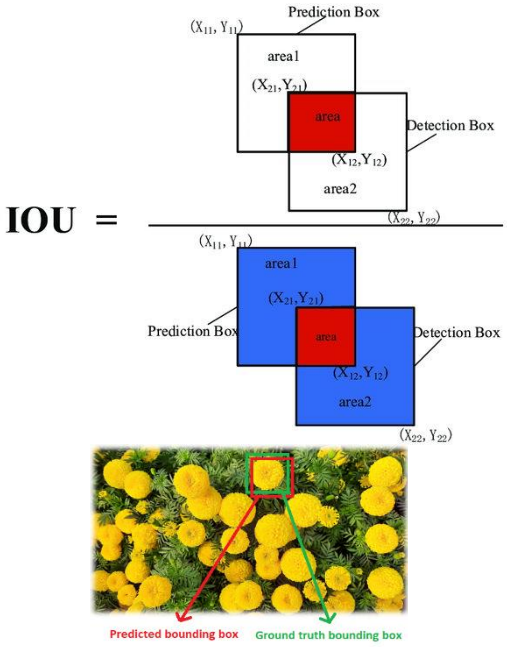


Fig. 6. A description of Intersection-over-Union (IoU)

There is no overlap between the boxes if the IoU is 0. When the union of the boxes equals their overlap and the IoU = 1, this signifies that the boxes are entirely overlapping. The equation for the same is illustrated as Eq. (3).

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (12)$$

VI. Results and Discussion

To experiment, we have used TensorFlow Object Detection API. It is an open-source framework which is built on top of the TensorFlow library, offers a variety of pre-trained object detection models as well as tools for creating and training unique object detection models. The pre-trained models, also known as the Model Zoo, feature various models that are pre-trained on the COCO dataset, which is a large-scale object detection, segmentation, and captioning dataset. We have used the learned weights from these pre-trained model and fine-tuning these pre-trained models on our own Marigold datasets.

The proposed Faster RCNN with ResNet50 model is compared with one-stage object detection model that is SSD (Single Shot Detector). By examining the Faster RCNN of various networks, the training model and the subsequent classification results of mean average precision are obtained and presented in Table I and Table II. We have experimented both of the object detection model with the original dataset and augmented dataset. On the original Marigold dataset, the proposed Faster RCNN with ResNet50 model provides 88.71% mAP score with 4.312 average detection speed per second. SSD MobileNet model obtained 74.30% mAP score which is lower as compare to the proposed Faster RCNN with ResNet50 model.

We have synthetically increases the size of the dataset and experimented both of the model. This time, the proposed Faster RCNN with ResNet50 model provides 89.47% mAP score and SSD MobileNet model obtaines 78.12% mAP score. Although, it is observed that the average detection speed of SSD MobileNet is quite high as compare to the Faster RCNN with ResNet50.

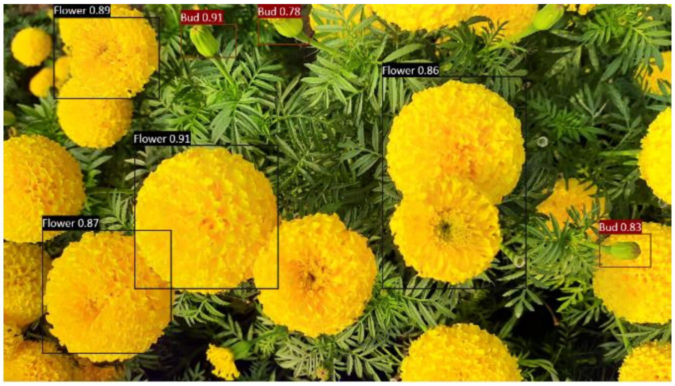
TABLE I. RESULTS AND PERFORMANCE COMPARISON AMONG DIFFERENT DETECTION MODELS

Data Augmentation	Target Detection Networks	mAP (%)
Without Data Augmentation	Proposed Faster RCNN ResNet50 V1 640X640	88.71
	SSD MobileNet V1 FPN 640X640	74.30
With Data Augmentation	Proposed Faster RCNN ResNet50 V1 640X640	89.47
	SSD MobileNet V1 FPN 640X640	78.12

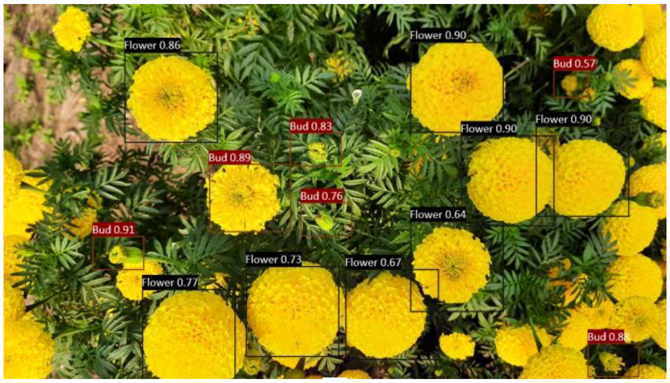
TABLE II. COMPARISON OF AVERAGE DETECTION SPEED AMONG DIFFERENT DETECTION MODELS

Target Detection Networks	Average Detection Speed (s/Image)
Proposed Faster RCNN ResNet50 V1 640X640	4.312
SSD MobileNet V1 FPN 640X640	0.64

Fig. 7 contains the examples of Intersection over Union (IoU) detection results of the proposed object detection on real-time Marigold Dataset captured in complex scene environment. The figure (a) and (b) are the images with detection results. It shows the detection of two classes; flower and bud. These two classes represent the flower growing stages that can be helpful to decide the harvesting time.



(a)



(b)

Fig. 7. Examples of Intersection over Union (IoU) detection results of the proposed object detection on real-time marigold dataset captured in complex scene environment. Fig. 7(a) and 7(b) are the images with detection of flower and bud objects

In conclusion, several studies have been conducted in the past to address the issue of marigold flower object identification and classification. Marigold flower blooming stage identification and classification using deep learning techniques such as Faster RCNN, YOLO, and SSD has gained significant attention in recent years. These techniques offer faster and more accurate detection and classification of marigold flower blooming stages. YOLO and SSD are faster, they are also less accurate than the two-stage models, particularly on small objects [48]. Several researchers proposed variety of methods for classification using the machine learning and deep learning algorithms that achieved good performance. However, they have considered image classification based on discrete features of flower image [49][50][51][52]. In this research, we have proposed a method based on two-stage object detection. The proposed method allows for more precise localization and classification of objects and can be used to quickly and accurately identify the blooming stage of marigold flowers. Moreover, two-stage object detectors are more robust to variations in lighting conditions, background clutter, and occlusions compared to single-stage detectors. This makes them ideal for identifying marigold flowers blooming stages, which can be affected by different lighting conditions and may have complex backgrounds.

VII. CONCLUSION

In order to detect Marigold flower stages in intricate agricultural field settings, a real-time and accurate identification strategy based on a two-stage Faster RCNN object detection network with data augmentation was presented in this study. We have gathered and analysed data on Marigold flowers in a variety of field settings as part of our research. All of the dataset's images were divided into two classes: bud and flower. The flower growth stage is represented by these two classes. Geometric data augmentation techniques were also used to improve the dataset. Then, utilising the ResNet50 backbone network, we fine-tune the two-stage object detector, namely Faster RCNN. We conducted an experiment and compared the outcomes with SSD MobileNet, a single-stage object identification model. The findings suggest that data augmentation can significantly enhance the proposed model's capacity for detection. The Faster RCNN with ResNet50 model has been proposed an 89.47 mAP score and a 4.312 average detection speed per second. The detection of two classes; flower and bud represent the flower growing stages that can be helpful to decide the harvesting time for the Marigold flowers. This research provides a pathway for the researchers who are working for the automatic detection and harvesting of flowers other than the Marigold.

REFERENCES

[1] M. A. Wani et al., “Floriculture sustainability initiative: The dawn of New Era,” in Sustainable Agriculture Reviews 27, Cham: Springer International Publishing, 2018, pp. 91–127.

[2] Jagdish, “Marigold Cultivation Income, yield, project report,” Agri Farming, 08-Sep-2019. [Online]. Available: https://www.agrifarming.in/marigold-cultivation-income-yield-project-report. [Accessed: 20-Feb-2023].

[3] “Harvesting and handling cut flowers,” Center for Agriculture, Food, and the Environment, 04-Aug-2016. [Online]. Available: https://ag.umass.edu/greenhouse-floriculture/fact-sheets/harvesting-handling-cut-flowers. [Accessed: 20-Feb-2023].

[4] Y. Jiang, C. Li, R. Xu, S. Sun, J. S. Robertson, and A. H. Paterson, “DeepFlower: a deep learning-based approach to characterize flowering patterns of cotton plants in the field,” Plant Methods, vol. 16, no. 1, p. 156, 2020.

[5] D. Thi Phuong Chung and D. Van Tai, “A fruits recognition system based on a modern deep learning technique,” J. Phys. Conf. Ser., vol. 1327, no. 1, p. 012050, 2019.

[6] Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, “Automatic fruit and vegetable classification from images,” Comput. Electron. Agric., vol. 70, no. 1, pp. 96–104, 2010.

[7] Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, “DeepFruits: A fruit detection system using deep neural networks,” Sensors (Basel), vol. 16, no. 8, p. 1222, 2016.

[8] T. Abbas et al., “Deep neural networks for automatic flower species localization and recognition,” Comput. Intell. Neurosci., vol. 2022, p. 9359353, 2022.

[9] N. Kumar et al., “Leafsnap: A computer vision system for automatic plant species identification,” in Computer Vision – ECCV 2012, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 502–516.

[10] Joly et al., “Interactive plant identification based on social image data,” Ecol. Inform., vol. 23, pp. 22–34, 2014.

[11] Patel and S. Patel, “An Optimized Deep Learning Model for Flower Classification Using NAS-FPN and Faster R-CNN,” International Journal of Scientific & Technology Research, vol. 9, no. 03, pp. 5308–5318, 2020.

[12] D. Wu et al., “Detection of Camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation,” Appl. Sci. (Basel), vol. 12, no. 22, p. 11318, 2022.

[13] S. Nuanmeesri, S. Chopvitayakun, P. Kadmateekarun, and L. Poomhiran, “Marigold flower disease prediction through deep neural network with multimodal image,” Int. J. Eng. Trends Technol., vol. 69, no. 7, pp. 174–180, 2021.

[14] “LabelImg,” PyPI. [Online]. Available: https://pypi.org/project/labelImg/1.4.0/. [Accessed: 20-Feb-2023].

[15] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” J. Big Data, vol. 6, no. 1, 2019.

[16] T. Luke and N. Geoff, Improving deep learning using generic data augmentation. 2017.

[17] Z. Keita, “Five simple image data augmentation techniques to mitigate overfitting in Computer Vision,” Towards Data Science, 19-Mar-2021. [Online]. Available: https://towardsdatascience.com/simple-image-data-augmentation-technics-to-mitigate-overfitting-in-computer-vision-2a6966f51af4. [Accessed: 20-Feb-2023].

[18] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, “On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data,” Remote Sens. (Basel), vol. 13, no. 1, p. 89, 2020.

[19] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, “An evaluation of deep learning methods for small object detection,” J. Electr. Comput. Eng., vol. 2020, pp. 1–18, 2020.

[20] I. Patel and S. Patel, “A comparative analysis of applying object detection models with transfer learning for flower species detection and classification,” International Journal on Emerging Technologies, vol. 11, no. 3, pp. 303-312, 2020.

[21] Maharana, S. Mondal, and B. Nemade, “A review: Data pre-processing and data augmentation techniques,” Global Transitions Proceedings, vol. 3, no. 1, pp. 91–99, 2022.

[22] Y. Gu, S. Wang, Y. Yan, S. Tang, and S. Zhao, “Identification and analysis of emergency behavior of cage-reared laying ducks based on YoloV5,” Agriculture, vol. 12, no. 4, p. 485, 2022.

[23] G. B. Rajendran, U. M. Kumarasamy, C. Zarro, P. B. Divakarachari, and S. L. Ullo, “Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM classifier on hybrid pre-processing remote-sensing images,” Remote Sens. (Basel), vol. 12, no. 24, p. 4135, 2020.

[24] R. Potter, “What is the use and purpose of image annotation in object detection?,” Becoming Human: Artificial Intelligence Magazine, 19-May-2021. [Online]. Available: https://becominghuman.ai/what-is-the-use-and-purpose-of-image-annotation-in-object-detection-8b7873a14cd0. [Accessed: 20-Feb-2023].

[25] Y. Lu and S. Young, “A survey of public datasets for computer vision tasks in precision agriculture,” Comput. Electron. Agric., vol. 178, no. 105760, p. 105760, 2020.

[26] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” arXiv [cs.CV], 2015.

[27] M. Shen et al., “Multi defect detection and analysis of electron microscopy images with deep learning,” Comput. Mater. Sci., vol. 199, no. 110576, p. 110576, 2021.

[28] S. Aparna, K. Muppavaram, C. C. V. Ramayanam, and K. S. S. Ramani, “Mask RCNN with RESNET50 for dental filling detection,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 10, 2021.

[29] S. Patel, R. Patel, N. Ganatra, and A. Patel, “Spatial feature fusion for biomedical image classification based on ensemble deep CNN and transfer learning,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 5, 2022.

[30] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-Maadeed, “Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches,” arXiv [cs.CV], 2022.

[31] G. Boesch, “Deep Residual networks (ResNet, ResNet50) - 2023 guide,” viso.ai, 01-Jan-2023. [Online]. Available: https://viso.ai/deep-learning/resnet-residual-neural-network/. [Accessed: 20-Feb-2023].

[32] He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” arXiv [cs.CV], 2015.

[33] E. Koech, “Cross-entropy loss function,” Towards Data Science, 02-Oct-2020. [Online]. Available: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e. [Accessed: 20-Feb-2023].

[34] S. Reshma Prakash and P. Nath Singh, “Object detection through region proposal based techniques,” Mater. Today, vol. 46, pp. 3997–4002, 2021.

[35] Rosebrock, “Intersection over Union (IoU) for object detection,” PyImageSearch, 07-Nov-2016. [Online]. Available: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/. [Accessed: 20-Feb-2023].

[36] F. Gad, “Faster R-CNN explained for object detection tasks,” Paperspace Blog, 13-Nov-2020. [Online]. Available: https://blog.paperspace.com/faster-r-cnn-explained-object-detection/. [Accessed: 20-Feb-2023].

[37] S. Ananth, “Faster R-CNN for object detection,” Towards Data Science, 09-Aug-2019. [Online]. Available: https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46. [Accessed: 20-Feb-2023].

[38] H. Yan, C. Chen, G. Jin, J. Zhang, X. Wang, and D. Zhu, “Implementation of a modified faster R-CNN for target detection

technology of coastal defense radar," Remote Sens. (Basel), vol. 13, no. 9, p. 1703, 2021.

[39] Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," J. Phys. Conf. Ser., vol. 1544, no. 1, p. 012033, 2020.

[40] T. Hoeser and C. Kuenzer, "Object detection and image segmentation with deep learning on Earth observation data: A review-part I: Evolution and recent trends," Remote Sens. (Basel), vol. 12, no. 10, p. 1667, 2020.

[41] Naukri.com. [Online]. Available: https://www.naukri.com/learning/articles/object-detection-using-rcnn/. [Accessed: 20-Feb-2023].

[42] Researchgate.net. [Online]. Available: https://www.researchgate.net/profile/Hadi-Ghahremannezhad/publication/346061113_Vehicle_Classification_in_Video_Using_Deep_Learning/links/5fb985ef92851c933f4d56be/Vehicle-Classification-in-Video-Using-Deep-Learning.pdf. [Accessed: 20-Feb-2023].

[43] "No title," Modelzoo.co. [Online]. Available: https://modelzoo.co/model/objectdetection. [Accessed: 24-Feb-2023].

[44] S. Patel, "Diabetic Retinopathy Detection and Classification using Pre-trained Convolutional Neural Networks," International Journal on Emerging Technologies, vol. 11, no. 3, pp. 1082–1087, 2020.

[45] T. Agrawal, Hyperparameter optimization in machine learning: Make your machine learning and deep learning models more efficient. Berkeley, CA: Apress, 2021.

[46] Kumar, R. C. Joshi, M. K. Dutta, M. Jonak, and R. Burget, "Fruit-CNN: An efficient deep learning-based fruit classification and quality assessment for precision agriculture," in 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2021, pp. 60–65.

[47] S. Cheng, K. Zhao, and D. Zhang, "Abnormal water quality monitoring based on visual sensing of three-dimensional motion behavior of fish," Symmetry (Basel), vol. 11, no. 9, p. 1179, 2019.

[48] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[49] International Journal of Computer (IJC), "View of marigold blooming maturity levels classification using machine learning algorithms," ijcjournal.org. [Online]. Available: https://ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/1870/686. [Accessed: 23-Mar-2023].

[50] S. Bondre and U. Yadav, "Automated flower species identification by using deep convolution neural network," in Intelligent Data Engineering and Analytics, Singapore: Springer Nature Singapore, 2022, pp. 1–10.

[51] T.-H. Hsu, C.-H. Lee, and L.-H. Chen, "An interactive flower image recognition system," Multimed. Tools Appl., vol. 53, no. 1, pp. 53–73, 2011

[52] A. Rawat and R. Kaur, "Proposed methodology of supervised learning technique of flower feature recognition through machine learning," Jetir.org. [Online]. Available: https://www.jetir.org/papers/JETIR1907J60.pdf. [Accessed: 23-Mar-2023].

Polarimetric SAR Characterization of Mangrove Forest Environment in the United Arab Emirates (UAE)

SoumayaFatnassi[1], Mohamed Yahia[2], Tarig Ali[3], Maruf Mortula[4]
MACS Laboratory-National Engineering School of Gabes, University of Gabes, Gabes, Tunisia[1, 2]
GIS and Mapping Laboratory, American University of Sharjah, Sharjah, UAE[3]
Civil Engineering Department, American University of Sharjah, Sharjah, UAE[4]

***Abstract*—This Mangrove forests in the United Arab Emirates (UAE) provide valuable ecosystem services such as coastal erosion protection, water purification and refuge for a wide variety of plants and animals. Therefore, the first step toward understanding the mangrove forests is the monitoring of this important ecological system. This paper proposes an original study to characterize the mangrove forest environment in the UAE by using polarimetric synthetic aperture radar (PolSAR) remote sensing. Free access C-band dual- PolSAR Sentinel 1 data have been exploited. The elements as of the covariance matrix as well as the entropy/alpha decomposition parameters have been studied. Results show that the VH intensity, the coherence between VV and VH polarimetric channels, the entropy and alpha angle provide the most pronounced signatures that discern mangrove forests. Thus, these parameters could be exploited to improve the accuracy of the remote sensing monitoring and mapping techniques of mangrove forests in the UAE.**

Keywords—Mangrove forests; dual-PolSAR; sentinel 1; United Arab Emirates; entropy/alpha decomposition

I. INTRODUCTION

Mangrove forests, which appear in the transitional zones between land and sea in most tropical and subtropical coastlines, play a major role in the coastal ecosystem. In the UAE, mangrove forests are mainly located in tidal lagoons with a total extent estimated to be 38km^2 [2]. They are dominated by gray mangroves (i. e. Avicennia Marina) which tolerate water with high salinity and dry weather conditions (see Fig. 1(c) and Fig. 1(d)). To preserve this important ecosystem, a number of approaches have been proposed to monitor and analyze mangrove forests. Studying mangroves using field methods is time consuming, expensive and difficult because of the harsh environment in mangrove ecosystems. Hence, remote sensing has served as a sustainable tool in studies of mangrove forests. A number of methods have been proposed to monitor and analyze mangrove forests using remote sensing data [17], [10], [11]. Regarding the remote sensing data sources, most previous studies can be grouped into two main groups, those employing optical data and those using Synthetic Aperture Radar (SAR) data.

Optical remote sensing data have been widely used for mangrove monitoring due to the availability of very high temporal and spatial resolution imagery [17], [10], [11]. Nevertheless, such systems are limited in utility by the cloud at mangrove sites and by narrow spatial coverage. Few studies have been conducted to map the mangrove forests in the UAE using optical remote sensing data [6], [7], [8].

Synthetic aperture radar (SAR) data have been explored for mangrove forest mapping. SAR offers benefits that include no sensitivity to cloud or precipitation, wide spatial coverage and sensitivity to the geometrical structure of forests (Zhang et al., 2018). In the literature, several studies have been proposed to study the mangrove forests including bio-mass estimation [16], mapping [5], discrimination of species [15], etc. However, there is no study exploiting SAR data in UAE mangrove forest monitoring. The objective of this paper is to fill this gap.

Full polarimetric SAR (PolSAR) data provide much more backscattering information of mangrove forests than single polarization data [9]. Nevertheless, the majority of currently-available SAR data, such as the free-access Sentinel-1 (VV and VH polarizations) and ALOS (HH and HV polarizations) data, are dual (not full) PolSAR. In comparison to full-pol, dual-pol mode is widely used in the radar remote sensing applications due to its high spatio-temporal coverage.

However, little attention has been given in the literature to entropy-alpha-anisotropy polarimetric target decomposition for mangrove forest analyses despite its wide application for the analysis of vegetation polarimetric responses [3]Using L-band ALOS PALSAR full-pol data, it has been demonstrated that Entropy-alpha-anisotropy target decomposition provided valuable measures of scattering mechanisms of the mangrove forest structure [3]. It has been demonstrated that ALOS PALSAR dual-pol entropy-alpha-anisotropy parameters improved the classification accuracies of mangrove species [20].

In this paper, an extended analysis of the dual-pol response of mangrove forest in the UAE is proposed. The dual-pol parameters including the elements of the covariance matrix as well as the entropy/alpha decomposition parameters are studied to derive strong signatures of the mangrove forests. Single look complex VV and VH Sentinel 1 dual-pol data are tested in this study.

This paper is organized as follows: Section II introduces the study area and the experimental data. Section III introduces dual SAR polarimetry. In Section IV, the polarimetric response of mangrove forests is studied. Finally, Section V gives the conclusions of this paper.

II. Study Area and Data

The study area (see Fig. 1(a) and Fig. 1(b)) is located in Ras Al-Khaimah (RAK) in the eastern coastal areas of the United Arab Emirates facing the Arabian Gulf (see zone A Fig. 1(d)). It consists of a coastal forest of Avicennia Marina mangrove trees with high density (zone B Fig. 1(b)) and low density (zone C Fig. 1(b)) with height ranges from a few centimeters to 3–8 m [1], [14] traversed by inundated (zone D Fig. 1(b) and non inundated (zone E Fig. 1(b)) flat saline loamy clay bare surfaces and surrounded by Ras Al-Khaimah city (zone F Fig. 1(b)). This ecosystem is common in mangrove forest sites of the UAE. Hence, the objective of this paper is to characterize the mangrove forests using the studied dual-pol data and to derive the most significant polarimetric descriptors that emphasize the zones B and C (i. e. mangrove forest) from zones A, D, E and F.

For the experimental data, C-band Sentinel-1 images in the interferometric wide Swath (IW) mode have been employed. The data were acquired on 07 October 2021. The VV and VH single look complex (SLC) products with a spatial resolution of approximately 3×20 m (range×azimut) have been considered in order to characterize the mangrove signatures. The data have preprocessed according to [13]. For better speckle filtering while preserving the polarimetric information, the speckle filtering has been performed using the iterative minimum square error (IMMSE) filter [18]with the input parameters (Initial filter: 11×11 Lee sigma filter [12], number of iterations: 7).

III. Dual Polarimetric SAR Descriptors

In dual polarization SAR, one polarization H (horizontal) or V (vertical) is transmitted. Both polarization H and V are received simultaneously. Hence, the number of parameters is lower compared to fully polarimetric data (i.e. transmitting and receiving both H and V polarization simultaneously). The dual-pol data can be represented by 2×2 covariance matrix:

$$C_2 = \begin{bmatrix} \left\langle \left| S_{VV} \right|^2 \right\rangle & \left\langle S_{VV} S_{VH}^* \right\rangle \\ \left\langle S_{VH} S_{VV}^* \right\rangle & \left\langle \left| S_{VH} \right|^2 \right\rangle \end{bmatrix} \tag{1}$$

Where $\langle \; \rangle$ is the averaging operator and * is the complex conjugate. S_{VV} and S_{VH} are the dual polarization complex SAR parameters.

The dual-pol descriptors can be obtained directly from the covariance matrix or derived by applying a decomposition. The most important dual-pol descriptors are the intensity channels which are the diagonal elements of the covariance matrix i.e. $\left\langle \left| S_{VV} \right|^2 \right\rangle$ and $\left\langle \left| S_{VH} \right|^2 \right\rangle$. From the off-diagonal element, the complex coherence between the VV and VH polarimetric channels can be estimated

$$\rho = \frac{\left\langle S_{VH} S_{VV}^* \right\rangle}{\sqrt{\left\langle \left| S_{VV} \right|^2 \right\rangle \left\langle \left| S_{VH} \right|^2 \right\rangle}} = |\rho| e^{j\varphi} \tag{2}$$

Hence two additional dual-pol descriptors which are the coherence $|\rho|$ and the phase difference φ between VV and VH polarimetric channels respectively are considered in this study. The eigen-decomposition of C_2 gives [4]

$$\mathbf{C}_2 = [\mathbf{U}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} [\mathbf{U}_2]^{*\mathbf{T}} \tag{3}$$

Where λ_1 and λ_2 are the eigen-values, U_2 contains the eigenvectors and ^{T} denotes the transpose operator.

The Cloude-Pottier parameters: entropy H and the mean α angle dual-pol descriptors are [4]:

$$H = -\sum_{i=1}^{2} p_i log_2(p_i) \tag{4}$$

$$p_i = \frac{\lambda_i}{\lambda_1 + \lambda_2} \tag{5}$$

Where

$$\alpha = \sum_{i=1}^{2} p_i \alpha_i \tag{6}$$

The mean amplitude of the mechanism is:

$$\lambda = \sum_{i=1}^{2} p_i \lambda_i \tag{7}$$

Fig. 2 to 5 gave the parameters of the studied zone.

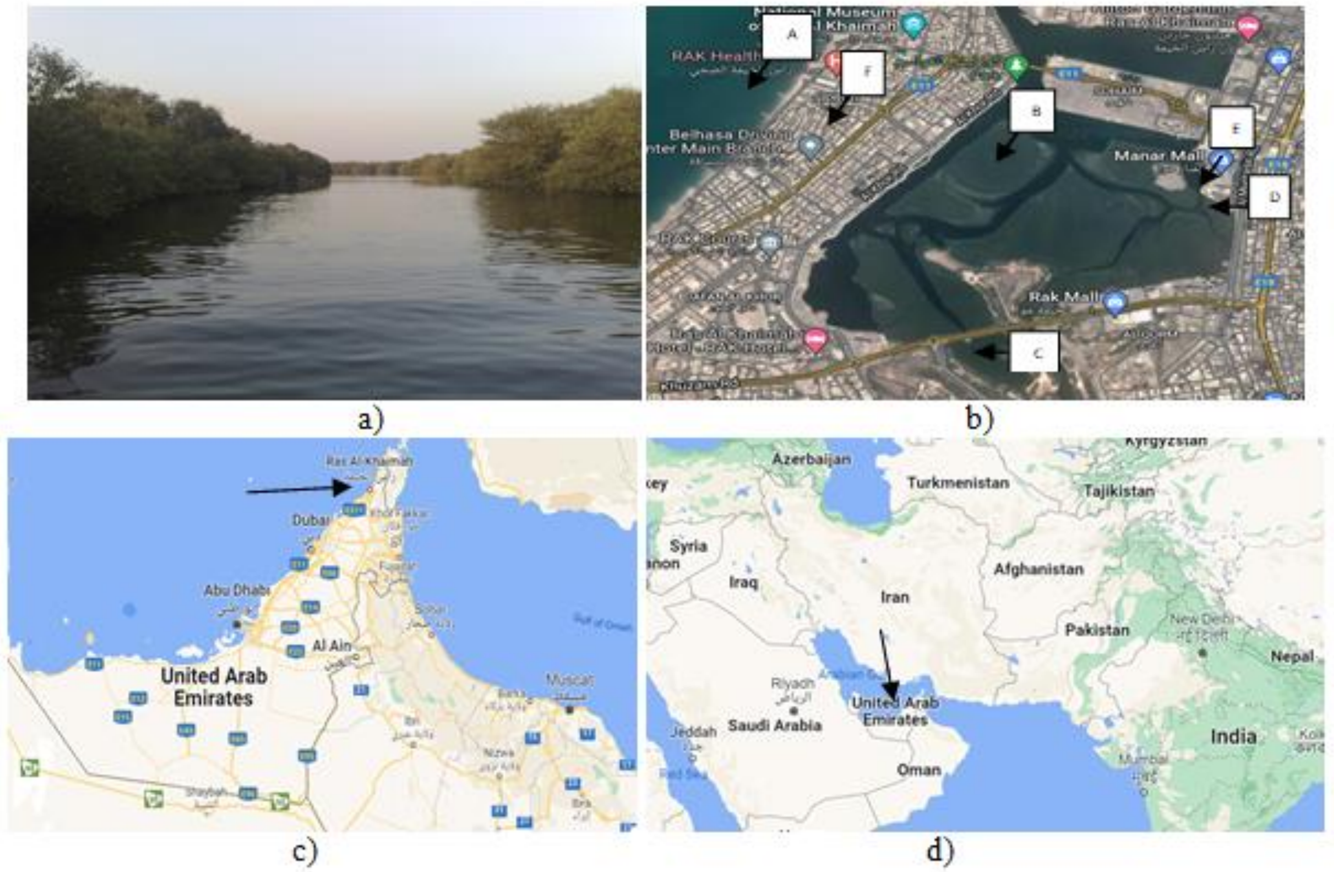


Fig. 1. (a) and (b) Spatial distribution and environment mangrove forests in Ras Al-Khaimah mangrove Greek in the UAE, (c) and (d) The study area (see arrows).

A. VV and VH Intensities

The mangrove forests grow between sea and land areas. They are characterized by a homogeneous crown shape, medium leaf size and arched aerial roots (see Fig. 1(c)). These factors define the roughness, topography, texture, and dielectric constant on which the backscatter, is highly sensitive. Mangrove forests generally are characterized by a volume scattering mechanism (i. e. multiple scattering). Hence, they are highly depolarizing targets. As a result, the VV and VH image exhibit high backscattering intensities (see Fig. 2 zone B). However, in double bounce scattering, one channel does not display high backscattering intensities depending on the orientation of the target. For example for dihedral oriented 45° and dihedral oriented 0°, the VV and VH channels does not exhibit strong backscattering intensity, respectively. These scattering mechanisms occur generally in man-made target such as buildings (see Fig. 2 zone F). Hence, very strong VV and HV intensities (i.e. specular backscattering) are observed in the city zone (see Fig. 2 zone F). In L-band data such those collected from ALOS PALSAR system, high backscattering signal observed in flooded forests originates from double bounce returns between the water surface and forest components can occur [9]. However, in actual C-band Sentinel 1 data, this phenomenon is not pronounced as for L-band since the wave have not the same ability to penetrate inside the mangrove canopy. Hence, this phenomenon is not emphasized as in the city zone where the double bounce is specular. However, double bounce can be noticeable in mangrove trees with low density (see Fig. 2 zone C) and in the boundaries between mangrove forests and flat surfaces (see arrows in Fig. 2).

The surface scattering mechanism does not depolarize the incident wave. The sea and the inundated zone constitute perfect surface reflections. Hence, they show weak backscattering intensities in HH and HV polarizations. The non inundated surface (zone E) is a rough and non-depolarizing target. Thus, it displays strong backscattering intensity in HH polarization only

B. Coherence

The coherence |ρ| measures the degree of linear dependency of VV and VH polarimetric channels. It ranges from 0 to 1. If VV and VH signals are linearly related then |ρ|=1. Fig. 3(a) displays the coherence image of the test site. The mangrove forests are characterized by random scattering mechanisms since they are depolarizing targets. As a result, it can be observed that the zone B (i.e. mangrove forest) displays low coherence values. On the other hand, the A, C, D and E and F zones provide more deterministic scattering (i.e. surface scattering for A, D and E zones and double bounce scattering for C and F zones). As a result, the coherence parameter display high values.

C. Phase Difference

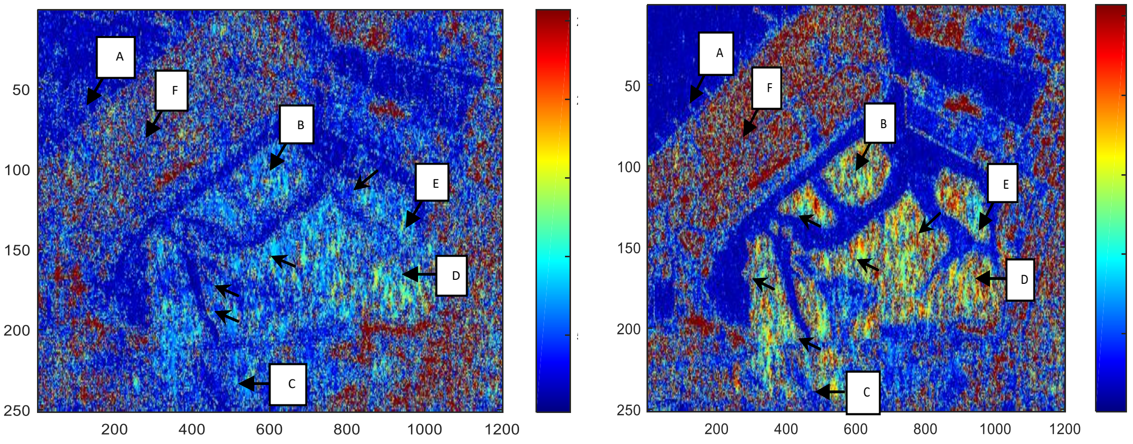
Fig. 4 (a) displays the phase difference image between VV and VH polarimetric channels. It can be observed that this descriptor does not display any particular signature. In fact, this parameter appears noisy and not able to discern the studied zones.

D. H/α

The entropy H is used to characterize the heterogeneity of the media scattering. For deterministic scattering H=0 (e. g. dihedral (i.e. double bounce scattering see Fig. 6) and surface scattering (i.e. simple reflection see Fig. 6) while for random scattering H=1 (e. g. volume scattering see Fig. 6). α angle is an indicator of the type of the scattering mechanism. For surface scattering, α =0°, α =90°for double bounce scattering and α≈50°in dual-pol case (i. e. dihedral scattering such in man-made structures see Fig. 6) for volume scattering such in forests (see Fig. 6).

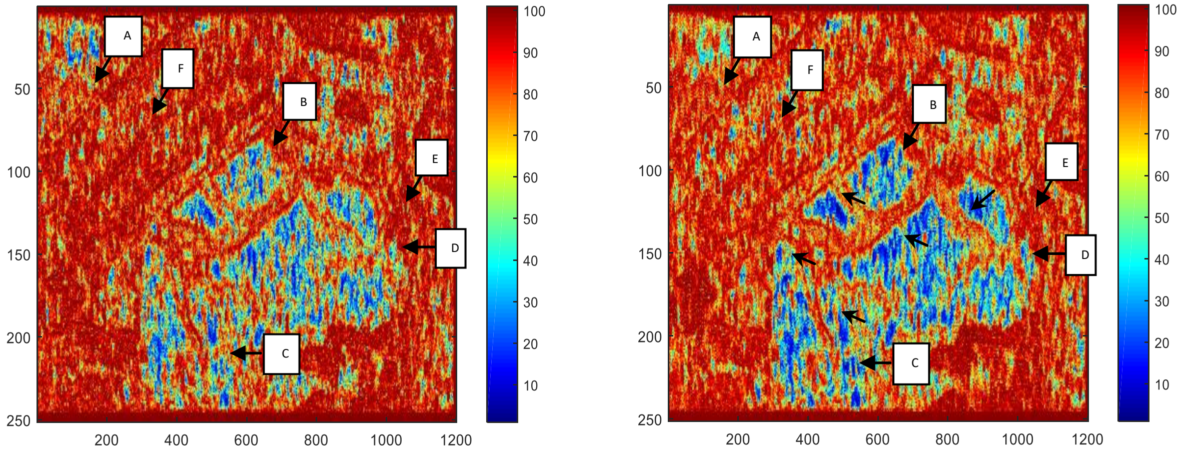
The sea area, the inundated zone and the non inundated zones represent surface scattering. Hence the entropy and alpha angle display relatively low values. The city zone is characterized by double bounce scattering (man-made structures) hence H= 0 and α =90°. Fig. 5 demonstrate that the majority of mangroves zones are characterized by volume scattering (i. e. H=1 and α≈50°). Some isolated points in the mangrove forests are characterized by a double bounce scattering (i. e. H=0 and α=90°) due to the double reflection ground-trunk.

In mangrove forests with low density (Fig. 5 zone C) and the borders between the mangrove forests and the flat surfaces (Fig. 5 arrows), double bounce scattering is also pronounced.



a) b)

Fig. 2. (a) VV intensity image, (b) VH intensity image.



a) b)

Fig. 3. (a) Coherence |ρ| (×100) image (b) 1-H (×100) image.

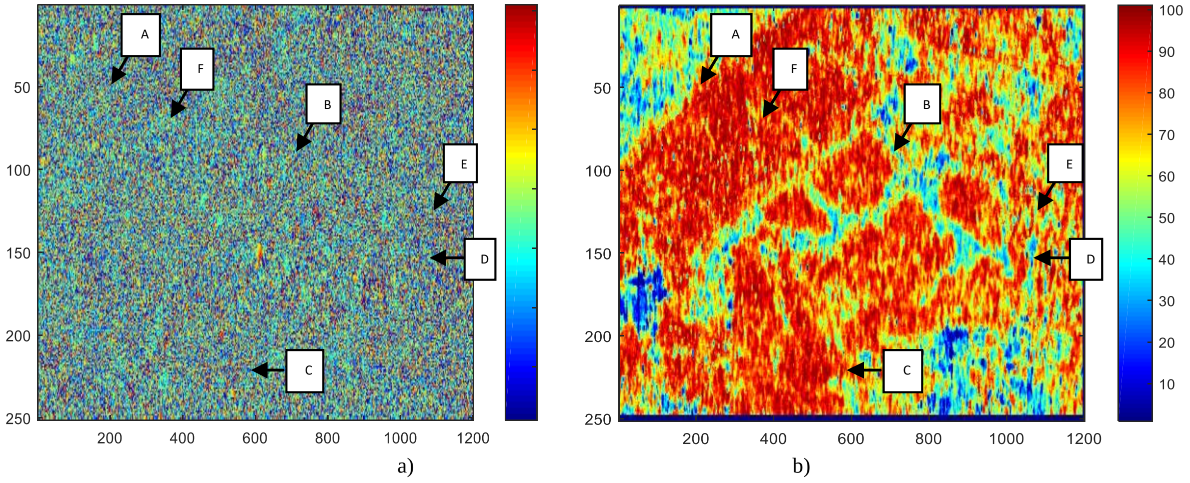


Fig. 4. a) Phase difference φ image, b) Entropy H (×100) image (Lee sigma filter).

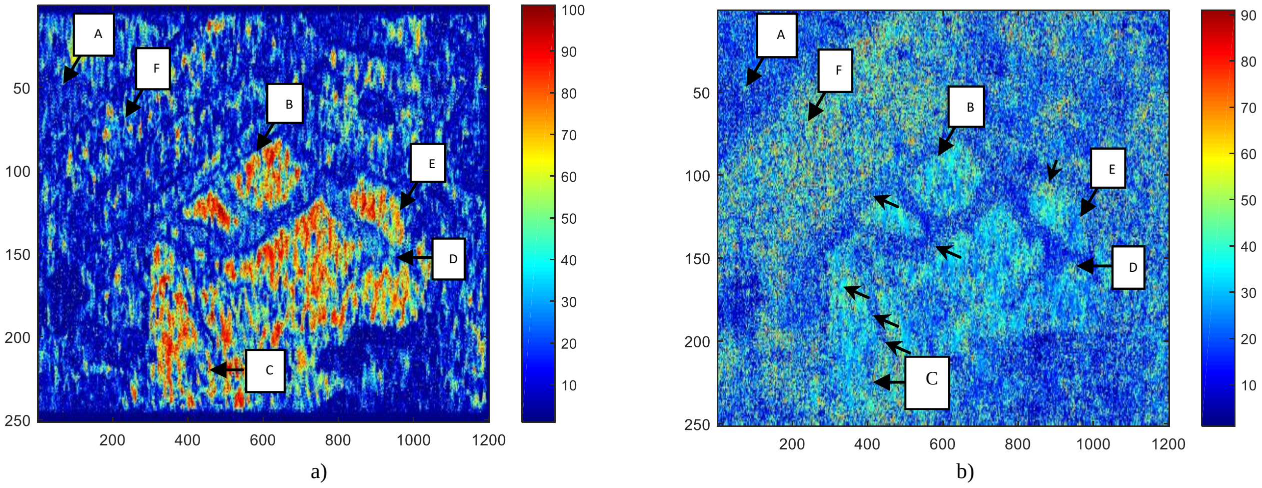


Fig. 5. a) Considered entropy H (×100) image, b) α image.

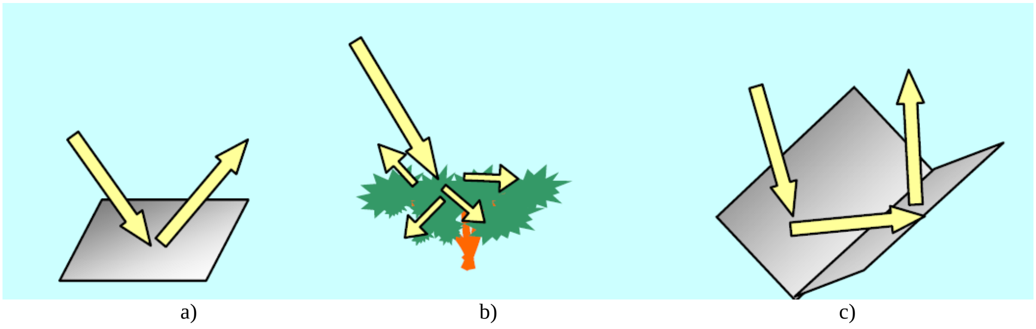


Fig. 6. a) Surface scattering (e.g. plate surface) H=0 and α =0°, b) volume scattering (e. g. random scattering in forests) H=1 and α =45°(in PolSAR case), c) double bounce scattering (e. g. dihedral) H=0 and α =90°.

E. Mean of Eigenvalues λ

The mean of eigen-values λ is an intensity parameter as VV and VH intensities. Fig. 7 displays the λ image. It can be observed that zone A and D zones display low values. The city zone d display very high values. B zone and D zone show high λ values.

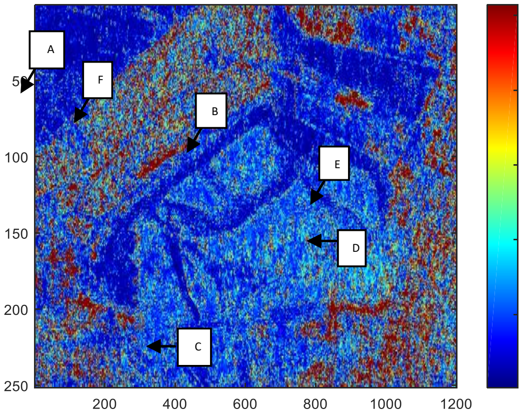


Fig. 7. λ image.

IV. Discussion

The polarimetric characterization of the mangrove forest environment of the UAE shows that the HV intensity image is more able to discern the mangrove forest than the VV intensity image. Since the mangrove forests are highly depolarizing targets, the VH image better discern their extent. Hence, since the inundated zones between the mangrove regions are no-polarizing zones, they are better described. The open area (E) is also a non-polarizing target. Hence, it is better emphasized from the mangrove trees. The coherence image emphasizes also the mangrove forest zones since all the behavior displayed low coherence values except the mangrove zones which are characterized by random scattering. The environment of the mangrove forest is non-depolarizing (H=0) whereas dense mangrove trees are highly depolarizing. Hence, the entropy parameter is a powerful indicator to discern dense mangrove trees. However, the entropy parameter cannot emphasize double bounce generated by ground-trunk scattering and surface scattering generated by the inundated and non inundated zones since H=0. It is interesting to observe that the entropy H and the coherence |ρ| are highly correlated since 1-|ρ| image provided practically the same information as H image (see Fig. 3). Hence, only the entropy can be kept to characterize the mangrove forest environment.

In the environment of mangrove forests in the UAE, only dense mangrove zones are characterized by a volume scattering. Hence, alpha angle is able to emphasize this area. However, in high entropy zones, the alpha angle cannot provide additional information. Alpha angle is generally significant in medium and low entropy values. In fact, contrary to the entropy parameter, alpha angle is able to discern double reflection from simple reflection (i. e. H=0 for both reflections whereas alpha =0 for simple reflection and alpha =90° for double reflection see Fig. 6). This phenomenon is observable mainly at the borders between mangroves and flat surfaces which present double bounce reflections (see arrows in Fig. 5 (b)). It is also interesting to observe that in low density mangrove forests double bounce reflection is dominant (see zone C in Fig. 5(b)). Concerning λ image, it can be observed that this descriptor does not introduce additional information to VV and VH intensity image.

In conclusion, the combination of polarimetric descriptors HV intensity, entropy (or coherence) and alpha angle is able to emphasize the mangrove forests of the UAE (with high and low density) from their environment. It is important to notice that the choice of the polarimetric filter it crucial. In fact, Fig. 4 (b) displays the entropy image of the test site after applying the 7×7 Lee sigma filter (using PolSAR-pro software). It can be observed that the filter is not able to preserve the polarimetric information in the city zone which displays high entropy. This phenomenon is due mixing of different scattering media [19]. As a future research direction, we propose to use the entropy and alpha angle parameters in segmentation and classification algorithms of mangrove forests in UAE. We could also use other types of decomposition for the characterization of polarimetric images in the mangrove forest environment in the UAE.

V. Conclusion

In this paper, the environment of mangrove forests in the UAE has been studied using free access C-band dual- PolSAR Sentinel 1 data. Among the studied polarimetric descriptors, results show that the VH intensity, the coherence between VV and VH polarimetric channels, the entropy and alpha angle provide the most pronounced signatures that discern mangrove forests. Thus, these parameters could be exploited for further applications related to mangrove forests in the UAE.

Acknowledgment

The authors would like to thank the Smart City Research grant from American University of Sharjah, UAE, Grant, FRG21-M-E77 and ESA for providing PolSAR data.

References

[1] Alsumaiti T S (2014) An assessment of avicennia marina forest structure and above ground biomass in eastern mangrove lagoon national park, AbuDhabi. Arab WorldGeogr 17:166–185.

[2] Blasco F, Carayon J L, Aizpuru M (2001) World mangrove resources. Glomis Electronic Journal 1:1-3.

[3] Brown I, Mwansasu S, Westerberg L O (2016) L-band polarimetric target decomposition of mangroves of the Rufiji Delta, Tanzania. Remote Sensing 8:, 140–. doi:10.3390/rs8020140 .

[4] Cloude S, Pottier E (1996) A review of target decomposition theorems in radar polarimetry. IEEE Trans. Geosci Remote Sens 34: 498–518. DOI: 10.1109/36.485127.

[5] de Souza Pereira F R, Kampel M, Cunha-Lignon M (2012) Mapping of mangrove forests on the southern coast of São Paulo, Brazil, using synthetic aperture radar data from ALOS/PALSAR. Remote Sens Lett 3 : 567–576.DOI:10.1080/01431161.2011.641511.

[6] Elmahdy S, Ali T, Mohamed M, Howari F M, Abouleish M, and Simonet D (2020) Spatiotemporal mapping and monitoring of mangrove forests changes from 1990 to 2019 in the northern Emirates, UAE using random forest, kernel logistic regression and naive Bayes tree models. Front. Environ. Sci 8: 102. DOI:10.3389/fenvs.2020.00102.

[7] Elmahdy S, Mohamed M M (2018) Monitoring and analysing the Emirate of Dubai's land use/land cover changes: an integrated, low-cost

remote sensing approach. Int. J. Digital Earth 11: 1132–1150. DOI:10.1080/17538947.2017.1379563.

[8] Elmahdy S, Mohamed M M (2013) Change detection and mapping of mangrove using multi-temporal remote sensing data: a case study of Abu Dhabi, UAE. J. Geomatics 7: 41–45.

[9] Ferrentino E, Nunziata F, Zhang H, Migliacci M (2020) On the ability of PolSAR measurements to discriminate among mangrove species. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 13: 2729 – 2737. DOI:10.3390/rs11080921.

[10] Heumann B W (2011) Satellite remote sensing of mangrove forests: Recent advances and future opportunities. Prog Phys Geogr 35: 87–108.DOI:10.1177/0309133310385371.

[11] Kuenzer C, Bluemel A, Gebhardt S, Quoc T V, Dech S (2011) Remote sensing of mangrove ecosystems: A review. Remote Sens 3: 878–928. DOI:10.3390/rs3050878.

[12] Lee J S, Ainsworth T, Wang L Y, Chen K S (2015) Polarimetric SAR speckle filtering and the extended sigma filter. IEEE Trans Geosci Remote Sens 53: pp. 1150–1160. DOI: 10.1109/TGRS.2014.2335114.

[13] Mandal D, Vaka D S, Bhogapurapu N R, Vanama V S K, Kumar V, Rao Y S, Bhattacharya (2019) A. Sentinel-1 SLC preprocessing workflow for polarimetric applications: A generic practice for generating dual-pol covariance matrix elements in SNAP S-1 toolbox. Preprints, 2019110393. DOI: 10.20944/preprints201911.0393.v1.

[14] Moore G E, Grizzle R E, Ward K M (2013) Mangrove resources of the United Arab Emirates: mapping and site survey 2011–2013,"in Final Report to the United Arab Emirates Ministry of Environment and Water, University of New Hampshire, (Durham, NC: Jackson Estuarine Laboratory).

[15] Pham T D, Bui D T, Yoshino K, and Le N N (2018) Optimized rule-based logistic model tree algorithm for mapping mangrove species using ALOS PALSAR imagery and GIS in the tropical region. Environ. Earth Sci 7.DOI:10.1007/s12665-018-7373-y.

[16] Rauste Y, Hame T, Pullianen J, Heiska K,Hallikainen M (1994) Radar-based forest biomass estimation. Int J Remote Sens 15: 2797–2808.

[17] Wang L, Jia M, Yin D, Tian J (2019) A review of remote sensing for mangrove forests: 1956–2018. Remote Sens Environ 231 : 111223.

[18] Yahia M, Ali T, Mortula M M, Abdelfattah R, Elmahdy S (2020) Polarimetric SAR speckle reduction by hybrid iterative filtering. IEEE Access 8.DOI:10.1016/j.rse.2019.111223.

[19] Yahia,M; Aguili,T. Characterization and correction of multilook effects on eigendecomposition parameters in PolSAR images IEEE Trans Geosci Remote Sens 2015 53:5237-5246.DOI:10.1109/TGRS.2015.2419717.

[20] Zhang,H;Wang,T; Liu,M; Lin,H; Chu,LM; Devlin,AT. Potential of combining optical and dual polarimetric SAR data for improving mangrove species discrimination using rotation forest. Remote sensing 2018 10:467. DOI:10.3390/rs10030467.

An In-depth Analysis of Uneven Clustering Techniques in Wireless Sensor Networks

Hai-yu Zhang
School of Finance and Economics, Taiyuan University of Technology, Taiyuan 030024, China

***Abstract*—The low-cost and convenient feature of Wireless Sensor Networks (WSNs) has made them popular in many sectors over the last decade. The WSNs are now widely used as a result of recent advancements in low-power communication and being energy-efficient. The WSNs typically use batteries to power sensor nodes. The finite stored energy in batteries and the hassle of battery replacement have led to a critical focus on energy efficiency for WSNs. Clustering and data aggregation are the most efficient methods to address the energy concerns of WSNs. This paper comprehensively reviews several uneven clustering methods and compares the various uneven clustering algorithms. The methods are described in terms of their goals, attributes, categories, advantages and disadvantages. Probabilistic clustering is used when there is a need of simplicity and speed. As a result, this study compared all these types of protocols based on their clustering properties, CHs properties, and on the type of clustering process; and current research gap effective techniques are also addressed.**

Keywords—Wireless sensor networks; data aggregation; uneven clustering; energy-efficient; review

I. Introduction

Wireless Sensor Networks (WSNs) can be distinguished from other wireless ad-hoc networks by their unique characteristics. WSN nodes are less powerful devices that have less memory and processing power [1, 2]. Recent advances in CMOS and nanotechnology have made sensor nodes smaller, cheaper, and more efficient computationally [3]. Sensor nodes are clustered together with microprocessors, batteries, transceivers, and sensors. Numerous sensor nodes can be connected in a network without requiring any predefined infrastructure, and the sensor nodes can configure themselves [4]. Sensor nodes in the WSN detect physical variables such as humidity, temperature, and pressure in the air. Furthermore, it can detect acoustic signals, infrared, and vehicle movements [5]. The micro-sized processing unit processes the value sensed by these sensors and forwards it to the base station with the help of a communication unit through one-hop or multi-hop transmission [6].

Many applications use WSNs to monitor and track, including surveillance in the military, healthcare, industrial automation, agriculture, inventory control, disaster management, etc. WSNs provide connectivity for remote regions, such as deserts, forests, deep seas, battlefields, and complex industrial and research facilities [7]. Sensor nodes are deployed in remote provinces with limited electricity for charging and other replacement options. The initial phase of transmitting data from the source nodes to the base station is more costly than sensing data in WSNs [8]. Therefore, an energy-efficient communication strategy is crucial in transmitting data from the sensor nodes to the base station, which will prolong the lifetime of the WSN. These problems can be resolved with the help of a well-known method called clustering [9].

Using the clustering strategy, different nodes are grouped into heterogeneous and homogeneous clusters. In general, clustering involves designating a single node as the CH and segmenting a given area into discrete sectors. In order to create an energy-efficient WSN model, CH is essential. To enhance network performance, the CH may be altered in practice during some iterations [62]. Furthermore, unique specialized nodes, such as CHs, make all significant choices on behalf of SNs with the aid of clusters. Clustering can be categorized as either single-hop or multi-hop. There are many drawbacks and benefits to single-to-multi hop communications, such as the energy losses associated with single-hop transmission due to greater range. In the meantime, the multi-hop technique can boost energy consumption to solve the wireless sensor network's scalability issue, thereby boosting both energy consumption and scalability [61, 63]. There are two types of sensors in these clusters: normal nodes and Cluster Head (CH) nodes [10]. Fig. 1 depicts the clustering architecture of WSNs. Communication in WSNs is divided into two types: intra-cluster and inter-cluster. Intra-cluster communication occurs between cluster members, whereas inter-cluster communication occurs between clusters. The cluster member nodes sense real-world parameters, and the measured values are transmitted to the CH. The CHs combine the values received to eliminate redundant information present due to the same values sensed by some nodes. The final data is transmitted directly to the base station through intermediary CHs [11]. Cluster members can send data to CHs only, and CHs can send data to the base station, eliminating the need to transmit redundant data. Clustered WSNs have many advantages over conventional WSNs, including better utilization of bandwidth, less overhead, enhanced link connectivity, efficient topology balancing, stability in network topology, and reduced routing table size [12].

In the last few decades, a variety of even and uneven clustering methods have been presented for enhancing energy efficiency in WSNs. The motivation behind this work is the lack of a thorough review of clustering protocols. The main contribution of this study is to do critical analysis and review outcomes of uneven clustering and provide a detailed analysis of each method. Furthermore, this study compares clusters based on a variety of cluster properties, including cluster properties and clustering processes. This paper is organized as follows. Section II discusses the characteristics and objectives of uneven clustering. The classification and explanation of clustering algorithms are presented in Section III. Section IV categorizes and tabulates algorithms. The paper concludes in Section V.

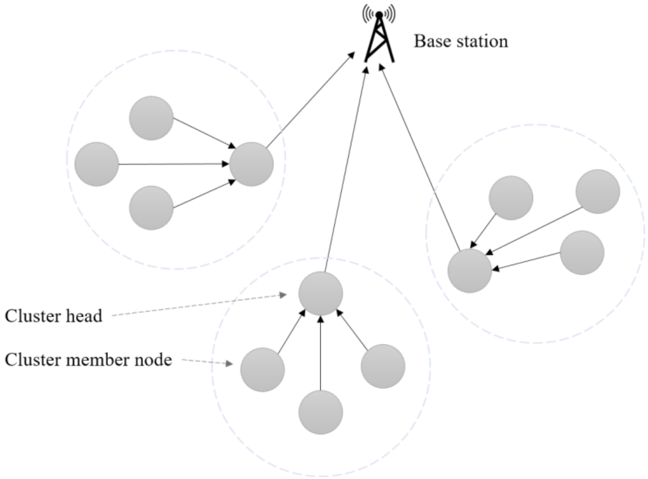


Fig. 1. Clustering architecture of WSN.

II. Background

In this section, an overview of uneven clustering is presented, as well as a detailed discussion of its objective and characteristics.

A. WSN Applications

The use of WSNs is expanding rapidly in various applications or is still in its infancy. There are four general categories of WSN applications according to their use, including; urban, industrial, environmental, and health. There are various subgroups within each category. Each category and subcategory in this section is described in more detail. Examples of their characteristics are discussed, highlighting their advantages and disadvantages.

- Urban applications: WSNs offer a wide range of sensing capabilities, allowing them to gather extensive information about any target area, indoors, outdoors, or even inside buildings. In an urban environment, WSNs can measure any phenomenon's spatial and temporal characteristics. Monitoring various parameters is crucial for ensuring citizens' optimal well-being in large cities [13]. WSNs provide authorities with real-time data needed for optimal city operation. Specifically, increasing the number of vehicles heading toward common destinations creates problems and time waste. WSNs can track traffic to reduce congestion, indicate parking spots, etc. Various systems can improve human lives in the era of informatics. WSNs also apply to indoor environments, like smart homes, which involve machine-to-machine connections. Indoor air quality monitoring and localization of motion are two examples [14]. Keeping public roads safe and functional depends on efficient vehicular transportation. In the transportation domain, various WSN-based systems have been proposed by researchers. Drivers can receive information from these systems during their use of roads, which can improve their safety on the road. Increasing demands for safety and security in cities are driving the use of WSNs in structural health monitoring. Thanks to the rapid advancement of wireless technologies, structural monitoring systems have been significantly improved by combining WSN technology. WSN-based structural health monitoring systems offer compelling advantages over traditional wired systems, including reduced installation and maintenance costs [15].

- Industrial applications: WSNs have been used in a variety of applications in industrial settings in recent years. In critical infrastructure, they are incorporated into supervisory control and data acquisition systems. In conjunction with IoT applications, these systems enable smart monitoring and control. Using WSNs in industrial automation systems also includes controlling and monitoring industrial equipment. WSN requirements for industrial wireless systems differ from general WSNs [16]. Incorporating sensors into designs, floor plans, manufacturing equipment, and other critical areas benefits industries greatly. Industrial application of WSNs reduces failure rate and enhances the effectiveness and efficiency of factory operations. Integration of wireless technology in the industrial sector requires security and reliability. Insecure routing protocols, hostile deployment locations, and open architecture render industrial WSNs subject to varied attacks. The limited processing, memory, range of communication, and energy of WSNs makes them easy targets for adversaries. Providing secure and reliable services, WSNs are a reliable and suitable technology for factories and industries [17].

- Environmental applications: WSNs provide environmental information to decision-makers. Environmental monitoring encompasses both natural and man-made environments, indoors and outdoors. There are various methods of monitoring, such as sensor-based monitoring on the ground, field sampling analysis in laboratories, and aerial and satellite remote sensing [18]. Diverse challenges remain for environmental monitoring nodes. Environment-specific WSNs are typically located in remote areas disconnected from the power grid. The primary challenge lies in selecting proper topologies and operating strategies aiming to ensure the nodes are as energy efficient as possible [19].

- Health applications: Having many advantages, like short delay, superior performance, economical cost, etc., WSNs are widely used in the healthcare sector. The healthcare WSN involves the real-time transmission of patient health information through the Internet to health professionals, so patient confidentiality and integrity are crucial. Physiological parameters are detected by placing sensor nodes on patients. This allows the monitoring center to track the patient's vital signs in real-time remotely [20]. The monitoring center receives the information and processes it on time. WSNs allow the collection of human health data, which is helpful when researching human diseases and conditions. A wide range of applications of WSN can also be found in drug management and drug development, blood management, etc. Future telemedicine monitoring systems can take advantage of WSNs more conveniently and cost-effectively [21].

B. Uneven Clustering

Clustering can be even or uneven. Compared to uneven clusters, even clusters have equal sizes. Network partitioning occurs when CHs near the base station die before they mature. The problem is commonly known as the energy hot-spot issue [22]. A disparity in power consumption results in hot spots in WSNs, decreasing the network's life expectancy [23]. As a solution to hot-spot unevenness, clustering techniques are used for load balancing among the CHs. In uneven clustering, cluster sizes near the base station are smaller than those far from it. This way, uneven clustering solves the load balancing or hot-spot problem in clustering [24]. Fig. 2 illustrates how clusters of uneven type are formed based on competition radius calculations. Eq. (1) gives the formula to calculate the competition radius of nodes that are homogeneous in type and size, where *c* is the weighted factor ranges between 0 and 1, R_0^c refers to the highest competition radius value, d(s_i, Ds) denotes the distance between the node s_i and base station, and d_{min} stands for minimum distance between the node and base station.

$$R_c = \left[1 - c\,\frac{d_{max} - d(s_{1,DS)}}{d_{max} - d_{min}}\right] R_c^0 \qquad (1)$$

In a network with nodes whose energy levels differ, i.e., heterogeneous type nodes, the formula for competition radius is given below. In Eq. (2), α and β are weighted factors having a value between 0 and 1, E_{max} refers to the maximum value of residual energy, R_{max} refers to the maximum competition radius value, and E_r signifies the remaining energy of node s_i.

$$R_{c=}\left[1 - \alpha\,\frac{d_{max} - d(s_{i,}DS)}{d_{max} - d_{min}}\,\beta\left(1 - \frac{E_r}{E_{max}}\right)\right] R_{max} \quad (2)$$

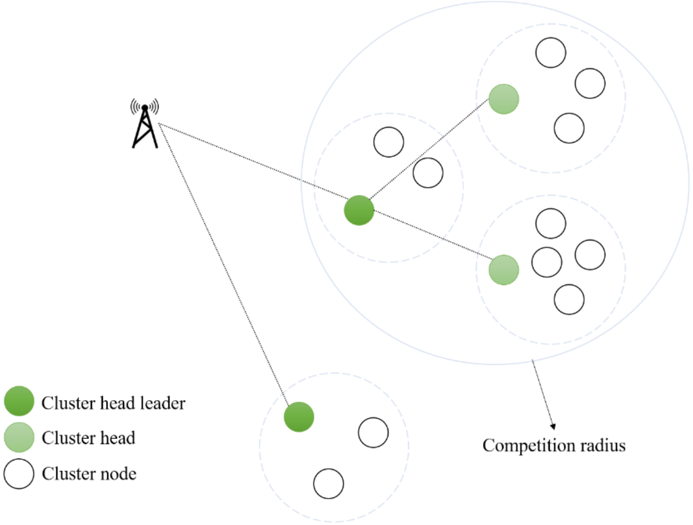


Fig. 2. Competition radius.

The intra-cluster communication does not require any aggregation of data, but the inter-cluster communication requires aggregation of data by the CHs before transmitting to the next node or directly to the base station. The parameter Erelay represents the energy consumed in the data transfer from node s_i to node s_j, and its calculation is given in Eq. (3).

$$Erelay = d^2(S_i + S_j) + d^2(S_i, DS) \qquad (3)$$

This requires that the node S_i have to select the next node S_j who have the highest remaining energy and minimum E_{relay} in the route candidate node set.

C. Objective

In uneven clustering, the main objective is to avoid hotspots, which arise from unequal energy consumption by the nodes of WSNs, thereby reducing their lifespan. The remaining objectives of uneven clustering are the same as those of even clustering. Additionally, uneven clustering aims to achieve the following.

- Load balancing: Uneven clustering provides load balancing by dividing networks into clusters of variable sizes, with smaller clusters closer to the base station and larger clusters located farther from the base station. Energy can be saved in inter-cluster data transmission, extending the network life [25, 26].

- Network lifetime: Uneven clustering has the primary purpose of enhancing the network's lifetime, which is essential for all the real-time applications that require energy conservation among the nodes of the WSNs due to the limited amount of energy available. A good routing algorithm and rotating CPUs can be used to conserve energy with uneven clustering [27, 28].

- Data aggregation: To reduce the redundancy caused by the same value sensed by several node members, CHs aggregate the received values. The final data is sent to the base station directly or indirectly through midway CHs, which minimizes the number of messages sent and minimizes the load on the network as a whole. Data fusion or data aggregation refers to a process combining all incoming packets. Data fusion enhances data and reduces unnecessary noise [29].

- Fault-tolerant: Sensor nodes are positioned in remote regions, such as deserts, forests, deep sea, battlegrounds, and complex industrial and research setups, where there may be the possibility of them being damaged or malfunctioning. The applications that may suffer causalities from data loss require fault-tolerant nodes. The self-organized WSNs solve this problem by re-clustering, but re-clustering not only interrupts the current process but also increases overhead [24, 30].

- Scalability: Sensor nodes ranging from hundreds to thousands are positioned in real-time according to application requirements. The routing method should be designed to handle these vast numbers of sensor nodes. The transmitting node should know the receiving CH when cluster data is exchanged using cluster heads. Hierarchical clustering allows networks to be subpartitioned into layers, and each layer can be subpartitioned into clusters that facilitate scalability and minimize routing table size [31].

- Stable network topology: In any type of clustering, sensor nodes are elected as CHs, and CHs are subject to changes in topology at the cluster level. CHs are equipped with all the information about their members' nodes, including their location, energy level, and ID. Hierarchical clustering is effective at the level of managing the topology of the networks. Changes in cluster membership should be reported to the base station, as re-clustering is required to maintain the effectiveness of the topology [32].

D. Uneven Clustering Characteristics

1) Clusters properties: Cluster properties include cluster size, cluster count, and types of communication within and between clusters.

- Size: Clustering can be even or uneven. The even clustering results in the same size of clusters. However, in uneven clustering, the clusters surrounding the base station are smaller, and the others are larger to balance the load. Hence, energy can be conserved in the inter-cluster data transfer, thereby extending network life [33].

- Count: Depending on the application, clusters may be fixed or variable in size. In some applications, the total number of CHs is 5%, while in others, CHs are selected by random elections [34].

- Inter and intra-cluster communication: Within WSNs, there are two types of communication: inter and intra-cluster. Within a cluster, intra-cluster communication occurs between members and CHs. Inter-cluster communication takes place between CHs [35].

2) Cluster heads properties: By single or multi-hop communication, CHs collect data from members, aggregate the collected data, and send the aggregated data to the base station.

3) Method of clustering properties: The clustering method has four basic characteristics, which are as follows:

- Objective: Clustering serves several objectives, such as energy conservation, fault tolerance, load balancing, and enhancing the lifespan of WSNs [36].

- Methods: Clustering or grouping of nodes can be centralized or distributed. A centralized approach controls cluster formation, selection of CHs, etc., whereas a distributed approach does not have central control [37].

- Nature: Clustering can take a reactive, proactive, or hybrid approach. Continuous transmission of data indicates that nature is proactive. In reactive nature, data is dispatched when sensed data has reached a predefined threshold. The term hybrid refers to the combination of proactive and reactive nature according to the requirements of specific applications [38].

- The base of selecting CH: CHs are selected according to three criteria: attribute-based strategies, probabilistic strategies, and preset. Probabilistic strategies are those in which CHs are chosen randomly without predetermined protocols. Attribute-based strategies determine CHs depending on various indicators such as remaining energy, distance from base stations, the density of nodes, etc. In the present method, the CHs are elected before the nodes or clusters are deployed [39].

III. Review of Uneven Clustering Algorithms

Uneven clustering is intended to prevent the hot-spot problem arising from the unequal distribution of energy consumption by the nodes of WSNs, reducing the network's life span. Uneven clustering accomplishes the same objectives as even clustering. Uneven clustering algorithms are categorized into three categories: probabilistic, deterministic, and preset. Additionally, "static, dynamic, and hybrid" classes are used to categorize the clustering protocols. Clusters in static clustering, once formed, remain the same for the duration of the network. The benefit of this approach is a decrease in clustering overhead. However, the static method might not work properly since some nodes might run out of energy and shut down before other nodes do. Dynamic clustering, in contrast, differs from the static technique in that it will carry out fresh clustering at each instance. Even so, it has a drawback of expensive overhead and won't generate much traffic to put strain on the nodes. The last benefit of hybrid clustering is that it decreases communication overhead while increasing energy efficiency and network longevity [64]. Following sections review the existing uneven clustering methods.

A. Preset Uneven Clustering Approaches

Preset clustering algorithms are used to design WSNs in which the locations of CHs are known in advance. Although WSN topologies change due to node or link failures, preset clustering algorithms ignore these conditions. Therefore, real-time applications cannot be implemented with these algorithms. The proposed approach by Soro and Heinzelman [40] organizes the network hierarchically with Unequal Clustering Sizes (UCS), distributing the energy uniformly among the CHs and extending the lifespan of the network. The positions of each CH are determined in advance and are arranged in a circle around the base station. Each cluster consisted of sensor nodes arranged in a Voronoi pattern around a CH, forming layers around the CH with a particular number of clusters. The Voronoi regions are pie-shaped areas. First layer clusters are equal in size, forming a symmetrical circular association. However, the clusters in the second layer differ in shape and size from the first layer. In every layer, the radius can be adjusted to vary the region that clusters cover, thereby changing the number of nodes within each cluster. According to experimental results, UCS is approximately 10%-30% more efficient than the even-sized cluster scheme that depends on the effectiveness of CH aggregation. UCS can work better for networks that collect a lot of data. This UCS has a long lifespan for heterogeneous and homogeneous networks with a static cluster. The UCS also achieved more even dissipation in uniform-type networks. A comparison of preset-type uneven clustering methods is provided in Table I.

B. Deterministic Uneven Clustering Approaches

Deterministic clustering algorithms control clusters and select CHs efficiently. In these algorithms, the CHs are selected by considering several factors, including available energy, the proximity of the base station, the density of nodes, etc. These criteria change as neighboring nodes exchange messages. Deterministic clustering algorithms can further be divided into three groups, compound-based, heuristic-based, and fuzzy-based.

1) Compound-based approaches: In unequal clustering, various metrics, such as linked graphs, the Sierpinski triangle, etc., are used to make the clusters compound-based. Numerous deterministic compound-based approaches have been suggested to minimize the energy consumption of clustering. As a solution to the hot-spot issue in WSNs, Guiloufi, and Nasri [41] proposed Energy Degree Distance Unequal Clustering Algorithm (EDDUCA). The CHs in EDDUCA are chosen based on reserved energy, node density, and distance between nodes and cluster centers. In addition, the size of clusters made by EDDUCA is the same regardless of their distance from each other and the base station. EDDUCA applies the Sierpinski triangle procedure to separate networks into uneven clusters based on their size. In the Sierpinski triangle procedure, n nodes are homogeneously distributed in the square shape zone. Two diagonal lines are drawn in the rectangle-type zone, and four triangles are drawn. Small triangles are then formed by linking the midpoints of the sides of each triangle. The results of the experiments showed that EDDUCA significantly reduced the consumed energy, balanced the consumed energy among clusters, and balanced the consumed energy among sensor nodes, thereby extending the longevity of wireless sensor networks.

Xia, and Zhang [42] have developed the Unequal Clustering Connected Graph Routing Algorithm (UCCGRA) to load balance between CHs of clusters and diminish the energy consumed in inter-cluster communication. Uneven clustering and connected graph theories are the basis of UCCGRA. Firstly, CHs are chosen by a vote to balance energy consumption. Because they are smaller and have more load than CHs, located far from the base station, CHs near the base station can maintain some power for inter-cluster communication. A multi-hop inter-cluster routing scheme is introduced based on the connected graph of CHs and base stations that allow multiple ways to reach the base station from CHs. Results prove that UCCGRA enhances the reliability and flexibility of data communication. It also prolongs the lifespan of a network by evenly distributing the CHs, balancing the scale, reducing energy consumption among CHs, and optimizing energy consumption among CHs.

Guo, Zhang [43] have proposed PEG-ant based on the PEGASIS protocol and used the improved ACO method instead of greedy algorithms. Each neighbor of the current node is considered a candidate, and residual energy, energy consumed, and pheromone quantity are considered factors while selecting the next node. In comparison to the original PEGASIS, PEG-ant achieved global optimization. Through the chain made by the PEG-ant, the path is more evenly distributed, and the total communication distance is also reduced. The chain nodes are balanced based on residual energy. Compared to existing protocols, PEG-ant significantly prolonged the lifespan of WSNs and effectively managed energy consumption.

2) Heuristic-based approaches: The uneven heuristics clustering in WSNs uses optimization algorithms to provide the optimal solution. To achieve the most optimal performance or solution, each algorithm uses a different fitness function. These methods are generally centralized but may be distributed in exceptional cases. There are various Deterministic heuristic-based clustering methods proposed to make this process energy-efficient.

Data aggregation and clustering based on tree structures can grant WSNs a longer lifespan. Kaur and Mahajan [44] offerd a hybrid approach based on Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms called ACOPSO. The first step is identifying CHs, followed by the shortest distance communication process to gather sensed information from CHs. CHs are connected to sinks by the shortest path calculated by ACOPSO. ACOPSO used comprehensive sensing to reduce the packet size in WSNs. The simulation results demonstrate that the proposed protocol prolongs the network life span compared to GSTEB based on stability measurements. In this system, all nodes are arbitrarily distributed in an area of 100 by 100 square meters, with the base station located 50 meters apart.

Sabor, and Abo-Zahhad [45] developed the Unequal Multi-hop Balanced Immune Clustering (UMBIC) approach to enhance the lifespan of heterogeneous/homogeneous WSNs with unequal densities at small and large scales. To manage both inter- and intra-cluster energy consumption, UMBIC uses Uneven Clustering Methods (UCM) and a Multi-objective Immune Algorithm (MOIA). In UMBIC, the UCM divides the network into clusters of different sizes based on the distance to the base station and remaining energy. With MOIA, the most advantageous clusters are identified, and a routing tree is created between them, ensuring connectivity between nodes and reducing communication costs. The UMBIC protocol enhances the lifespan of the networks, eliminates hotspots, and optimizes energy consumption. Additionally, UMBIC reduces overhead and simplifies computation since the base station gathers information only once.

A novel chemical reaction optimization (nCRO) paradigm is used by Srinivasa Rao and Banka [46] to propose unequal clustering and routing algorithms. The nCRO-UCRA method was developed to resolve the hot-spot problem using the nCRO method, which is a novel chemical reaction optimization. To accomplish this, nCRO-UCRA formulates linear programming for two main issues: multi-hop routing and the choice of CHs. The network is divided into uneven-type clusters. CHs are selected on the basis of a derived cost function. Finally, nCRO-based routing is applied to select a next-hop CH based on the degree of each node and distance. The encoding of molecular configurations and the potential energy function of novel types are used to implement these methods. Experiments have demonstrated that the nCRO-UCRA outperforms existing technologies such as EPUC, PSO, EAUCF, and EBUC regarding remaining energy, the number of alive nodes, network lifespan, and the average packet delivery ratio.

Fuzzy and Ant Colony Optimization Based Combined MAC, Routing, and Unequal Clustering Cross-Layer protocol developed by Gajjar, Sarkar [47], named FAMACROW, combined hierarchical energy-efficient clustering with media access to reduce energy consumption and prolong network life. In the fuzzy logic system, three factors are used to determine CHs: remaining energy, node density, and communication link quality. As a solution to the hot-spot issue, FAMACROW employs an uneven clustering method. In the inter-cluster routing protocol, the relay node is chosen based on the distance between the CH and the base station, the remaining energy, congestion control length, and delivery probability. Compared to the IFUC, FAMACROW provided a 75%-88% increased network lifespan, 82% extra packets, and 41% improved energy efficiency.

Xunli and Feiefi [48] proposed Sink Mobility based Energy Balancing Unequal Clustering (SMEBUC). In energy-restricted WSNs, coverage and connectivity are two main QoS factors that affect the efficiency of WSNs by decreasing the energy consumed by nodes and extending the coverage lifetime. The distribution of nodes in cluster-based WSNs leads to increased energy inequality. SMEBUC solves these problems by selecting the CHs based on the energy and by making the clusters unequal in size using the improved Shuffled Frog Leaping Algorithm (SFLA). The CHs are used constantly to identify node weights and CH replacement times. The objective is to determine which relay node between CH and the base station would be most appropriate. Additionally, SMEBUC employs the sink location mobile algorithm to prevent the appearance of hot-spot problems. The results of the experiments indicate that SMEBUC outperforms LEACH and EBUCP, achieves more energy efficiency, and provides a solution to the hot-spot problem in a multi-hop routing protocol by replacing CHs after a period of time.

The compact Bat Algorithm (cBA) was used by Nguyen, Pan [49] to present a novel optimization algorithm for the uneven clustering problem in WSNs. The cBA generated new candidate solutions for space search based on a probabilistic model, which has proven to be promising so far. An explicit probabilistic model was built, and samples were taken from the built models to find the candidate solution. In order to construct the operations of selecting and optimizing behaviors, cumulative distribution and probability density functions are used extensively. Probabilistic modeling is a valid alternative optimization strategy for devices with restricted hardware capabilities. The presented algorithm achieves the efficient use of limited memory devices and provides competitive results.

Zhu and Wang [50] propose an energy-balanced unequal clustering routing protocol based on the improved sine cosine algorithm (DUCISCA). As a first step, DUCISCA uses a time-based algorithm for cluster head competitions. The number of neighbors, the residual energy of the candidate CHs, and the distance to the base station determine the broadcast time in this algorithm. Secondly, a competition radius based on node distance and residual energy is proposed. Energy consumption can be balanced between nodes at different locations to prevent hot spots. It also implements a time-based broadcast mechanism. Finally, the improved sine-cosine algorithm (ISCA) based on Latin hypercube sampling and adaptive mutation is introduced to improve convergence and jump out of the local optimum. The experiment results indicate that DUCISCA balances energy consumption across the entire network and extends the lifetime of the network more effectively than previous approaches.

3) Fuzzy-based approaches: The presence of uncertainties in WSNs requires the use of the fuzzy logic system to make effective and efficient decisions. Various clustering algorithms use fuzzy logic to make effective decisions about clustering. Over classical approaches, fuzzy logic offers lower execution complexity, development costs, less memory, greater flexibility, and auto fault tolerance. Fuzzy inputs for selecting the cluster heads are remaining energy, the density of nodes, distance to the base station, and distance among the neighbor nodes. Fuzzy outputs offer the option to select the CH and the size of the cluster. Several deterministic fuzzy-based clustering strategies have been proposed to minimize clustering energy consumption.

Baranidharan and Santhi [51] offered Distributed load balancing Unequal Clustering (DUCF). Fuzzy logic is used to select the CHs. DUCF arranges clusters of unequal sizes in order to keep energy usage balanced among them. The CHs are chosen using a fuzzy inference procedure incorporating residual energy, degree, and distance between the base station and the nodes. DUCF limits cluster sizes based on fuzzy input parameters. DUCF balances the load between clusters by varying cluster sizes. Multi-hop data transmission by DUCF conserves energy by balancing energy consumption among nodes and reducing energy usage in CHs. DUCF produced better results in various network scenarios than LEACH, CHEF, and EAUCF.

Multi-objective fuzzy clustering algorithm (MOFCA) is offered by Sert, and Bagci [52]. Hot spots can currently be solved by developing unequal-sized clustering methods, which generate smaller clusters to reduce the intra-cluster relays when reaching the base station. Changing the location of deployed nodes may also introduce the energy-hole problem. Prior studies have not considered both problems (Hot-Spot and Energy-Hole) jointly in uniformed or non-uniform distributions. Thus, MOFCA was a solution to static and dynamic network problems. MOFCA uses the remaining energy, the distance to the base station, and the density of nodes as input variables for the fuzzy inference system to overcome the uncertainty in WSNs in nature. Results show that MOFCA is a more energy-efficient protocol in terms of FND, HND, and total residual energy. MOFCA considers both dynamic and static nodes, and mobility is implemented by changing the location of nodes without consuming any energy.

The Secure Unequal Clustering Protocol with Intrusion Detection (SUCID) proposed by Maheswari and Karthika [53] achieved QOS variables such as security, lifespan, and power. It contains several different procedures, including initialization of the nodes, election of the TCH, the election of the FCH, cluster maintenance, and identification of intruders. Using the neuro-fuzzy grouping method, the tentative CHs (TCHs) were initially selected based on the three input variables, distance to the base station, residual energy, and distance between the nodes in close proximity. Using "DHO: Deer hunting optimization" based on TCH, final CHs and optimal cluster heads were selected based on the fitness derived from five variables, including distance to the base station, remaining energy, degree of nodes, the centrality of nodes, and link quality. Cluster maintenance was used to balance the load, and a deep belief network was used on the cluster heads to detect intruders in the WSN. Results from experiments showed that SUCID maximized energy efficiency, network lifespan, delayed average, and intruder detection rate by combining DBN and IDS.

Siqing, Yang [33] introduced Fuzzy Logic for Multi-hop Networks (FLCMN), which partitions the time into rounds with equal intervals. Each round comprises two stages: clustering and data transmission. The FLCMN algorithm uses fuzzy logic inference to determine the CH. There are three input parameters for fuzzy logic inference systems: density of nodes, residual energy of neighbors, and the average remaining energy of neighbors. Cluster Head Formation is the subsequent phase, where nodes broadcast a message, and after comparing the degree, a node is selected as the CH. Once CH has sent a message to join, nodes select their nearest CH. FLCMN supports multi-hop data transmission. Hot-spot problems can be prevented by FLCMN, as well as the distribution of nodes within clusters can be regulated. As part of FLCMN, clusters are also connected in a multi-hop manner in a Fibonacci series sequence, which prevents unnecessary data transmissions by the CHs, thereby extending the life of WSNs. The FLCMN algorithm performs better regarding network lifetime and energy consumption than existing DFLC, LEACH, and EAMMH algorithms when comparing network lifetime and energy consumption between nodes.

C. Probabilistic Uneven Clustering Approaches

CHs are selected at random without regard to predetermined protocols in probabilistic clustering algorithms. These types of algorithms are simple, energy-efficient, and optimized. Probabilistic clustering is primarily used to extend the life of WSNs. Energy-efficient clustering requires a low message and time complexity. Algorithms for probability clustering comprise two types: random and hybrid algorithms.

1) Random probabilistic algorithms: In random probabilistic algorithms, CHs are randomly chosen, simple in nature, and have the lowest overhead. Lee, and Choe [54] presented a mathematical framework and developed a clustering algorithm called the Location-based Unequal Clustering Algorithm (LUCA). Cluster sizes are determined by their distance from the CH. The cluster size of LUCA clusters increases as the distance from the base station increases to diminish the energy consumption of the whole network. According to simulation results, LUCA outperforms the conventional equal clustering algorithm in terms of energy efficiency.

Kim and Hussain [55] suggest a randomized and distributed clustering algorithm, PRODUCE, that includes clusters of unequal size. CHs located near the base station may concentrate on inter-cluster communication, whereas CHs farther away may focus on intra-cluster communication. This prevents excessively long-distance communications in the network from negatively affecting signal strength. Simulation results demonstrate that the proposed algorithm considerably boosts coverage time and network lifetime at high node densities, especially when coverage time and network lifetime are important factors.

Huang and Hong [56] present an energy-efficient multi-hop routing algorithm based on grid clustering, called EEMRP. Several parameters, including network area levels, the position of nodes, and the energy of nodes, are considered by the algorithm to minimize energy consumption. Cluster heads are relieved of the burden of transferring data among clusters via multi-hop routing by introducing communication nodes. EEMRP extends the lifetime of the network by 17.5–25.2% compared to other algorithms, according to simulation results.

Handy, Haase [57] modified the LEACH protocol and extended its stochastic CH selection algorithm by a deterministic component. The network lifetime is increased by about 30%, depending on the configuration. Further, they developed a three-factor metric to define the lifespan of microsensor networks, the last node dies, half of nodes alive, and the first the node dies.

2) Hybrid probabilistic algorithms: In hybrid probabilistic algorithms, CHs are selected using a hybrid method, which involves randomness and parameters like distance from the base station, residual energy, node density, etc., which balance the clusters. Due to the iterative and competition-based nature of hybrid probabilistic algorithms, these algorithms have a higher message and time complexity.

Li, and Liu [58] examine unequal clustering in a uniform distribution of WSNs from a theoretical point of view. They then develop an efficient clustering scheme to minimize energy consumption. Additionally, lightweight and optimal protocols are designed for the routing and rotation of cluster heads to ensure even power consumption. Simulation results indicate that the proposed approach extends network lifetimes over the best-so-far unequal clustering-based routing approach.

Bozorgi and Bidgoli [59] propose improving the energy efficiency of previous clustering methods and extending the lifetime of the network with a hybrid unequal clustering method. A novel clustering strategy is used in the proposed protocol. Nodes in a network use their neighbors' information based on the arrangement of nodes. By using this strategy, overhead can be reduced significantly. Clustering results in nodes nearer to the base station receiving and relaying data with more energy than nodes farther away. As a hybrid static-dynamic clustering scheme, no control message must be transmitted at each round to reduce overhead. There are two new routing techniques proposed. Member nodes support CHs with sufficient energy and distance to share the cluster's load with cluster heads. Discretion licensing is another new technique that prevents incomplete packets from being sent in real-time. According to simulation results, the proposed method reduces network overhead, improves network stability, balances energy, and extends the lifetime of networks.

Priyadarshi, Singh [60] propose a novel Hybrid Energy Efficient Distributed (HEED) algorithm for networks with non-uniformly distributed nodes. A novel HEED protocol has been developed for extending the lifetime of clustered non-uniform sensor networks. HEED and its variants are compared based on the amount of energy dissipated and the number of live nodes. The planned HEED protocol has a longer lifespan and is more energy efficient than the existing HEED protocol.

TABLE I. COMPARISON OF PRESET UNEVEN CLUSTERING APPROACHES

Protocol	**Nodes type**	**Nature**	**Location-aware**	**Objective**	**Node or relay aggregation**	**Mobility-aware**	**Cluster count**	**Inter-cluster communication type**	**Intra-cluster communication type**
UCS	Heterogeneous and Homogeneous	Proactive	Yes	Network lifetime and load balancing	Yes	No	Variable	Multi-hop	Single-hop

TABLE II. COMPARISON OF DETERMINISTIC UNEVEN CLUSTERING APPROACHES

Protocol	**Nodes type**	**Nature**	**Location-aware**	**Objective**	**Node or relay aggregation**	**Mobility-aware**	**Cluster count**	**Inter-cluster communication type**	**Intra-cluster communication type**
EDDUCA	Heterogeneous	Proactive	No	Network lifetime and load balancing	Yes	No	Variable	Multi-hop	Single-hop
UCCGRA	Homogeneous	Proactive	Yes	Load balancing, scalability, and network lifetime	Yes	No	Static	Multi-hop	Multi-hop
PEG-ant	Heterogeneous	Reactive	No	Energy consumption and network lifetime, and fault-tolerant	Yes	Yes	Variable	Multi-hop	Single-hop
ACOPSO	Homogeneous	Reactive	No	Fault-tolerant and load balancing	Yes	No	Variable	Multi-hop	Single-hop
UMBIC	Homogeneous	Proactive	Yes	Network lifetime and load balancing	Yes	No	Static	Multi-hop	Multi-hop
nCRO	Heterogeneous	Reactive	Yes	Network lifetime and load balancing	Yes	No	Variable	Multi-hop	Single-hop
FAMACROW	Heterogeneous	Proactive	No	Network lifetime and load balancing	Yes	No	Variable	Multi-hop	Single-hop
SMEBUC	Heterogeneous	Proactive	No	Network lifetime and energy consumption	Yes	Yes	Static	Multi-hop	Multi-hop
cBA	Heterogeneous	Reactive	Yes	Network lifetime and energy consumption	No	No	Variable	Multi-hop	Single-hop
DUCISCA	Homogeneous	Proactive	No	Network lifetime and energy consumption	Yes	No	Static	Single-hop	Single-hop
DUCF	Homogeneous	Reactive	No	Network lifetime, scalability, and load balancing	Yes	No	Variable	Multi-hop	Multi-hop
MOFCA	Heterogeneous	Proactive	No	Network lifetime and fault-tolerant	No	Yes	Variable	Multi-hop	Multi-hop
SUCID	Heterogeneous	Proactive	No	Scalability, delay, and energy consumption	Yes	No	Variable	Multi-hop	Single-hop
FLCMN	Heterogeneous	Proactive	Yes	Load balancing and energy consumption	Yes	No	Variable	Multi-hop	Single-hop

TABLE III. COMPARISON OF PROBABILISTIC UNEVEN CLUSTERING APPROACHES

Protocol	Nodes type	Nature	Location-aware	Objective	Node or relay aggregation	Mobility-aware	Cluster count	Inter-cluster communication type	Intra-cluster communication type
LUCA	Homogeneous	Proactive	No	Network lifetime and scalability	No	Yes	Variable	Multi-hop	Single-hop
PRODUCE	Homogeneous	Reactive	No	Energy consumption and fault-tolerant	No	No	Static	Single-hop	Single-hop
EEMRP	Homogeneous	Proactive	No	Network lifetime and scalability	Yes	Yes	Variable	Multi-hop	Multi-hop
LEACH –DCH	Heterogeneous	Proactive	Yes	Network lifetime	No	No	Variable	Multi-hop	Single-hop
COCA	Homogeneous	Proactive	No	Network lifetime and energy consumption	No	Yes	Variable	Multi-hop	Single-hop
HEEC	Homogeneous	Reactive	No	Energy consumption	Yes	No	Variable	Multi-hop	Single-hop
HEED	Homogeneous	Proactive	No	Energy consumption	No	No	Variable	Multi-hop	Single-hop

As comparison tables show, CHs are chosen using a hybrid probabilistic method that takes into account randomness and a variety of factors, including residual energy, distance from the base station, node density, etc. Higher message and time complexity result from hybrid probabilistic algorithms' iterative and competition-based nature. Deterministic clustering algorithms manage clusters and choose cluster leaders wisely. Based on factors including remaining energy, node density, and distance from the base station, deterministic clustering algorithms choose cluster heads.

IV. DISCUSSION AND RESULTS

Tabular comparisons of uneven clustering are presented in Tables I, II, and III. As a starting point, clustering algorithms are evaluated according to the following characteristics: size (even or uneven), cluster count (static or dynamic), and type of communication (multi-hop or single-hop). In the next step, we compared several clustering algorithms based on the characteristics of cluster heads. For example, mobility can be fixed or mobile, nodes can be heterogeneous or homogeneous, roles can be aggregated or relayed, methods like probabilistic, deterministic, and preset, and objectives for each algorithm. In addition, we examined several clustering algorithms based on selecting cluster heads, the nature of the cluster, and its location awareness.

In uneven clustering, three protocols are distinguished: preset, probabilistic, and deterministic. There are two types of probabilistic clustering: random clustering and hybrid clustering. Although random-type clustering fails to conserve energy, it is simple and achieves very low overhead. A hybrid probabilistic algorithm selects CHs based on randomness and parameters such as residual energy, distance from the base station, node density, etc. Hybrid probabilistic algorithms' iterative and competition-based nature leads to higher message and time complexities. Clustering algorithms based on deterministic principles control clusters and elect cluster heads optimally. Deterministic clustering algorithms select cluster heads based on parameters such as remaining energy, node density, distance from the base station, etc. These parameters can be obtained locally, and the neighbor nodes exchange messages to update them. There are three deterministic clustering algorithms: fuzzy-based, compound-based, and heuristic-based.

Clustering algorithms based on fuzzy logic consume more power during algorithm execution and message exchange. It is impractical for various applications to use heuristic-based clustering algorithms due to the need for global information and the fact that the base station fully manages the algorithms. Depending on the application, this is the most appropriate choice. Clustering algorithms based on weights are iterative, which adds complexity to messages. In compound-based unequal clustering, different metrics are used, such as Sierpinski triangles and linked graphs. Deterministic compound-based methods are proposed to make clustering more energy-efficient.

Most clustering methods are static at the moment, and they cannot adapt to network changes. Therefore, dynamic processes of clustering may prove more effective in the future. The mobility of networks is not intended in this case. For clustered WSNs, we have three components: members of clusters, cluster heads, and finally, the base station. WSNs with mobility require frequent configuration changes, increasing overhead. The WSN is typically data-centric and requires data-centric methods, as discussed earlier. Most algorithms focus on proactive networks, but very few are reactive. Clustering methods may become more reactive in the future.

V. CONCLUSION

The purpose of WSNs is to provide valuable information over long periods with the least amount of energy consumption, exhibiting optimal performance with reduced delays. The low capacity of the battery may, however, pose a serious problem regarding energy consumption. Clustering is found to be an effective technique in power management in WSNs. However, clustering is susceptible to hot-spot problems. Using uneven-type clustering, the load is distributed equally, the hot-spot problem is resolved, and the WSN's lifetime is enhanced. We classified the various uneven clustering algorithms into three broad groups: preset, deterministic, and probabilistic clustering algorithms. The methods are described in terms of their goals, attributes, categories, advantages, and disadvantages. Probabilistic clustering is used when we need simplicity and speed. We employ this type of clustering when we need WSN surveillance on a large scale. To implement more robust and reliable applications, we need deterministic clustering. The heuristic type clustering can be used when we need an optimized solution for an application-specific situation. In this paper, we also compared all these types of protocols based on their clustering properties, CHs properties, and also on the type of clustering process.

REFERENCES

[1] O.A. Khashan, R. Ahmad, N.M. Khafajah, An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks, Ad Hoc Networks, 115 (2021) 102448.

[2] A. Singh, S. Sharma, J. Singh, Nature-inspired algorithms for wireless sensor networks: A comprehensive survey, Computer Science Review, 39 (2021) 100342.

[3] J. Li, Y. Zhang, L. Xu, L. Zhang, Z. Yuan, Modeling and planning a transmission network expansion system in a regulated electricity market by considering demand-side management via a developed fuzzy-salp optimization algorithm, Advances in Engineering and Intelligence Systems, 1 (2022).

[4] M.S. Yousefpoor, E. Yousefpoor, H. Barati, A. Barati, A. Movaghar, M. Hosseinzadeh, Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review, Journal of Network and Computer Applications, 190 (2021) 103118.

[5] S.R. Jondhale, R. Maheswar, J. Lloret, Fundamentals of Wireless Sensor Networks, Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks, Springer2022, pp. 1-19.

[6] H. Nazemi, A. Joseph, J. Park, A. Emadi, Advanced micro-and nano-gas sensor technology: A review, Sensors, 19 (2019) 1285.

[7] H. Golpîra, S.A.R. Khan, S. Safaeipour, A review of logistics internet-of-things: Current trends and scope for future research, Journal of Industrial Information Integration, 22 (2021) 100194.

[8] D.K. Sah, K. Cengiz, P.K. Donta, V.N. Inukollu, T. Amgoth, EDGF: Empirical dataset generation framework for wireless sensor networks, Computer Communications, 180 (2021) 48-56.

[9] Y. Liu, A Reliable Approach for Solving Transmission Network Expansion Planning with Objective of Planning Cost Reduction, Advances in Engineering and Intelligence Systems, 1 (2022).

[10] P. Singh, R. Singh, Energy-efficient QoS-aware intelligent hybrid clustered routing protocol for wireless sensor networks, Journal of Sensors, 2019 (2019).

[11] V. Kumar, V. Kumar, D. Sandeep, S. Yadav, R.K. Barik, R. Tripathi, S. Tiwari, Multi-hop communication based optimal clustering in hexagon and voronoi cell structured WSNs, AEU-International Journal of Electronics and Communications, 93 (2018) 305-316.

[12] I. Daanoune, B. Abdennaceur, A. Ballouk, A comprehensive survey on LEACH-based clustering routing protocols in Wireless Sensor Networks, Ad Hoc Networks, 114 (2021) 102409.

[13] D. Kandris, C. Nakas, D. Vomvas, G. Koulouras, Applications of wireless sensor networks: an up-to-date survey, Applied System Innovation, 3 (2020) 14.

[14] G.T.S. Ho, Y.P. Tsang, C.H. Wu, W.H. Wong, K.L. Choy, A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities, Sensors, 19 (2019) 1796.

[15] A. Kaul, I. Altaf, Vanet-TSMA: A traffic safety management approach for smart road transportation in vehicular ad hoc networks, International Journal of Communication Systems, 35 (2022) e5132.

[16] B. Bhushan, G. Sahoo, Requirements, protocols, and security challenges in wireless sensor networks: An industrial perspective, Handbook of computer networks and cyber security, Springer2020, pp. 683-713.

[17] X. Zhu, Complex event detection for commodity distribution Internet of Things model incorporating radio frequency identification and Wireless Sensor Network, Future Generation Computer Systems, 125 (2021) 100-111.

[18] A. Kaul, I. Altaf, Vanet-TSMA: A traffic safety management approach for smart road transportation in vehicular ad hoc networks, International Journal of Communication Systems, 35 (2022) e5132.

[19] M.A. Khan, A. Saboor, H.-c. Kim, H. Park, A systematic review of location aware schemes in the Internet of Things, Sensors, 21 (2021) 3228.

[20] S. Singh, A.S. Nandan, A. Malik, R. Kumar, L.K. Awasthi, N. Kumar, A GA-Based Sustainable and Secure Green Data Communication Method Using IoT-Enabled WSN in Healthcare, IEEE Internet of Things Journal, 9 (2021) 7481-7490.

[21] V. Kavitha, Privacy preserving using multi-hop dynamic clustering routing protocol and elliptic curve cryptosystem for WSN in IoT environment, Peer-to-Peer Networking and Applications, 14 (2021) 821-836.

[22] L. Manoharan, A. Leni, Distributed Uneven Clustering Mechanism for Energy Efficient WSN, Wireless Personal Communications, 121 (2021) 153-169.

[23] D. Jain, P.K. Shukla, S. Varma, Energy efficient architecture for mitigating the hot-spot problem in wireless sensor networks, Journal of Ambient Intelligence and Humanized Computing, (2022) 1-18.

[24] P. Rao, P. Lalwani, H. Banka, G. Rao, Competitive swarm optimization based unequal clustering and routing algorithms (CSO-UCRA) for wireless sensor networks, Multimedia Tools and Applications, 80 (2021) 26093-26119.

[25] N. Moussa, A. El Belrhiti El Alaoui, An energy-efficient cluster-based routing protocol using unequal clustering and improved ACO techniques for WSNs, Peer-to-Peer Networking and Applications, 14 (2021) 1334-1347.

[26] V. Rajaram, N. Kumaratharan, Multi-hop optimized routing algorithm and load balanced fuzzy clustering in wireless sensor networks, Journal of Ambient Intelligence and Humanized Computing, 12 (2021) 4281-4289.

[27] C. Zhao, Q. Wu, D. Lin, Z. Zhang, Y. Zhang, L. Kong, Y.L. Guan, An energy-balanced unequal clustering approach for circular wireless sensor networks, Ad Hoc Networks, 132 (2022) 102872.

[28] O.I. Khalaf, C.A.T. Romero, S. Hassan, M.T. Iqbal, Mitigating hotspot issues in heterogeneous wireless sensor networks, Journal of Sensors, 2022 (2022).

[29] G.V. Selvi, V. Muthukumaran, A. Kaladevi, S.S. Kumar, B. Swapna, Integrated dominating and hit set-inspired unequal clustering-based data aggregation in wireless sensor networks, International Journal of Intelligent Computing and Cybernetics, (2022).

[30] R.F. Mansour, S.A. Alsuhibany, S. Abdel-Khalek, R. Alharbi, T. Vaiyapuri, A.J. Obaid, D. Gupta, Energy Aware Fault Tolerant Clustering with Routing Protocol for Improved Survivability in Wireless Sensor Networks, Computer Networks, (2022) 109049.

[31] V. Chauhan, S. Soni, Energy aware unequal clustering algorithm with multi-hop routing via low degree relay nodes for wireless sensor networks, Journal of Ambient Intelligence and Humanized Computing, 12 (2021) 2469-2482.

[32] B.M. Sahoo, T. Amgoth, An improved bat algorithm for unequal clustering in heterogeneous wireless sensor networks, SN Computer Science, 2 (2021) 1-10.

[33] Z. Siqing, T. Yang, Y. Feiyue, Fuzzy logic-based clustering algorithm for multi-hop wireless sensor networks, Procedia computer science, 131 (2018) 1095-1103.

[34] J. Amutha, S. Sharma, S.K. Sharma, Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: Review, taxonomy, research findings, challenges and future directions, Computer Science Review, 40 (2021) 100376.

[35] M.S. Thekiya, M.D. Nikose, Energy efficient clustering routing protocol using novel admission allotment scheme (AAS) based intra-cluster communication for Wireless Sensor Network, International Journal of Information Technology, 14 (2022) 2815-2824.

[36] B.S. Kumar, P.T. Rao, An optimal emperor penguin optimization based enhanced flower pollination algorithm in WSN for fault diagnosis and prolong network lifespan, Wireless Personal Communications, (2021) 1-18.

[37] D.K. Kotary, S.J. Nanda, R. Gupta, A many-objective whale optimization algorithm to perform robust distributed clustering in wireless sensor network, Applied Soft Computing, 110 (2021) 107650.

[38] F.S. Mukti, J.E. Lorenzo, R. Zuhdianto, A. Junikhah, A. Soetedjo, A.U. Krismanto, A Comprehensive Performance Evaluation of Proactive, Reactive and Hybrid Routing in Wireless Sensor Network for Real Time Monitoring System, 2021 International Conference on Computer Science and Engineering (IC2SE), IEEE, 2021, pp. 1-6.

[39] D. Thomas, R. Shankaran, M.A. Orgun, S.C. Mukhopadhyay, SEC 2: A Secure and Energy Efficient Barrier Coverage Scheduling for WSN-Based IoT Applications, IEEE Transactions on Green Communications and Networking, 5 (2021) 622-634.

[40] S. Soro, W.B. Heinzelman, Prolonging the lifetime of wireless sensor networks via unequal clustering, 19th IEEE international parallel and distributed processing symposium, IEEE, 2005, pp. 8 pp.

[41] A.B.F. Guiloufi, N. Nasri, A. Kachouri, An energy-efficient unequal clustering algorithm using 'Sierpinski Triangle'for WSNs, Wireless Personal Communications, 88 (2016) 449-465.

[42] H. Xia, R.-h. Zhang, J. Yu, Z.-k. Pan, Energy-efficient routing algorithm based on unequal clustering and connected graph in wireless sensor networks, International Journal of Wireless Information Networks, 23 (2016) 141-150.

[43] W. Guo, W. Zhang, G. Lu, PEGASIS protocol in wireless sensor network based on an improved ant colony algorithm, 2010 Second international workshop on education technology and computer science, IEEE, 2010, pp. 64-67.

[44] S. Kaur, R. Mahajan, Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks, Egyptian Informatics Journal, 19 (2018) 145-150.

[45] N. Sabor, M. Abo-Zahhad, S. Sasaki, S.M. Ahmed, An unequal multi-hop balanced immune clustering protocol for wireless sensor networks, Applied Soft Computing, 43 (2016) 372-389.

[46] P. Srinivasa Rao, H. Banka, Novel chemical reaction optimization based unequal clustering and routing algorithms for wireless sensor networks, Wireless Networks, 23 (2017) 759-778.

[47] S. Gajjar, M. Sarkar, K. Dasgupta, FAMACROW: Fuzzy and ant colony optimization based combined mac, routing, and unequal clustering cross-layer protocol for wireless sensor networks, Applied Soft Computing, 43 (2016) 235-247.

[48] F. Xunli, D. Feiefi, Shuffled frog leaping algorithm based unequal clustering strategy for wireless sensor networks, Appl. Math. Inf. Sci, 9 (2015) 1415-1426.

[49] T.-T. Nguyen, J.-S. Pan, T.-K. Dao, A compact bat algorithm for unequal clustering in wireless sensor networks, Applied Sciences, 9 (2019) 1973.

[50] F. Zhu, W. Wang, A Distributed Unequal Clustering Routing Protocol Based on the Improved Sine Cosine Algorithm for WSN, Journal of Sensors, 2022 (2022).

[51] B. Baranidharan, B. Santhi, DUCF: Distributed load balancing unequal clustering in wireless sensor networks using fuzzy approach, Applied Soft Computing, 40 (2016) 495-506.

[52] S.A. Sert, H. Bagci, A. Yazici, MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks, Applied Soft Computing, 30 (2015) 151-165.

[53] M. Maheswari, R. Karthika, A novel QoS based secure unequal clustering protocol with intrusion detection system in wireless sensor networks, Wireless Personal Communications, 118 (2021) 1535-1557.

[54] S. Lee, H. Choe, B. Park, Y. Song, C.-k. Kim, LUCA: An energy-efficient unequal clustering algorithm using location information for wireless sensor networks, Wireless Personal Communications, 56 (2011) 715-731.

[55] J.-H. Kim, C.S. Hussain, W.-C. Yang, D.-S. Kim, M.-S. Park, Produce: A probability-driven unequal clustering mechanism for wireless sensor networks, 22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops 2008), IEEE, 2008, pp. 928-933.

[56] J. Huang, Y. Hong, Z. Zhao, Y. Yuan, An energy-efficient multi-hop routing protocol based on grid clustering for wireless sensor networks, Cluster Computing, 20 (2017) 3071-3083.

[57] M. Handy, M. Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection, 4th international workshop on mobile and wireless communications network, IEEE, 2002, pp. 368-372.

[58] H. Li, Y. Liu, W. Chen, W. Jia, B. Li, J. Xiong, COCA: Constructing optimal clustering architecture to maximize sensor network lifetime, Computer Communications, 36 (2013) 256-268.

[59] S.M. Bozorgi, A.M. Bidgoli, HEEC: A hybrid unequal energy efficient clustering for wireless sensor networks, Wireless Networks, 25 (2019) 4751-4772.

[60] R. Priyadarshi, L. Singh, A. Singh, A novel HEED protocol for wireless sensor networks, 2018 5th international conference on signal processing and integrated networks (SPIN), IEEE, 2018, pp. 296-300.

[61] Alomari, M.F., Mahmoud, M.A. and Ramli, R. A Systematic Review on the Energy Efficiency of Dynamic Clustering in a Heterogeneous Environment of Wireless Sensor Networks (WSNs). Electronics, 2022, 11(18), p.2837.

[62] Priyadarshini, R.R.; Sivakumar, N. Cluster head selection based on minimum connected dominating set and bi-partite inspired methodology for energy conservation in WSNs. J. King Saud Univ. Inf. Sci. 2021, 33, 1132–1144.

[63] Jain, K., Kumar, A. and Jha, C.K. Probabilistic-based energy-efficient single-hop clustering technique for sensor networks. In Communication and Intelligent Systems: Proceedings of ICCIS 2020 (pp. 353-365). Springer Singapore.

[64] Jasim, A.A., Idris, M.Y.I., Razalli Bin Azzuhri, S., Issa, N.R., Rahman, M.T. and Khyasudeen, M.F.B. Energy-efficient wireless sensor network with an unequal clustering protocol based on a balanced energy method (EEUCB). Sensors, 2021, 21(3), p.784.

Univariate and Multivariate Gaussian Models for Anomaly Detection in Multi Tenant Distributed Systems

Prof. Pravin Ramdas Patil[1], Dr. Geetanjali Kale[2]
Assistant Professor[1]
Head of Department and Associate Professor[2]
Department of Computer Engineering, SCTR’s Pune Institute of Computer Technology,
Pune, Maharashtra, India[1, 2]

***Abstract*—Due to the flaws in shared memory, settings, and network access, distributed systems on a network always have been susceptible to cyber intrusions. Co-users on the same server give attackers the chance to monitor the activity of many other users and launch an attack when those users' security is at risk. Building completely secure network topologies immune from risks and assaults has traditionally been the goal. It is also hard to create an architecture that is 100 percent safe due to its open-ended nature. The precise parameters and infrastructure design whereby the strike is instantiated are a constant which can always be detected regardless of the sort of attack. This work now have the chance to simulate any abnormality and subsequent attack possibilities using network parameter values thanks to the increased usage of algorithms for machine learning and data-gathering tools. This work proposes a Gaussian model to forecast the likelihood of an attack occurring depending on certain system parameters. This work model a univariate and a multivariate Gaussian model on the training dataset. This work makes use of various threshold values to predict whether the data point is an inlier or an outlier. This research examines accuracies for various threshold values. An important challenge in an anomaly detection situation is class imbalance. As long as this work just utilizes training data, a class imbalance is not a problem. Our data-driven results show that combining machine learning with Gaussian-based models might be a useful tool for analyzing network intrusions. Although more steps are being made to boost digital space security, machine learning algorithms may be utilized to examine any abnormal behavior that is left uncontrolled.**

Keywords—Multi-tenant distributed system; anomaly detection; outlier detection; machine learning; Gaussian model

I. Introduction

One of today's most demanding technologies is cloud computing. Cloud computing offers an infinite quantity of IT facilities to deliver amazing computing speed, but on the flip side, it has serious security problems with public clouds for multitenant cloud environments. Most government and commercial companies are compromising with the limited IT resources and performance from existing resources since they are not migrating their sensitive and private data over the public cloud due to security concerns. The aforementioned problems will be solved by finding a way to protect private space over public clouds.

Multiple clients can use the services provided by multi-tenant distributed systems. As a result, each client has access to the activity of the others. By being one of the clients of such a system and taking advantage of such surveillance, attackers can launch assaults against one or more other tenants of the system [1]. To stop any entity in the system from suffering damage, such an attack must be promptly detected [2]. The scourge of attacks in such distributed systems has been a hot topic among researchers despite improvements in cyber security measures. Although cyber security protections have improved, experts continue to focus on the problem of intrusions in such distributed multi-tenant systems. Multiple tenants can cohabit on the same network thanks to multi-tenant distributed systems (MTDS). The MTDS service provider does not inquire about the tenant's motivations when they request co-allocation. This situation presents a chance for renters with bad intentions to observe and collect confidential information about the target occupants. [4] Because the attacker tenant has access to sensitive information, the tenant may prepare an attack that has a greater likelihood of success. [3]

There have already been several attempts to use a variety of techniques to identify the existence of intrusions in distributed applications. [5], [6] Earlier, the emphasis was on applying statistical techniques to compute specific function values, but more recently, cutting-edge approaches including deep learning have been applied. In this regard, artificial neural networks have been investigated.

Although rule-based engines were used to identify assaults, they frequently fall short of spotting any newly discovered threats. Transfer learning may be helpful in this situation, but there is no guarantee that the variables of the source work and the destination job are identical, which has been a significant obstacle to its application. [7]

This work suggests a Gaussian-based classifier strategy in this research for identifying the potential for intrusions in a multi-tenant distributed system to identify inliers and outliers. This work defines a threshold value. This work also looks at the accuracy of different threshold values. Authors are thankful to Patil and Ingale [8] for providing us with the dataset.

Section II of paper includes literature survey of research work done in the area of network attack detection. It explores

Machine learning algorithms used to detect network attacks and to improve cyber security. Section III describes experimentation performed to create and collect dataset. As network attack is not a continuously or regularly occurring event hence lesser number of attacks are performed to create dataset. This dataset includes majority non attack instances and very few attack instances. This section includes statistical and graphical representation of collected dataset. Section IV explains creation of univariate and multivariate Gaussian models for anomaly detection and respective models performance analysis. Section V contains conclusion of research work done.

II. Related Work

Network attack detection has historically made heavy use of signature-based detection. This approach uses an analysis of an attack's "signature," or distinctive qualities, to foretell potential hazards in the future [9]. Methods to discover the best attack signatures were suggested by Hilker et al. [10]. Han et al. [11] advocated crafting network traffic using several attributes. The system cannot identify any new attacks that were previously undiscovered owing to a lack of knowledge about them, which is a significant problem with this technique. Additionally, each new effort to locate signatures requires human labor in addition to time.

Additionally, there have been initiatives to employ machine learning algorithms in this field. Algorithms based on supervised learning have traditionally been used to identify network attacks. [12] For assault detection, Zseby et al. favoured the use of selecting features and subsequent mapping [13]. Evolutionary algorithms were used by Rafique et al. [14] to evaluate the effectiveness of classifying malware. The chance of assault is extremely low, it should be highlighted, therefore a model may get away with forecasting all data as non-negative and yet show good accuracy, making the entire process exceedingly costly.

Prior strategies likewise emphasized the application of boosting techniques and feature reduction in transfer learning. TrAdaBoost was introduced by Dai et al. [15] and reweights the data from the positive and negative classes to give the uncommon examples that indicate attacks more weight in the outcome. TCA-transfer component analysis was used by Pan et al. to feature project the domains closer to one another in the common space [16]. HeMap is a technique created by Shi et al. [17] that projects features using linear transformations. Patil and Ingale [8] tackled the class imbalance problem and used an ensemble based meta classifier to detect anomaly.

The detection of assaults has also been done using model-based methods. This strategy falls under the category of transfer learning and makes the crucial assumption that the source task and the target task share at least some parameters or model priors. Bekerman demonstrated how transfer learning may help increase the resilience of malware detection in uncharted situations. [17].

A noteworthy finding in all of these prior methods was that the stark class disparity seen in network assaults was hardly discussed. Additionally, due to this imbalance, effectiveness of other measures should also be discussed in order to shed light on the results that were produced. We model a Gaussian model on the training dataset. The advantage of this method is that class imbalance does not cause any hindrance.

Research community is contributing towards improving cyber security and security of multi-tenant distributed systems. Despite being all these efforts, attackers are successfully able to place compromised or virtual machine having anomaly to reside with target virtual machine. This leads to increase in the probability of having successful attack on a target virtual machine. Detection of new types of attack possible because of co-residence, co-location and co-tenant of attacker virtual machine with a target virtual machine is still remains a challenge to researchers. Univariate and Multivariate Gaussian models are created to detect network attacks. Performance analysis of individual models created is performed.

III. Dataset Preparation

A. Dataset Collection

Dataset has been collected by Patil and Ingale [8] by using Netdata, a programme for real-time performance monitoring that creates system logs. The logs have been collected across 28 files. This work combines all the files into a single dataset for easy handling. The dataset consists of 4986 inliers instances and 60 outlier instances with 63 columns. All the columns names are noted in Table I.

B. Dataset Preparation

Contributors dropped the column ‘anomaly score’ as it is generated by the software. Authors also separate ‘label’ from the remaining dataset. Authors also drop the columns whose standard deviation is less than 0.3 but also store the original dataset. Contributors are left with 36 columns in the remaining dataset. This work plot some of the important columns as a categorical plot except anomaly score from Fig. 1 to 12. Authors don’t have to worry about class imbalance because model on the training dataset while training is done.

TABLE I. Column Names

Sr. No.	Column name
1	app_cpu_sys_netdata
2	app_cpu_sys_apps.plugin
3	app_cpu_sys_tc-qos-helper
4	app_cpu_sys_go.d.plugin
5	app_cpu_sys_logs
6	app_cpu_sys_ssh
7	app_cpu_sys_system
8	app_cpu_sys_kernel
9	app_cpu_sys_other
10	app_cpu_usr_netdata
11	app_cpu_usr_apps.plugin
12	app_cpu_usr_tc-qos-helper
13	app_cpu_usr_go.d.plugin
14	app_cpu_usr_logs
15	app_cpu_usr_ssh
16	app_cpu_usr_system
17	app_cpu_usr_kernel
18	app_cpu_usr_other
19	app_mem_netdata
20	app_mem_apps.plugin
21	app_mem_tc-qos-helper
22	app_mem_go.d.plugin
23	app_mem_ssh

24	app_mem_cron
25	app_mem_system
26	app_mem_other
27	app_mem_X
28	app_soc_ssh
29	app_soc_system
30	app_soc_other
31	app_soc_X
32	sda_writes
33	ops_sda_writes
34	utilization
35	packets_received
36	packets_sent
37	packets_delivered
38	socket_used
39	udp_packets_received
40	udp_packets_sent
41	avail
42	Dirty
43	Writeback
44	sys_cpu_softirq
45	sys_cpu_user
46	sys_cpu_system
47	sys_cpu_iowait
48	switches
49	interrupts
50	io_out
51	ip_received
52	ip_sent
53	net_received
54	net_sent
55	pgio_out
56	Running
57	Free
58	Used
59	Cached
60	Buffers
61	Uptime
62	Label
63	anomaly_score

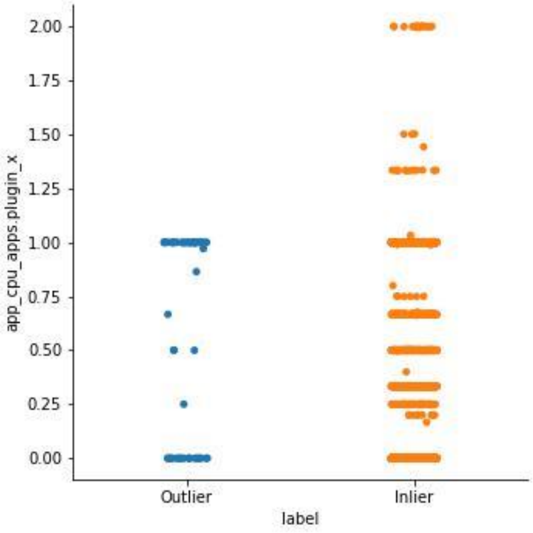


Fig. 1. Categorical plot of app_cpu_apps.plugin_x.

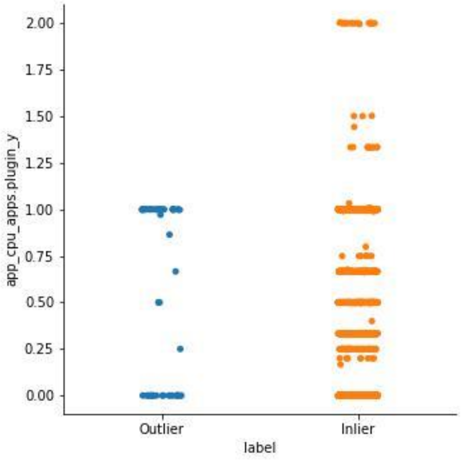


Fig. 2. Categorical plot of app_cpu_apps.plugin_y.

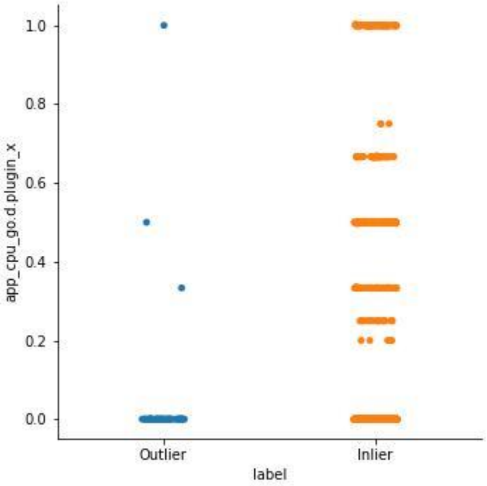


Fig. 3. Categorical plot of app_cpu_go.plugin_x.

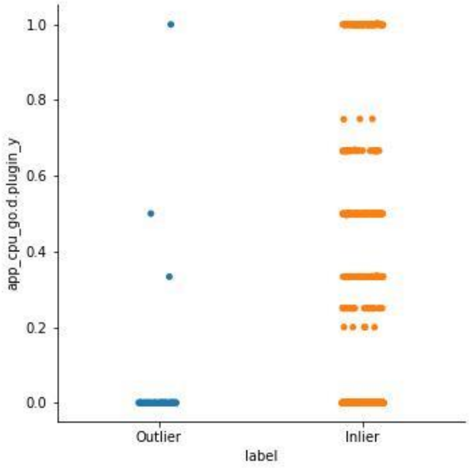


Fig. 4. Categorical plot of app_cpu_go.plugin_y.

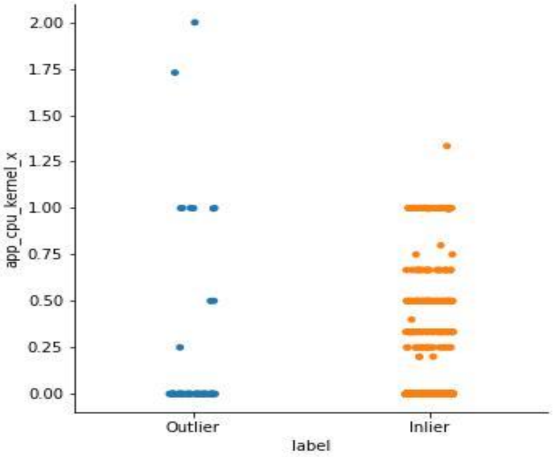


Fig. 5. Categorical plot of app_cpu_kernel_x.

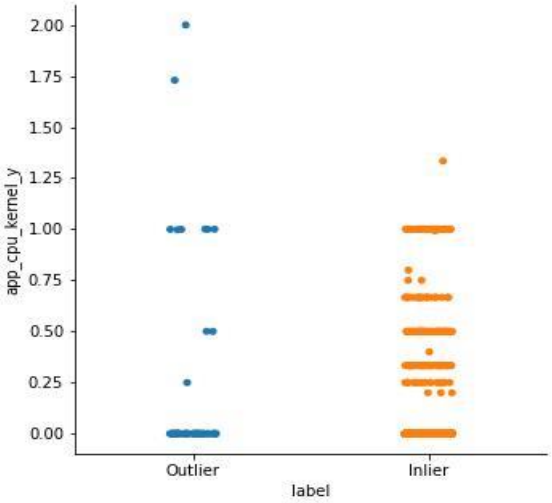


Fig. 6. Categorical plot of app_cpu_kernel_y.

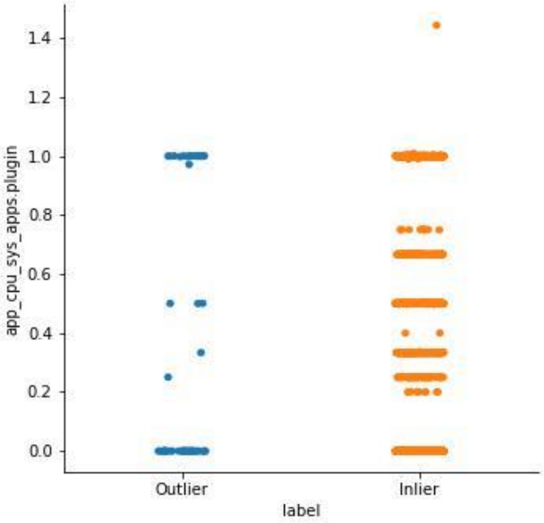


Fig. 7. Categorical plot of app_cpu_sys_apps.plugin.

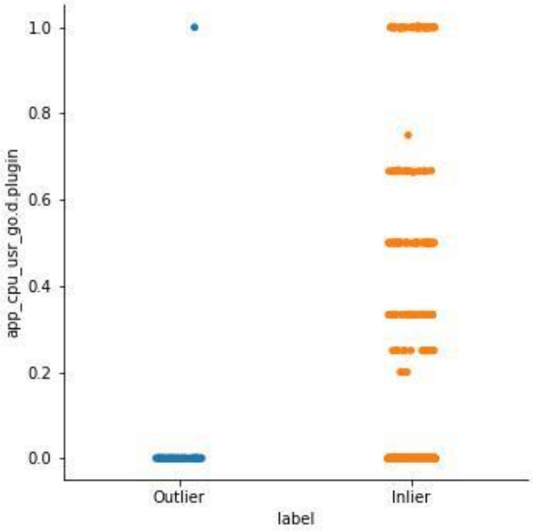


Fig. 8. Categorical plot of app_cpu_usr_go.d.plugin.

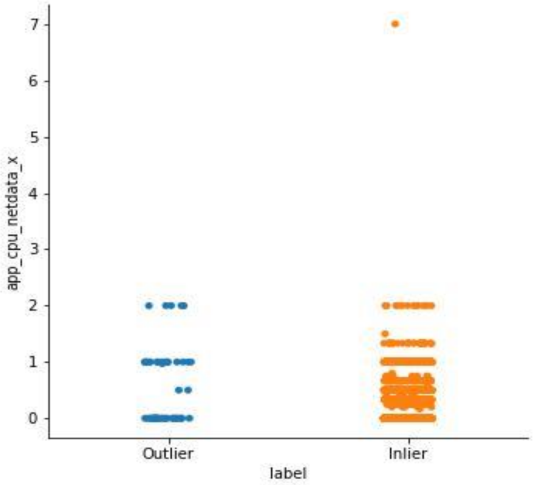


Fig. 9. Categorical plot of app_cpu_netdata_x.

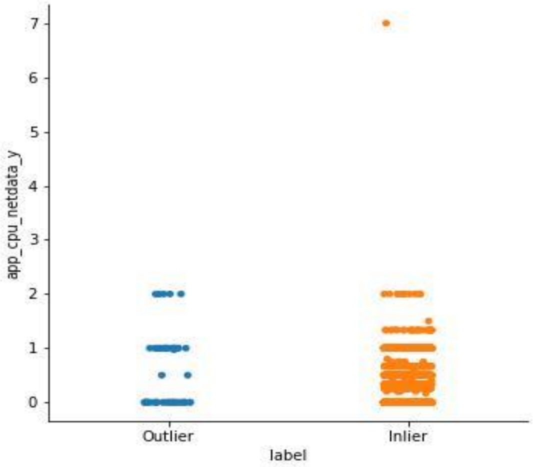


Fig. 10. Categorical plot of app_cpu_kernel_y.

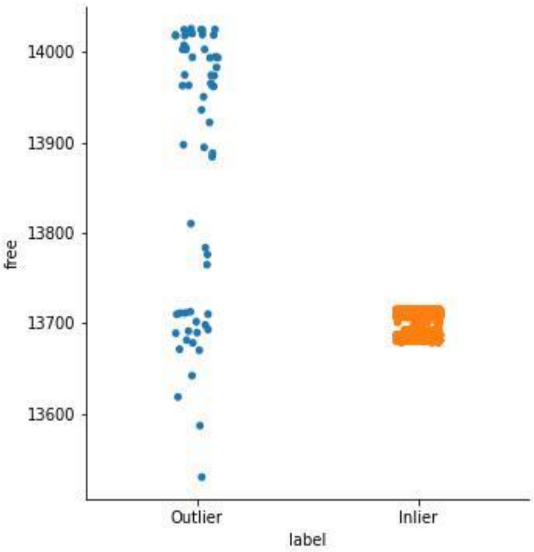


Fig. 11. Categorical plot of free.

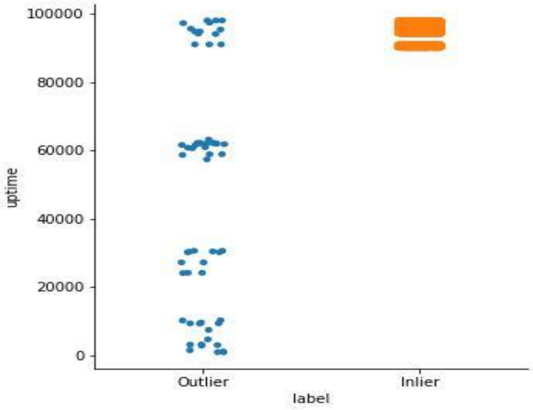


Fig. 12. Categorical plot of uptime.

Authors then standardize the dataset as there is need to perform PCA on it. PCA is applied by a keeping 98% variance. After applying PCA, Dataset have 38 columns in the original dataset and 18 columns in the dataset on which columns were removed having standard deviation less than 0.3. This work plots the first two components of the new dataset on a 2D axis as shown in Fig. 13. Authors also perform PCA on the original dataset.

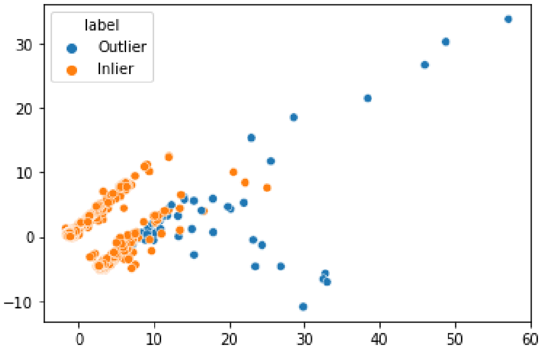


Fig. 13. First two components of the dataset after PCA.

This work, as shown in Fig. 14, plots the first three components of the new dataset on 3D axes. Here authors can clearly see a separation between inliers and outliers.

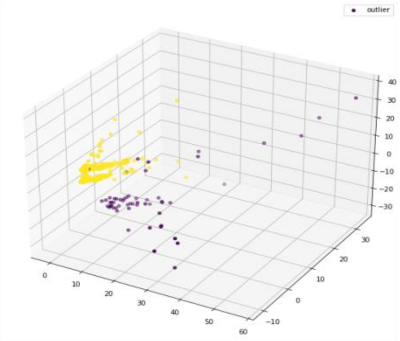


Fig. 14. First three components of the dataset after PCA.

Now authors have two datasets, one with all columns and another with columns left after removing columns with a standard deviation less than 0.3. Authors apply PCA to both datasets. This work split both datasets into three sets, train, test, and cross-validation set. The training set consists of 4000 inliers. The testing set consists of 586 inliers and 30 outliers. The cross-validation set consists of 400 inliers and 30 outliers.

IV. Gaussian Model for Anomaly Detection

A. Univariate Gaussian Model

Gaussian distribution is a continuous probability density function for a real-valued random variable in statistics. It is given by Eq. (1).

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (1)$$

Where f(x) is the probability density function, μ is the mean and σ is the standard deviation.

This work calculates the mean and standard deviation of each column of both datasets and model a Gaussian distribution on all columns. The final probability is calculated by taking the product of the probabilities of all columns. Negative logarithms of probabilities are plotted as histograms as shown in Fig. 15 to 19. Fig. 15 shows probabilities of train inliers. Fig. 16 shows probabilities of test set. Fig. 17 shows probabilities of test set with columns having standard deviation less than 0.3 removed. Fig. 18 shows probabilities of cross validation set. Fig. 19 shows probabilities of cross validation set with columns having standard deviation less than 0.3 removed.

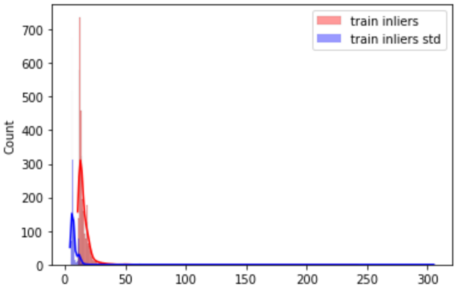


Fig. 15. Probabilities of train inliers.

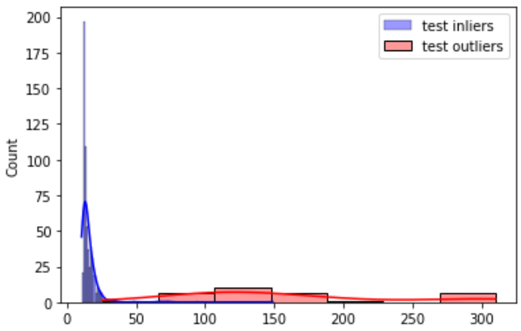


Fig. 16. Probabilities of test set.

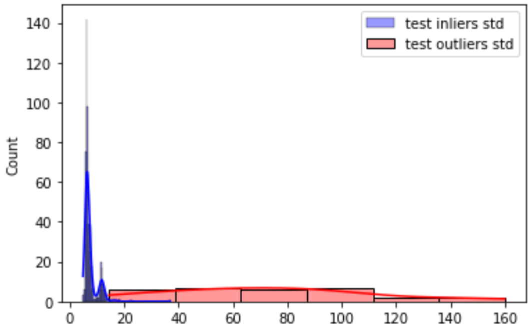


Fig. 17. Probabilities of test set with columns having standard deviation less than 0.3 removed.

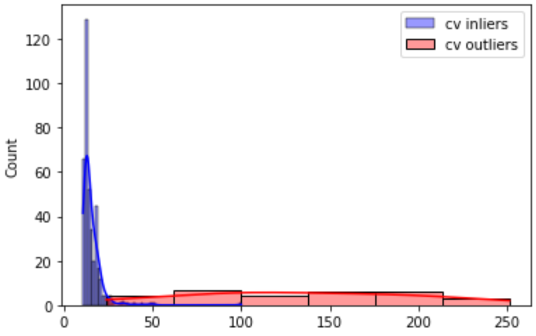


Fig. 18. Probabilities of cross validation set.

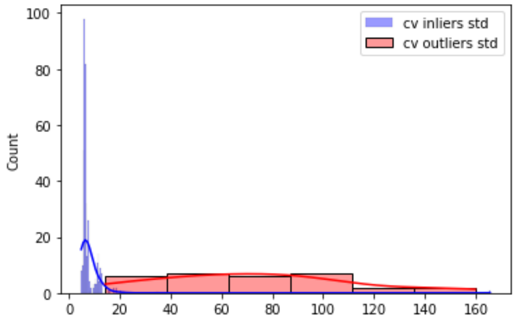


Fig. 19. Probabilities of cross validation set with columns having standard deviation less than 0.3 removed.

Authors set a threshold probability value to classify the test and cross-validation set. Different thresholds are set and accuracy is observed. Table II and III show accuracies for the original dataset.

TABLE II. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR TRAIN AND TEST INLIER SET

Threshold	Train accuracy	Test inlier accuracy
1e-10	0	0
1e-15	63.625	64.16
1e-20	89.725	89.078
1e-25	95.925	96.075
1e-30	97.825	97.78
1e-35	98.625	98.12
1e-40	99.05	98.63
1e-45	99.175	98.63

TABLE III. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR TEST OUTLIER, CROSS-VAL INLIER AND OUTLIER SET

Threshold	Test outlier accuracy	Cross Val inlier accuracy	Cross Val outlier accuracy
1e-10	100	0	100
1e-15	100	66.25	100
1e-20	100	91.75	100
1e-25	100	96.75	96.66
1e-30	96.66	98.0	93.33
1e-35	96.66	98.5	90.0
1e-40	93.33	99.25	90.0
1e-45	93.33	99.25	90.0

Table IV and V shows accuracies for the dataset whose columns were removed which had a standard deviation of less than 0.3.

TABLE IV. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR TRAIN INLIERS, TEST INLIERS AND TEST OUTLIERS SET

Threshold	Train accuracy	Test inlier accuracy	Test outlier accuracy
1e-10	82.475	83.95	100
1e-15	98.475	98.805	96.66
1e-20	99.4	99.65	93.33
1e-25	99.575	99.65	93.33
1e-30	99.6	99.65	99.65
1e-35	99.675	99.658	90.0
1e-40	99.725	99.82	90.0
1e-45	99.725	99.82	90.0

TABLE V. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR CROSS-VAL INLIERS AND OUTLIERS

Threshold	Cross Val inlier accuracy	Cross Val outlier accuracy
1e-10	86.25	96.66
1e-15	98.25	96.66
1e-20	99.5	90
1e-25	100	86.66
1e-30	100.0	86.66
1e-35	100.0	83.33
1e-40	100.0	80.0
1e-45	100.0	76.66

B. *Multivariate Gaussian Model*

The multivariate normal distribution, multivariate Gaussian distribution, or joint normal distribution are expansions of the one-dimensional normal distribution to higher dimensions in probability theory and statistics. It models the probability in one shot instead of calculating individual probabilities and multiplying them. Multivariate Gaussian distribution is given by the Eq. (2).

$$\mathcal{N}(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}}|\Sigma|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^{T}\right\}(2)$$

Where μ is the length-d row vector of means of all columns, ∑ is the covariance matrix of shape d x d. d is the number of features.

Authors calculate the mean and covariance matrices of both datasets to model a multivariate Gaussian distribution. Authors set a threshold value and classify the dataset between inlier and outlier and calculate accuracies for various threshold values. Negative logarithms of probabilities are plotted as a histogram as shown in Fig. 20 to 24. Fig. 20 shows probabilities of train inliers. Fig. 21 shows probabilities of test inliers and test outliers. Fig. 22 shows probabilities of test inliers and test outliers with columns having standard deviation less than 0.3 removed. Fig. 23 shows probabilities of cross-val inliers and cross-val outliers. Fig. 24 shows probabilities of cross-val inliers and cross-val outliers with columns having standard deviation less than 0.3 removed.

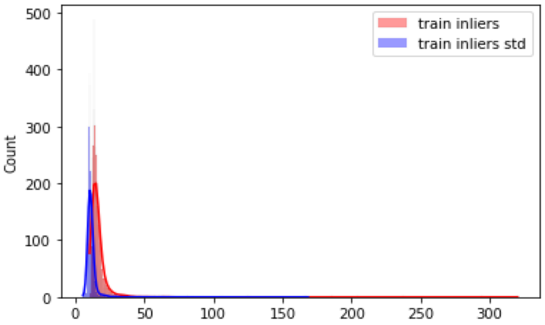


Fig. 20. Probabilities of train inliers.

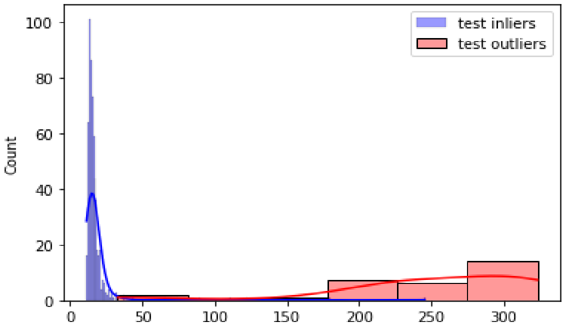


Fig. 21. Probabilities of test inliers and test outliers.

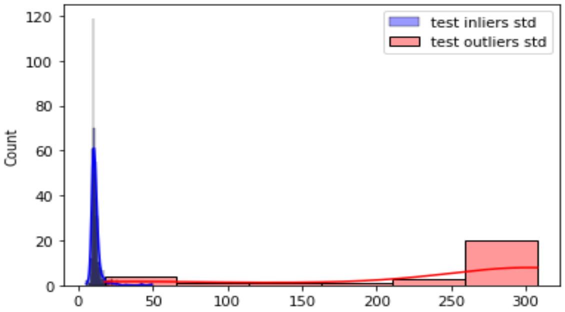


Fig. 22. Probabilities of test inliers and test outliers with columns having standard deviation less than 0.3 removed.

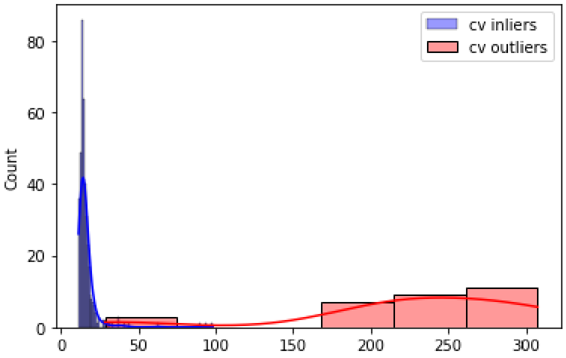


Fig. 23. Probabilities of cross-val inliers and cross-val outliers.

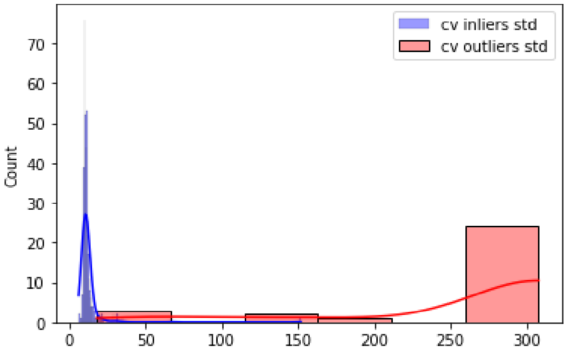


Fig. 24. Probabilities of cross-val inliers and cross-val outliers with columns having standard deviation less than 0.3 removed.

Table VI and VII show accuracies for the original dataset.

TABLE VI. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR TRAIN INLIERS, TEST INLIERS, AND TEST OUTLIERS SET

Threshold	Train accuracy	Test inlier accuracy	Test outlier accuracy
1e-10	0.0	0.0	100.0
1e-15	29.2	30.88	100.0
1e-20	83.1	84.47	100.0
1e-25	93.075	94.02	100.0
1e-30	96.325	97.26	100.0
1e-35	97.575	97.78	93.33
1e-40	98.5	98.80	90.0
1e-45	98.8	99.146	90.0

TABLE VII. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR CROSS-VAL INLIERS AND CROSS-VAL OUTLIERS SET

Threshold	Cross Val inlier accuracy	Cross Val outlier accuracy
1e-10	0.0	100.0
1e-15	30.0	100.0
1e-20	82.5	100.0
1e-25	94.0	100.0
1e-30	96.0	100.0
1e-35	97.5	96.66
1e-40	98.25	96.66
1e-45	98.75	96.66

Table VIII and IX show accuracies for the dataset whose columns were removed which had a standard deviation of less than 0.3.

TABLE VIII. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR TRAIN INLIERS, TEST INLIERS, AND TEST OUTLIERS

Threshold	Train accuracy	Test inlier accuracy	Test outlier accuracy
1e-10	16.675	16.21	100.0
1e-15	94.125	93.68	100.0
1e-20	97.2	97.95	96.66
1e-25	98.625	99.48	93.33
1e-30	99.125	100.0	93.33
1e-35	99.325	100.0	93.33
1e-40	99.4	100.0	93.33
1e-45	99.45	100.0	93.33

TABLE IX. VARIATION OF ACCURACY WITH DIFFERENT THRESHOLD VALUES FOR CROSS-VAL INLIERS AND CROSS-VAL OUTLIERS

Threshold	Cross Val inlier accuracy	Cross Val outlier accuracy
1e-10	18.0	100.0
1e-15	93.25	100.0
1e-20	97.25	96.66
1e-25	98.0	93.33
1e-30	98.75	93.33
1e-35	99.25	90.0
1e-40	99.5	90.0
1e-45	99.5	90.0

V. CONCLUSION

This work states that univariate and multivariate Gaussian models for anomaly detection are successfully created. Data imbalance is not an issue here because these models fit on the train set and this work uses a threshold to predict inliers and outliers. This work examines the trend between various threshold values and accuracies. The proposed method, a Gaussian model to forecast the likelihood of an attack occurring based on certain system parameters uses a univariate and a multivariate Gaussian model on the training dataset and examines accuracies for various threshold values. It also addresses the challenge of class imbalance in anomaly detection situations. This method presents the successful creation of univariate and multivariate Gaussian models for anomaly detection. The data imbalance is not an issue in these models because they fit on the train set and use a threshold to predict inliers and outliers. The study also examines the relationship between various threshold values and accuracies. For univariate Gaussian model variation of accuracy with different threshold values ranges up to 99.175 percent and for train accuracy up to 98.6 percent for test inlier accuracy and up to 100 percent for test outlier accuracy. For multivariate Gaussian model variation of accuracy with different threshold values ranges up to 99.45 for train accuracy, up to 100 for test inlier accuracy and up to 100 for test outlier accuracy with validation.

Future work is about using deep learning techniques such as auto encoders. Machine learning is revealing a plethora of potential for cybersecurity aficionados to explore as more and more data is gathered, specifically with the data that they already own. When this work talks about escalating warfare in the internet age, timely automated identification of any threats or suspicious conduct can avoid a number of mistakes from occurring.

REFERENCES

[1] Mohammad-Mahdi Bazm, Thibaut Sautereau, Marc Lacoste, Mario Sudholt, Jean-Marc Menaud, ¨ ”Cache-Based Side-Channel Attacks Detection through Intel Cache Monitoring Technology and Hardware Performance Counters”, FMEC2018 - The Third IEEE International Conference on Fog and Mobile Edge Computing, Apr 2018, Barcelona, Spain. IEEE, pp.1-6. ¡hal-01762803¿

[2] M. Schwarz, S. Weiser, D. Gruss, C. Maurice, and S. Mangard, “Malware guard extension: Using sgx to conceal cache attacks”, arXiv preprint arXiv:1702.08719,2017.

[3] C. Disselkoen, D. Kohlbrenner, L. Porter, and D. Tullsen, “Prime+abort: A timer-free high-precision l3 cache attack using intel TSX”, in 26th USENIX Security Symposium (USENIX Security 17), (Vancouver, BC), pp. 51–67, USENIX Association, 2017.

[4] M. Chiappetta, E. Savas, and C. Yilmaz, “Real time detection of cache-based side-channel attacks using hardware performance counters”, Applied Soft Computing, vol. 49, pp. 1162–1174, 2016.

[5] Ziqi Wang, Rui Yang, Xiao Fu, Xiaojiang Du, and Bin Luo, ”A shared memory based cross-vm side channel attacks in iaas cloud”, In Computer Communications Workshops (INFOCOM WKSHPS), IEEE Conference on, pages 181–186. IEEE, 2016.

[6] A. Valdes, K. Skinner, Adaptive, ”Model-Based Monitoring for Cyber Attack Detection”, International Workshop on Recent Advances in Intrusion Detection, Berlin, Heidelberg, 2000.

[7] Deri, Luca, and Alfredo Cardigliano. "Using cyberscore for network traffic monitoring." In 2022 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 56-61. IEEE, 2022.

[8] P. Patil and R. Ingle, "Meta-ensemble based classifier approach for attack detection in multi-tenant distributed systems," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154077.

[9] A. Valdes, K. Skinner, Adaptive, ”Model-Based Monitoring for Cyber Attack Detection”, International Workshop on Recent Advances in Intrusion Detection, Berlin, Heidelberg, 2000.

[10] Michael Hilker, Christoph Schommer,” Description of bad-signatures for network intrusion detection”, n Conf.s in Research and Practice in Information Technology Series, vol. 54, ACSW, 2006, pp. 175–182

[11] N. Stakhanova, M. Couture, A. A. Ghorbani, ”Exploring network-based malware classification”, 2011 6th International Conference on Malicious and Unwanted Software

[12] F. Iglesias, T. Zseby, ”Analysis of network traffic features for anomaly detection”, Mach. Learn.101(1-3), 59–84 (2014).

[13] M. Z. Rafique, P. Chen, C. Huygens, W. Joosen, ”Evolutionary algorithms for classification of malware families through different network behaviors”, Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, Pages 1167-1174.

[14] W. Dai, Q. Yang, G. -R. Xue, Y. Yu, ”Boosting for transfer learning”, 24th International Conf. on Machine Learning(ICML) 2007

[15] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, ”Domain adaptation via transfer component analysis”, IEEE Transaction on Neural Netw.22(2), 199–210, 2011.

[16] Shi, Q. Liu, W. Fan, P. S. Yu, R. Zhu, ”Transfer learning on heterogenous feature spaces via spectral transformation”, IEEE International Conf. on Data Mining (ICDM), 2010.

[17] D. Bekerman, B. Shapira, L. Rokach, A. Bar, ”Unknown malware detection using network traffic classification”, 2015 IEEE Conference on Communications and Network Security (CNS).

Cloud Service Composition using Firefly Optimization Algorithm and Fuzzy Logic

Wenzhi Wang, Zhanqiao Liu
School of Distance Education, Jiaozuo University, Jiaozuo, 454000, China

***Abstract*—Cloud computing involves the dynamic provision of virtualized and scalable resources over the Internet as services. Different types of services with the same functionality but different non-functionality features may be delivered in a cloud environment in response to customer requests, which may need to be combined to satisfy the customer's complex requirements. Recent research has focused on combining unique and loosely-coupled services into a preferred system. An optimized composite service consists of formerly existing single and simple services combined to provide an optimal composite service, thereby improving the quality of service (QoS). In recent years, cloud computing has driven the rapid proliferation of multi-provision cloud service compositions, in which cloud service providers can provide multiple services simultaneously. Service composition fulfils a variety of user needs in a variety of scenarios. The composite request (service request) in a multi-cloud environment requires atomic services (service candidates) located in multiple clouds. Service composition combines atomic services from multiple clouds into a single service. Since cloud services are rapidly growing and their Quality of Service (QoS) is widely varying, finding the necessary services and composing them with quality assurances is an increasingly challenging technical task. This paper presents a method that uses the firefly optimization algorithm (FOA) and fuzzy logic to balance multiple QoS factors and satisfy service composition constraints. Experimental results prove that the proposed method outperforms previous ones in terms of response time, availability, and energy consumption.**

Keywords*—Cloud computing; service composition; QoS; firefly algorithm; fuzzy logic

I. Introduction

Recent rapid growth in artificial intelligence [1, 2], machine learning [3], optical networks [4, 5], smart grids [6], cloud computing [7], 5G connectivity [8], Blockchain [9, 10], and Internet of Things (IoT) [11, 12] have resulted in an explosion of data in almost all fields of engineering and commerce. With cloud computing, large-scale applications can be deployed quickly and efficiently, affordably on a per-use basis [13]. Cloud computing relies on virtualization to share resources among customers [14]. Virtualization technology enables cloud data centres to dynamically share physical resources, allowing multiple applications to run on different platforms known as Virtual Machines (VMs) [15]. As a result of virtualization, cloud service providers can ensure the quality of service (QoS) distributed among different users while achieving maximum resource utilization and minimal power consumption [16]. Cloud environments deliver services tailored to users' requirements. Existing services are combined into composite services to provide users with value-added services. As cloud computing has proliferated, more providers are providing similar functional cloud services but with diverse nonfunctional characteristics [17]. Consequently, cloud service composition must consider QoS awareness in choosing appropriate services and combining them to meet users' expectations, known as QoS-aware cloud service composition [18]. Cloud services can potentially encapsulate many resources as new technologies develop, and combinatorial optimization problems can be transformed into QoS-aware cloud service composition problems [19].

Cloud services currently available to consumers offer similar functionality at varying QoS levels. Cloud services offer different levels of QoS for a given task, so ranking these services based on QoS makes it easier for users to choose cloud services [20]. There might be a service rated best under one QoS parameter yet rated worst under another. Various QoS parameters can be used to rank the performance of a service. Cloud service composition contexts prioritize QoS parameters differently [21]. In choosing a cloud service, all QoS parameters must be taken into account without overlooking the influence of a primary QoS factor. QoS-aware cloud service composition generally considers only one or two QoS factors and ignores balancing QoS parameters or satisfying connectivity constraints [22]. In this paper, we present a novel method based on firefly optimization (FOA) and fuzzy logic to balance multiple QoS parameters and satisfy the connectivity constraints of service composition. The rest of the paper appears as follows. The next section summarizes related works. The proposed strategy is discussed in detail in Section III. Section IV reports the simulation results. Section V concludes the paper with a discussion of future directions. Generally, this work contributes to:

- A model maturity metric is introduced in this paper in order to provide a comprehensive evaluation of the simulation model lifecycle in cloud environments.
- Based on the cooperation relationship between model services, this paper dynamically calculates the maturity score of the combined model.
- A new algorithm based on FOA and fuzzy logic is proposed in this paper for the composition and optimization of cloud model services.

II. Related Work

Kritikos and Plexousakis [23] presented an approach for composing cloud services that optimally satisfy various user requirements while simultaneously composing different cloud services. Cloud application design tools do not simultaneously support quality, deployment, security, placement, or cost requirements. In addition, the proposed approach considers a type of design choice currently not considered in the literature. Huo, Zhuang [24] propose a novel technique for cloud service composition that formalizes service composition as a nonlinear integer programming problem by incorporating a time attenuation function. Additionally, the discrete gbest-guided artificial bee colony algorithm is presented, representing the exploration of bee hives for food in search of the optimal service composition strategy. Based on experiments, it appears that the time attenuation function can improve the quality of services by making them more consistent with their characteristics at present. In comparison with other algorithms, especially for large-scale data, this algorithm provides the best possible solution in a short amount of time.

Liu, Chu [25] use social learning optimization algorithm (SLOA) to resolve the QoS-aware cloud service composition problem. Improvements to differential evolutionary algorithms and SLOA are incorporated into micro-space and learning spaces. Performance comparison and experimental results demonstrate the efficacy of the SLOA. This work will improve search capabilities and convergence rates by extending the theory of the swarm intelligence optimization algorithm and exploring a novel swarm intelligence optimization model. The challenge of correlation-aware QoS in networks and cloud services is addressed by Huang, Li [26]. This problem is formulated as a multi-constraint optimal path problem, and a novel approach is proposed for solving it. The proposed algorithm is evaluated using extensive simulation. The proposed algorithm yields superior service composition solutions with improved QoS guarantees by taking into account the QoS correlations between the different service types.

Karimi, Isazadeh [27] employed the genetic algorithm to optimize service level agreements globally. Service clustering was applied to reduce the search space, and association rules were applied to a composite service based on their histories to improve service composition efficiency. As compared with similar related works, the proposed method demonstrated higher efficiency. Low-cost access to a simplified, centralized platform or resource is provided by cloud computing. This type of computing allocates resources based on individual needs. However, resources need to be allocated efficiently to meet the expanding needs of cloud users. Service providers are responsible for distributing and sharing resources effectively, preventing resource waste. Furthermore, the user receives the appropriate service based on their request, with the cost of the resource being optimized. Singh, Juneja [28] present an algorithm for automated service composition based on agents, which addresses both service requests and automatic service composition. The algorithm searches for the best services and reduces the cost of on-demand virtual machines.

Wang, Zhou [29] analyzed the relationship between energy consumption, network resource consumption, and QoS performance in a service composition process. An approach to green service composition is then presented. The system prioritizes composite services that run on the same physical server, virtual machine, or edge switch. Green service composition optimization minimizes energy and network resource consumption on physical servers and switches in cloud data centers. Based on experiments, the proposed approach reduces energy consumption by 20-50 percent and network resource consumption by 10-50 percent compared to other approaches. Jian, Li [30] presents an algorithm incorporating the two-order oscillating equation and the historical positions of birds. It enhances bird diversity, strengthens global search algorithms, and improves bird feeding and migration dynamics. Based on simulation results with and without local QoS restrictions, the algorithm minimizes overall QoS restriction execution times. The new eagle search procedure was developed by Jin, Lv [31] by integrating regular mutations with an improved whale optimization algorithm. In order to verify the performance of the new approach, a variety of benchmark functions and problems of cloud service composition are used. The proposed method outperforms the other methods, according to experiments.

Studies presented above suggest a variety of solutions to the problem of service composition. The majority of researchers, however, use a simple additive weighting proposal in order to combine multiple aspects of QoS into a single-objective function. Service composition is primarily a multi-objective problem, in which multiple, often conflicting attributes of QoS must be optimized simultaneously. A number of proposals have been considered to address the problem of service composition from a multi-objective perspective, including a tri-objective service composition [32], a bi-objective service composition [33], and a bi-objective service composition from an energy and global quality of service utility's perspective [14]. However, these studies only identify up to three representative objectives by assigning priorities to QoS factors or reducing a large proportion of QoS factors within two or three objectives. The literature rarely examines service composition scenarios with four or more objectives. The QoS criteria for many applications have expanded to include security, maintainability, reputation, energy consumption, carbon emissions, and ecological impact. With a range of trade-off options available, the decision-making process becomes more flexible. Service composition is strictly a multi-objective optimization problem if we treat all QoS attributes equally.

III. Proposed Method

This section will provide a new IoT-enabled cloud service composition method based on FOA and fuzzy logic systems. We aim to improve the QoS parameters, reduce energy consumption, extend the network's life, improve packet delivery rates, and decrease delays. Due to two advantages of automatic segmentation of the network and diversity in solutions, the FOA can perform better than other algorithms in finding the optimal solution to optimization problems. In the following, we have discussed the network model and problem assumptions, optimization models, objective function, and finally, the proposed method.

A. The Problem Expression and Formulation

Under one set of QoS parameters, a cloud service may be rated as the best, whereas under another set of QoS parameters, it may be rated as the worst. Various QoS parameters can be used to rank cloud service performance. Cloud services that rank highest under one QoS parameter, such as reliability, may rank lowest under another QoS parameter, such as response time. Therefore, QoS parameters are prioritized differently depending on the composition context. A cloud service that is optimal under one QoS parameter may fail if that cloud service is not optimal under another QoS parameter. Therefore, including any of these cloud services as candidate services in the composition will have an impact on the overall performance, since these cloud services perform poorly under parameters other than the primary QoS parameter.

QoS-aware services composition problem is finding a set of candidate services with different functional features that comply with the determined limitations by the user and optimize an objective function. This problem is explained in this section.

- A service composition request as a workflow that is modeled using a Directed Acyclic Graph (DAG), G=(V, E).
- $V = \{T_1.T_2.T_3.\dots.T_n\}$ That n is the number of tasks in the workflow.
- E is the set of edges showing the priority of executing the works.
- Each T_i for $0 \le i \le n$ is a set of candidate services in the workflow.
- $CS_i = \{CS_i^1.CS_i^2 \dots.CS_i^m\}$: That CS_i^J for $1 \le j \le m$ is a candidate service.
- M_i is the number of existing candidate services for T_i.
- Each candidate service CS_i^J has a set of different QoS information.
- $QOS = \{Q_1.Q_2.Q_3.\dots.Q_n\}$ That Q_i for $1 \le i \le k$ shows a QoS feature of cloud services.

Considering the above cases, the goal of the QoS-aware service composition problem is near optimal so that:

$$\forall j = 1 \dots k \begin{cases} \sum_{i=0}^{n} S_i.Q_i < C_j & if\ Q_j\ is\ additive \\ \prod_{i=1}^{n} S_i.Q_i > C_j & if\ \ Q_j\ is\ multiplicative \end{cases} \tag{1}$$

Composition and service selection problem in all computational platforms like cloud environment and IoT is appropriate to meet a user or system requirements that should provide QoS requirements in addition to the user's needs. A fitting function in a service composition problem, like any optimization problem, is used to determine whether the selected service is valuable. This function's output determines the efficiency of the selected services for composition.

Generally, the most important factor for the development and scalability of IoT is considering energy consumption limitations. Hence, energy saving is one of the important challenges in these networks because energy sources in IoT nodes are limited, and changing the energy sources of the nodes on large scales is practically impossible. Based on the mentioned notes, energy consumption is critical for the candidate services' host nodes. Thus, each candidate service must propose its required energy to the service provider module to select one of the most energy-efficient nodes for service composition. Two variables are used in the proposed energy model for the model definition. The first one is the amount of remaining energy and the second one is the amount of consumed energy of the node, which is the amount of consumed energy when executing the user's request. Based on the model, $RE(cs_{ij})$ is the remaining energy of the candidate service host (cs_{ij}), and $CE(cs_{ij})$ is the consumed energy when running the candidate's request (cs_{ij}). Hence, the amount of remaining energy of the host is calculated using 3:

$$RE(cs_{ij}) = CDE(cs_{ij}) - E_{th}(cs_{ij}) \tag{2}$$

where $CDE(cs_{ij})$ is the level of energy in the IoT node equipped with the battery, which is the host of cs_{ij} service, and $E_{th}(cs_{ij})$ is the minimum threshold energy of IoT so that if a node's energy is less than the threshold, the node cannot work in the network and dies.

Since the energy model in this thesis is based on service-oriented calculations, the consumed energy by the service candidate when running cs_{ij} is estimated using equation (3). We assume that the energy consumption of the candidate service cs_{ij} is constant because the services are run on the same IoT platform and use the same resources to answer the service. Moreover, they consume the same amount of energy when data transmission and reception.

$$CE(cs_{ij}) = ECR(cs_{ij}) \times T(cs_{ij}) \tag{3}$$

where $CE(cs_{ij})$ is the consumed energy, and $T(cs_{ij})$ is the run time of the candidate service (cs_{ij}). Finally, the consumed energy of the cs_{ij} candidate service is defined as an EP and calculated by Eq. (4). The profile is estimated based on the ratio of the cs_{ij} candidate service consumed energy to its remaining energy.

$$EP(cs_{ij}) = \frac{CE(cs_{ij})}{RE(cs_{ij})} \tag{4}$$

Based on Eq. (3) to (4), the lower amount of $EP(cs_{ij})$ in a node means that the node is optimal for cs_{ij} service at time T_i. Finally, the EP of service composition is computed using Eq. (5). Hence, the consumed energy of the service composition path x^i is as follows:

$$EP(x) = \sum_{i=1}^{n} EP(x^i | x^i \in s^h) \tag{5}$$

where $s^h$is the set of IoT nodes with a battery. The nodes with lower energy than the threshold energy is removed from the set.

- Response Time: the required time duration for a request transmission and its response reception.

- Availability: the number of successful calls to the total number of calls ratio.
- Successability: the number of responses to the number of request messages ratio.
- Reliability: the number of faulty messages to the total number of messages ratio.
- Latency: the duration of a request process by a server.

In the selection and composition of IoT services, the composing services can be composed and meet the users' needs. Hence, calculating the composed QoS features summation is important, and each serial, parallel, switch, and loop pattern uses its formulas. The requested works and the proposed services are considered a graph for service composition. Thus, the following equations are used for the QoS parameter values calculation in the composing services. Eq. (6) calculated the response time parameter, and Eq. (7) to (11) are used to calculate delay, energy, reliability, availability, and successability parameters in the composing services, respectively.

$$Q_{Responce\ Time}(s) = \sum_{i=1}^{n} Q_{RT}\ (s) \tag{6}$$

$$Q_{Latency}(s) = \sum_{i=1}^{n} Q_{L}\ (s) \tag{7}$$

$$Q_{Energy\ Consumption}(s) = \sum_{i=1}^{n} Q_{EC}\ (s) \tag{8}$$

$$Q_{Reliability}(s) = \prod_{i=1}^{n} Q_{R}(s) \tag{9}$$

$$Q_{Accessability}(s) = \prod_{i=1}^{n} Q_{A}(s) \tag{10}$$

$$Q_{Successability}(s) = \prod_{i=1}^{n} Q_{S}(s) \tag{11}$$

Generally, the fitting function (objective function) is used to determine the value of each solution generated by the optimization problems. This function output determines how much a solution can be useful for meeting the user's or the system's requirements. The fitting function is defined by Eq. (12).

$$Fitness(Sol) = W_1 * Resp + W_2 * Late + W_3 * \frac{1}{Avail} + W_4 * \frac{1}{Succ} + W_5 * \frac{1}{Reli} + W_6 * Energy \tag{12}$$

where W_1, W_2, W_3, W_4, W_5, and W_6 are the positive weights showing the importance of each QoS parameter determined by the user. Eq. (13) and (14) are used for normalizing the parameters' output in the objective function. Eq. (13) is for the minimizer parameters, and (14) is for the maximizer parameters.

$$N_{CS.Q^i} \begin{cases} \frac{Q^i_{max} - CS.Q^i}{Q^i_{max} - Q^i_{min}} & Q^i_{max} \neq Q^i_{min} \\ 1 & Q^i_{max} = Q^i_{min} \end{cases} \tag{13}$$

$$N_{CS.Q^i} \begin{cases} \frac{CS.Q^i - Q^i_{min}}{Q^i_{max} - Q^i_{min}} & Q^i_{max} \neq Q^i_{min} \\ 1 & Q^i_{max} = Q^i_{min} \end{cases} \tag{14}$$

where $CS \cdot Q^i$ is the ith parameter value of QoS related to the CS selected service, $Ncs \cdot Q^i$ is the normalized value, and Q^i_{max} and Q^i_{min} are the maximum and minimum of the ith parameter among all the services.

B. The Proposed Method's Steps

This section has two subsections. The first subsection includes general information about the firefly algorithm and Fuzzy logic, and the proposed method steps are explained in the second subsection. The FOA algorithm was inspired by how fireflies attract mates by producing flashlights. The FOA uses three idealized features of the firefly's flashing light to produce an optimal solution. Fireflies are unisex and attract other fireflies irrespective of their gender.

Further, when two fireflies are distant from each other, their attractiveness decreases, which is directly proportional to their brightness. Finally, the firefly flashing light is incorporated into the optimization process as an objective function [34, 35]. Fuzzy logic is an approach to variable processing that allows multiple possible truth values to be processed through the same variable. Fuzzy logic attempts to solve problems with an open, imprecise spectrum of data and heuristics that makes it possible to obtain an array of accurate conclusions. The fuzzy logic architecture consists of the following components:

- **Rule base:** This contains the rules and membership functions that regulate or control decision-making in the fuzzy logic system. It also contains the IF-THEN conditions for conditional programming and controlling the system.
- **Fuzzifier:** This component transforms raw inputs into fuzzy sets. The fuzzy sets proceed to the control system, where they undergo further processing.
- **Inference engine:** This tool establishes the ideal rules for a specific input. It then applies these rules to the input data to generate a fuzzy output.
- **Defuzzifier:** This component transforms the fuzzy sets into an explicit output (in the form of crisp inputs). Defuzzification is the final stage of a fuzzy logic system.

The proposed method uses the fuzzy logic system and firefly optimization algorithm for optimal service composition. Since the firefly algorithm has a low speed for local search, Fuzzy logic is used to ease finding the optimal composition in terms of the problem objectives. The firefly algorithm can find and compose more optimal services in less time by applying fuzzy values for each QoS parameter. The basis of the work is as follows: first, a firefly is assigned to each service. Then the fireflies create paths based on their movement pattern and

Luciferin value to find the optimal service composition. Finally, the best services are used for composition by evaluating the route found by the fireflies. The Fuzzy values for each service in the Luciferin function are saved for each firefly. The fireflies find the best composition by applying the Fuzzy values with the neighbour nodes' Luciferin.

1) First step: generating primary population: The primary population is generated in this step using random distribution. The IoT nodes generated based on the geographical positions are distributed in the objective area. After the distribution of nodes, the parameters' initial values, like energy, Luciferin value, and the nodes transmission's radius, are assigned to the parameters. Moreover, data related to each QoS parameter are read from the dataset and put in the parameters.

2) Second step: QoS parameters fuzzification: The proposed method's first step includes fuzzifying each IoT service QoS information. This information includes response time, availability, successability, reliability, energy, and delay. These values are the Fuzzy system inputs in this step, in which a Fuzzy value is generated based on the Fuzzy rules for all inputs to evaluate and select the services based on the value. This Fuzzy value is defined as Low, High, Medium, Very Low, and Very High. Then the system analyzes the information based on the inputs and the written rules in the knowledge base of the fuzzy system and generates the final value after defuzzification. These values then transfer to the firefly algorithm as inputs.

The fuzzy rules are written considering different states of the input parameters and finally are saved in the database. Then using a Fuzzy Inference System (FIS), an output value is generated based on different input states. The Fuzzy rules are written using nested If-Then. The proposed Fuzzy system used 25 rules shown in Table I. Fig. 1 to 5 illustrate the membership functions of the FIS input and output parameters. In the Fuzzy rules table, RT is the response time, D is the delay, En is energy, Av is availability, and Fitness is the output parameter.

After generating different fuzzy values based on the fuzzy rules and input parameters, defuzzification is performed on each fuzzy value. Membership functions of the fuzzy system output variables are presented in Fig. 3 to 11.

3) Third step: using the firefly algorithm to select the best services for the composition: Each firefly in this algorithm is an answer to the problem. All answers have fitting values calculated by the fuzzy system that should be optimized. The first step is creating a list of the proposed services to do the existing tasks in the workflow. Each workflow proposed by the user has n tasks. There is a determined number of services to perform each task, and m is the maximum number of the proposed selected services for each task. At first, an n*1 array of random numbers with size m are generated to initialize the algorithm. The array shows the number of existing services in the accumulator that can perform the workflow tasks. In the next step, the matrix is generated to save the list of the services that can perform the tasks in the workflow. The number of proposed services listed to do the workflow's tasks is generated in the next step. Each presented workflow by the user has m tasks, and there is a determined number of services to do each of the services. The optimization process started with some random solutions. The primary solution is selected among the proposed services for each task. In the firefly algorithm, the primary solution is a service composition, an n*1 array with n tasks in the workflow. The stored number in index i of the array shows the ID of the candidate service that executes the Ti task. The following steps are performed to select and compose the most proper services based on the firefly algorithm steps:

- In the first step, some primary population is generated and distributed randomly. Initialization of Luciferin is applied to each firefly, and the primary solutions are generated. These solutions are valued based on the fitting function value.
- In the second step, the considered proposed QoS parameters as inputs of the problem are fuzzified based on the proposed Fuzzy system. Their values in each round of the algorithm are inserted and updated in the Luciferin matrix as the next step fitting value (the neighbour nodes). Since the parameter values change in each round of the algorithm, the proposed fuzzy system is executed based on the Fuzzy rules in each round, and new values are generated.
- In the third step, the firefly algorithm generates new solutions based on the movement steps. A population of fireflies creates new solutions in the proposed algorithm. The generated solutions of each iteration go to the next iteration for more optimal solutions. A firefly position shows a solution for the composing service problem that the primary position of the fireflies is random. Each firefly generated new solutions in each algorithm iteration based on the fitting value using Eq. (15). The solutions are updated based on their quality. If the new solutions are better than the existing ones, they are replaced.

$$x_{\mathrm{j}}=x_{\mathrm{i}}+\beta_0 . e^{-\gamma r i j^2} \quad (15)$$

After that, all fireflies generate their solutions. A probability is used for selecting the most appropriate solution for each firefly, given by Eq. (16). After calculating the selection probability of each node as the next step, one of the nodes is selected randomly.

$$p_{ij}(\mathrm{t})=\frac{L_j(t)-L_i(t)}{\sum_{k\epsilon N_i(t)} L_k(t)-L_i(t)} \quad (16)$$

In the above equation, $L_i(t)$ is the amount of luciferin of each firefly that is calculated using Eq. (17).

$$L_i(t)=(1-\rho)L_i(t-1)+\gamma J(x_i(t)) \quad (17)$$

In the Eq. (17), $L_i(t)$, $L_i(t-1)$ and $x_i(t)$ are respectively the new value of luciferin, the previous value of luciferin, and the fitness of the location of worm i in iteration t of the algorithm, respectively, and ρ and γ are fixed numbers for modeling Gradual decline and fitness effect on luciferin.

The step-by-step movement of the fireflies continues until selecting the most suitable service for composition. After selection, the generated solution evaluation is performed so that the average distance of the selected solution is calculated rather than other solutions. Then the fitting function value is computed for new solutions. If the new solution is better than the existing solution in the memory, the firefly memory is updated using the new solution. Then the parameters are updated. The proposed method flowchart is illustrated in Fig. 6.

TABLE I. PROPOSED FUZZY RULES

The fuzzy rules
If (RT is VL) and (D is VL) and (En is VL) and (Av is VH) then (Fitness is VH)
If (RT is VL) and (D is VL) and (En is L) and (Av is VH) then (Fitness is H)
If (RT is VL) and (D is VL) and (En is L) and (Av is H) then (Fitness is H)
If (RT is VL) and (D is VL) and (En is VL) and (Av is H) then (Fitness is VH)
If (RT is VL) and (D is VL) and (En is VL) and (Av is M) then (Fitness is H)
If (RT is VL) and (D is VL) and (En is VL) and (Av is L) then (Fitness is L)
If (RT is VL) and (D is VL) and (En is VL) and (Av is VL) then (Fitness is VL)
If (RT is L) and (D is L) and (En is L) and (Av is VH) then (Fitness is VH)
If (RT is L) and (D is L) and (En is L) and (Av is H) then (Fitness is H)
If (RT is L) and (D is L) and (En is L) and (Av is M) then (Fitness is H)
If (RT is L) and (D is L) and (En is L) and (Av is L) then (Fitness is L)
If (RT is L) and (D is L) and (En is L) and (Av is VL) then (Fitness is VL)
If (RT is M) and (D is M) and (En is M) and (Av is VH) then (Fitness is M)
If (RT is M) and (D is M) and (En is M) and (Av is H) then (Fitness is M)
If (RT is M) and (D is M) and (En is M) and (Av is M) then (Fitness is M)
If (RT is M) and (D is M) and (En is M) and (Av is L) then (Fitness is L)
If (RT is M) and (D is M) and (En is M) and (Av is VL) then (Fitness is VL)
If (RT is H) and (D is H) and (En is H) and (Av is VH) then (Fitness is M)
If (RT is H) and (D is H) and (En is H) and (Av is H) then (Fitness is H)
If (RT is H) and (D is H) and (En is H) and (Av is M) then (Fitness is L)
If (RT is H) and (D is H) and (En is H) and (Av is L) then (Fitness is VL)
If (RT is H) and (D is H) and (En is H) and (Av is VL) then (Fitness is VL)
If (RT is VH) and (D is VH) and (En is VH) and (Av is VH) then (Fitness is VL)
If (RT is VH) and (D is VH) and (En is VH) and (Av is H) then (Fitness is VL)
If (RT is VH) and (D is VH) and (En is VH) and (Av is M) then (Fitness is L)
If (RT is VH) and (D is VH) and (En is VH) and (Av is L) then (Fitness is M)
If (RT is VH) and (D is VH) and (En is VH) and (Av is VL) then (Fitness is VL)

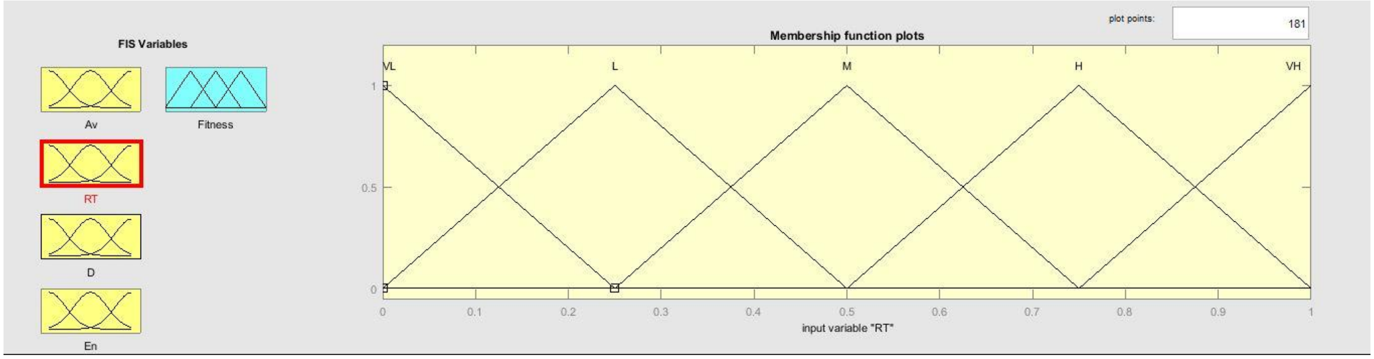


Fig. 1. Membership function for response time parameter.

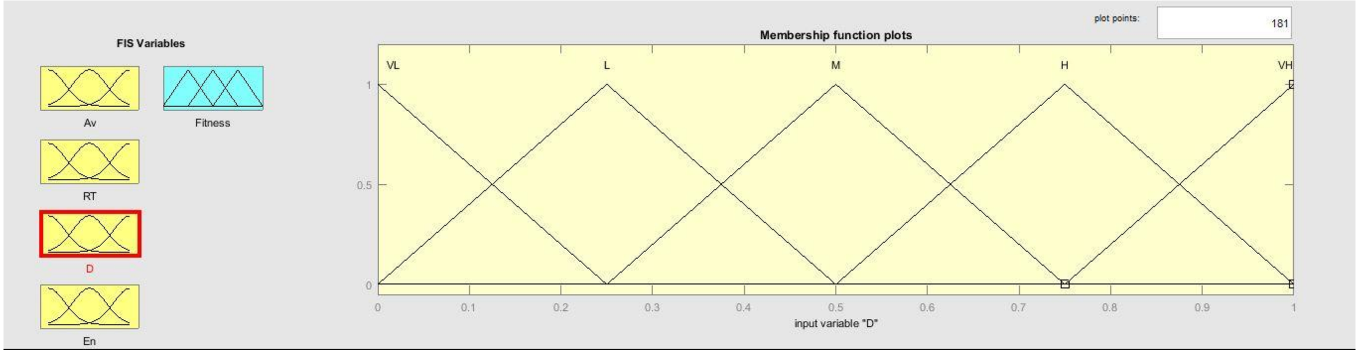


Fig. 2. Membership function for delay parameter.

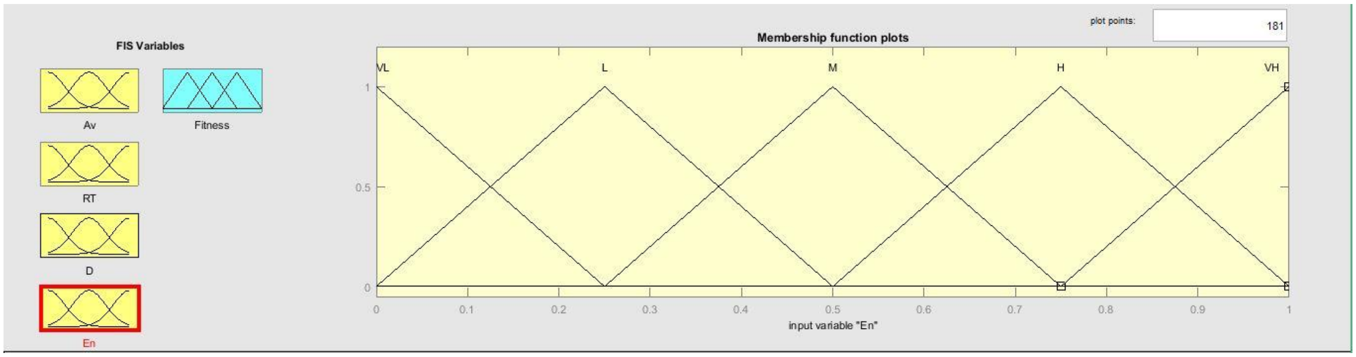


Fig. 3. Membership function for energy parameter.

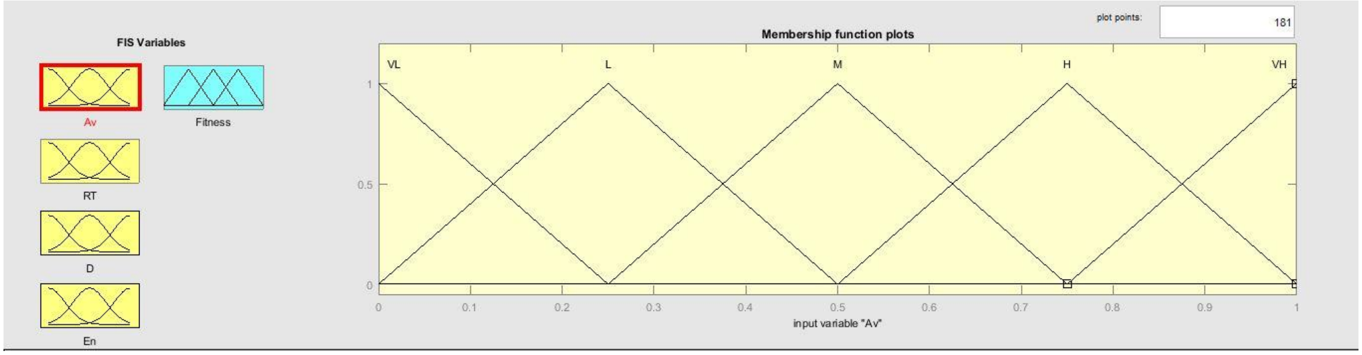


Fig. 4. Membership function for accessibility rate parameter.

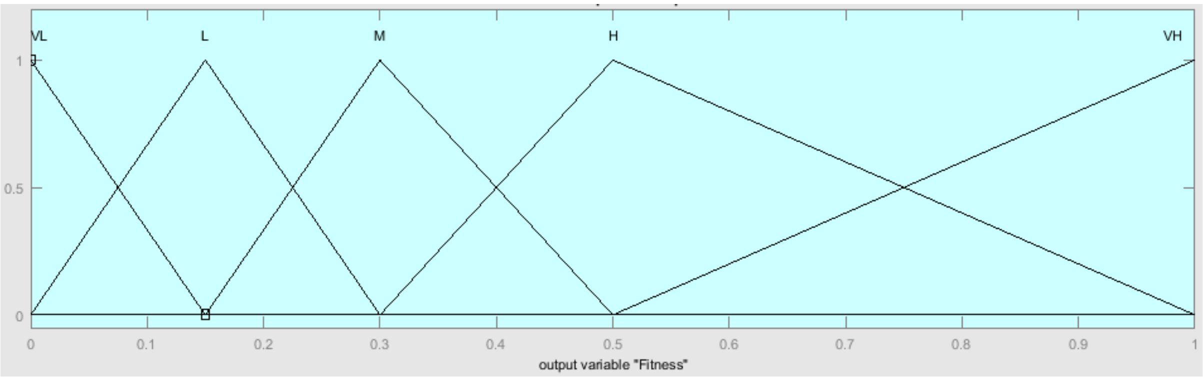


Fig. 5. Membership function for fuzzy system output.

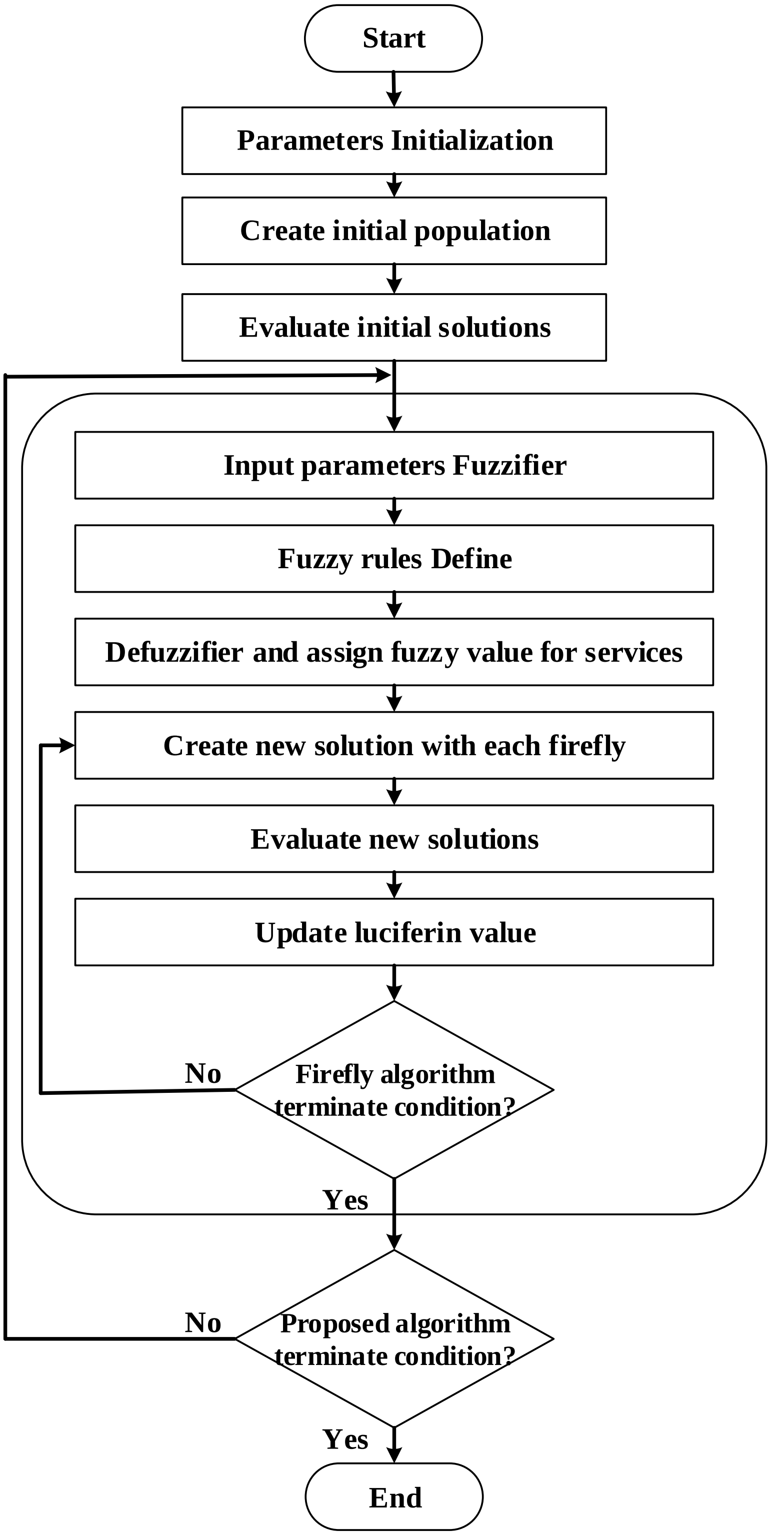

Start
Parameters Initialization
Create initial population
Evaluate initial solutions
Input parameters Fuzzifier
Fuzzy rules Define
Defuzzifier and assign fuzzy value for services
Create new solution with each firefly
Evaluate new solutions
Update luciferin value
Firefly algorithm terminate condition?
No
Yes
Proposed algorithm terminate condition?
No
Yes
End

Fig. 6. Proposed method flowchart.

IV. EVALUATION AND SIMULATION

Matlab is used for the simulation of the proposed method. All these experiments are performed on an HP computer with a Core i5 2.0 GHz CPU and 4 GB main memory. The utilized dataset in the proposed method is a dataset provided by Al-Masri et al., called Quality of Web Service (QWS), and provides a base for the services researchers. The QWS dataset includes a set of 2507 web services and measures their QoS. These data are presented in Table II. The parameters are considered objective parameters. Firefly algorithm parameters also are presented in Table III.

A. The Simulation Results' Study

The proposed method's results are compared with [36], [37], and [38] in this section and presented in different tables and figures. Moreover, two experiments are explained in the following to study the proposed method's convergence to optimal solutions and stability test.

In the first experiment, the resulting data of [36] are used to evaluate the proposed method in this thesis. The proposed method in [36] is based on a multi-objective genetic algorithm to investigate the service composition of IoT that is evaluated with the different number of tasks and candidate services for the tasks. Hence, our proposed method is evaluated in this experiment with ten tasks and 10, 30, and 50 candidate services. Fig. 7 shows the proposed method's execution time and comparison with the multi-objective genetic algorithm [36]. Based on the results, the proposed method's execution time is less than the genetic algorithm to find appropriate services for composition. Because the proposed method can find the most suitable network node for the services selection and composition that provides QoS requirements by tracing the most optimal IoT nodes. The routing method and selecting the most suitable resource (IoT node) with proper services for the user's requests will be explained in detail with mathematical equations in the third chapter.

In the second experiment, the resulting data of [37] are used to evaluate the proposed method in this thesis. The proposed method in [37] is based on the Markov chain and ant colony optimization algorithm for IoT services composition. The authors of [37] evaluate their method with the different number of requests. Hence, our proposed method (firefly-fuzzy logic) is simulated and evaluated with the different number of requests in this experiment.

Fig. 8 to 11 show the availability rate, response time, reliability rate, and consumed energy of the proposed method, respectively, which are compared with the method in [37]. Based on the results in Fig. 8, the proposed method (firefly-fuzzy logic) has a higher accessibility rate than the Markov-ant colony because of tracing the network nodes based on energy and distance. Moreover, selecting the most suitable route between the network's nodes (network's resources) is based on the movement pattern of the fuzzy firefly. According to Fig. 9, the response time of the proposed method is less than the method in [37]. Fig. 10 to 11 also show better performance of the proposed method than the method in [37] in terms of reliability.

In the third experiment, the resulting data of [39] are used to evaluate the proposed method in this thesis. The proposed method in [39] is based on the Concurrent Requests Integration Optimization (CRIO) mechanism and Gray Wolf (GWO) for IoT services composition. Fig. 12 shows the consumed energy by the proposed method. The results in Fig. 12 show that the proposed method has a lower energy consumption than the CRIO-GWO.

TABLE II. QWS PARAMETERS DEFINITION

Parameters	Definition
Response Time	The required time duration for a request transmission and its response reception.
Availability	The number of successful calls to the total number of calls ratio.
Energy	The amount of the consumed energy of the host nodes of the candidate services.
Successability	The number of responses to the number of request messages ratio.
Reliability	The number of faulty messages to the total number of messages ratio.
Latency	The duration of a request process by a server

TABLE III. SIMULATION VARIABLES

Parameters	Definition	Value	measurement Unit
N	Number of nodes	100-600	-
MaxIT	Maximum algorithm iteration	100-1000	-
L	Data packet length	1024	bit
$\boldsymbol{E_{initial}}$	Initial energy	0.5 – 10	joule
$\boldsymbol{\alpha_0}$	Firefly algorithm constant	[0 – 1]	-
$\boldsymbol{\beta_0}$	Firefly algorithm constant	1	-
γ	Firefly algorithm constant	0.1 – 10	-

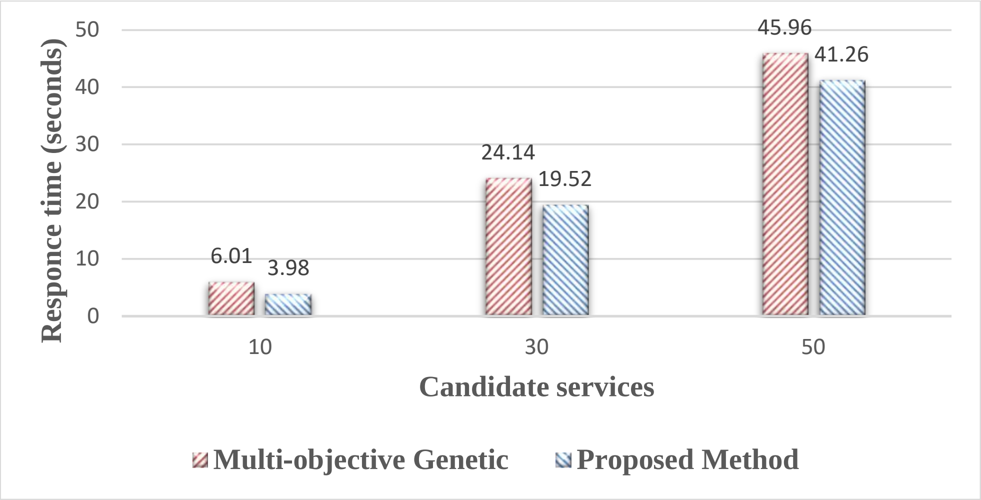


Fig. 7. Response time vs. the number of candidate services.

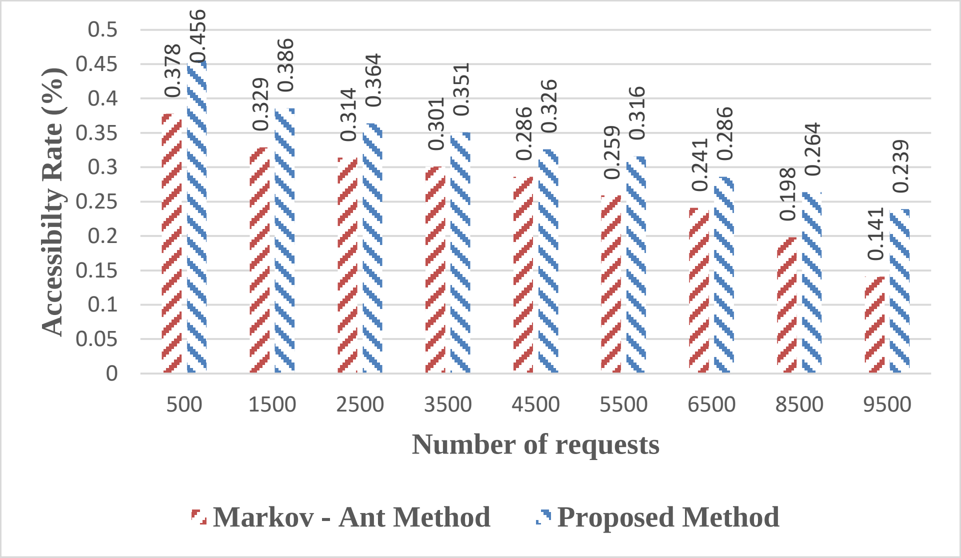


Fig. 8. Accessibility rate vs. the number of requests.

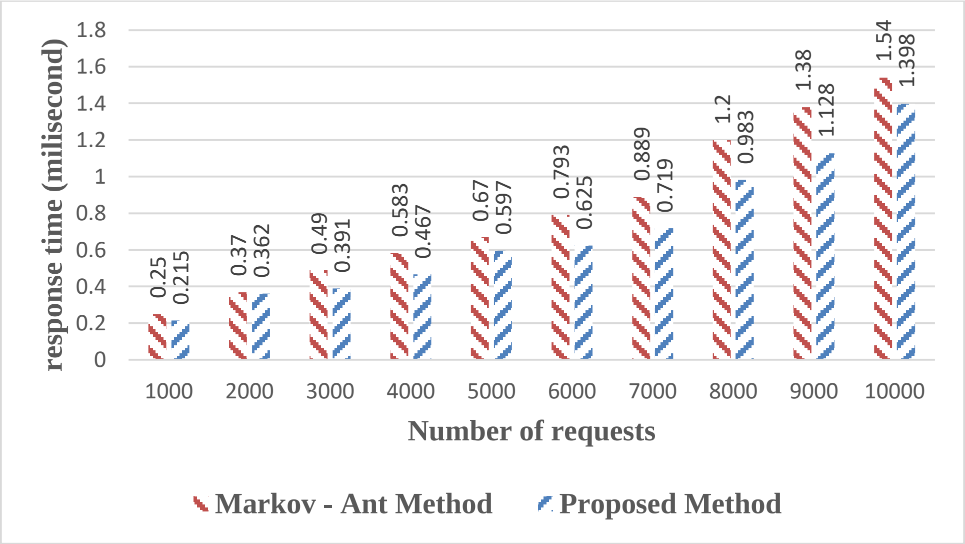


Fig. 9. Response time vs. the number of requests.

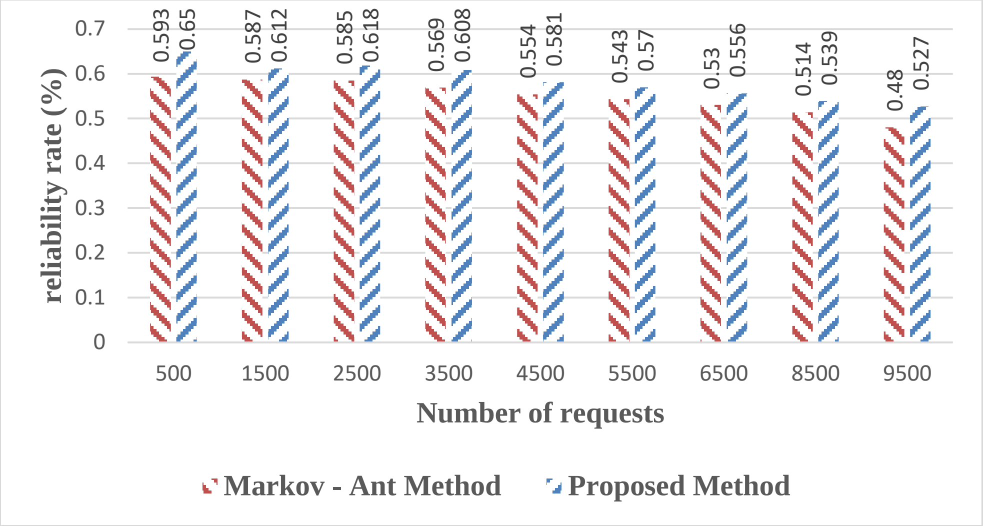


Fig. 10. Reliability rate vs. the number of requests.

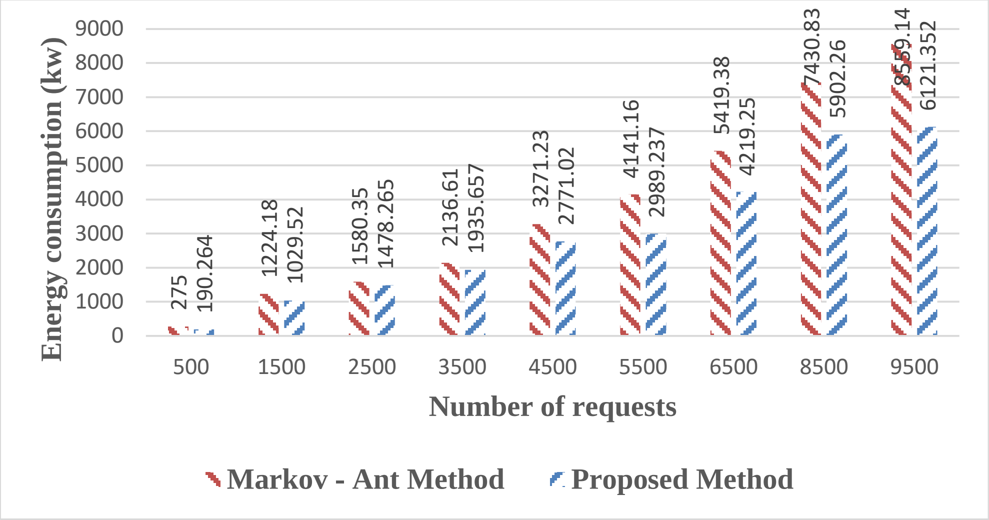


Fig. 11. Energy consumption vs. the number of requests.

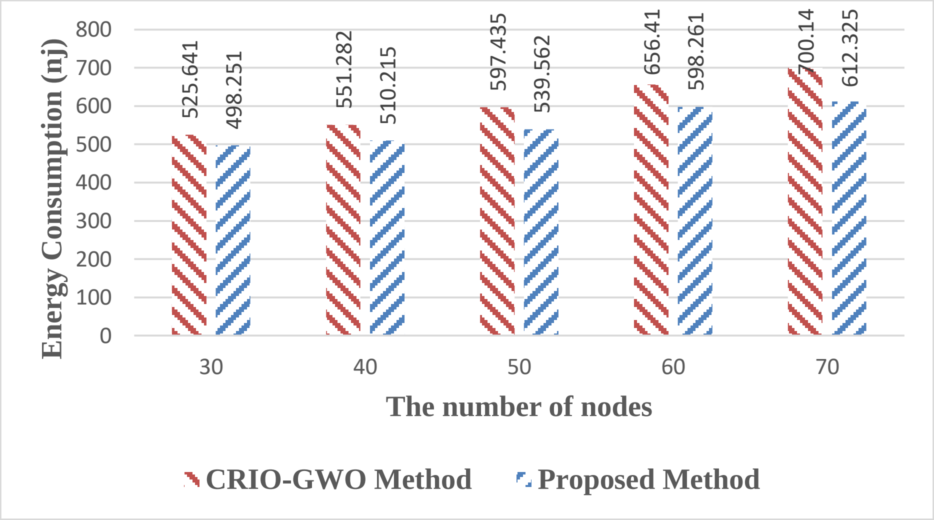


Fig. 12. Energy consumption vs. the number of nodes.

V. Conclusion

In general, cloud computing refers to the provision of dynamically scalable and virtualized resources over the Internet as services. Depending on the user's needs, a variety of cloud-based services may be delivered, which are often composited to meet those requirements. To combine and consolidate cloud services, service composition is emerging as a universal technology. By combining previously existing services, this idea aims to reduce costs and improve efficiency through a new cloud service. An IoT-enabled cloud service composition method using the fuzzy logic system and FOA is proposed in this paper. Energy, reliability, delay, and availability are the objective parameters in the proposed method. There are three main steps involved in this method in order to provide the proper nodes for the composition of cloud services. First, the requirements to start the algorithm are provided, like initialization of parameters and the dataset values normalization. Second, fuzzyification is performed on the parameters after determining the input parameters. Third, the firefly algorithm selects the most suitable nodes (resources) for service composition. Finally, in the fourth section of this paper, the proposed method is simulated using Matlab. A comparison of the proposed method with previous ones showed that the proposed method performed better in terms of response time, availability, and energy consumption. The dynamic nature of the cloud may be considered in our future work. A cloud's processing capabilities are limited, and the number of atomic services may change over time. Future research should focus on how to compose atomic services. In addition to addressing the energy consumption issue, we hope to maintain other metrics in the scheduling, including reliability, stability, trust degree, etc.

References

[1] Vahedifard, F., et al., Artificial intelligence for radiomics; diagnostic biomarkers for neuro-oncology. World Journal of Advanced Research and Reviews, 2022. 14(3): p. 304-310.

[2] Saeidi, S.A., et al. A novel neuromorphic processors realization of spiking deep reinforcement learning for portfolio management. in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022. IEEE.

[3] Akhavan, J. and S. Manoochehri. Sensory data fusion using machine learning methods for in-situ defect registration in additive manufacturing: a review. in 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). 2022. IEEE.

[4] Khosravi, F., et al. Implementation of an Elastic Reconfigurable Optical Add/Drop Multiplexer based on Subcarriers for Application in Optical Multichannel Networks. in 2022 International Conference on Electronics, Information, and Communication (ICEIC). 2022. IEEE.

[5] Khosravi, F., et al., Improving the performance of three level code division multiplexing using the optimization of signal level spacing. Optik, 2014. 125(18): p. 5037-5040.

[6] Haghshenas, S.H., M.A. Hasnat, and M. Naeini, A Temporal Graph Neural Network for Cyber Attack Detection and Localization in Smart Grids. arXiv preprint arXiv:2212.03390, 2022.

[7] Taami, T., S. Krug, and M. O'Nils. Experimental characterization of latency in distributed iot systems with cloud fog offloading. in 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS). 2019. IEEE.

[8] He, P., et al., Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review. Sustainable Computing: Informatics and Systems, 2022. 36: p. 100822.

[9] Meisami, S., M. Beheshti-Atashgah, and M.R. Aref, Using Blockchain to Achieve Decentralized Privacy In IoT Healthcare. arXiv preprint arXiv:2109.14812, 2021.

[10] Mehbodniya, A., et al., Modified Lamport Merkle Digital Signature blockchain framework for authentication of internet of things healthcare data. Expert Systems, 2022. 39(10): p. e12978.

[11] Pourghebleh, B., et al., A roadmap towards energy-efficient data fusion methods in the Internet of Things. Concurrency and Computation: Practice and Experience, 2022: p. e6959.

[12] Kumar, A., et al., Smart power consumption management and alert system using IoT on big data. Sustainable Energy Technologies and Assessments, 2022: p. 102555.

[13] Pourghebleh, B., et al., The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments. Cluster Computing, 2021: p. 1-24.

[14] Ataie, I., et al. D 2 FO: Distributed Dynamic Offloading Mechanism for Time-Sensitive Tasks in Fog-Cloud IoT-based Systems. in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC). 2022. IEEE.

[15] Bermejo, B., C. Juiz, and C. Guerrero, Virtualization and consolidation: a systematic review of the past 10 years of research on energy and performance. The Journal of Supercomputing, 2019. 75(2): p. 808-836.

[16] Sefati, S., M. Mousavinasab, and R. Zareh Farkhady, Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation. The Journal of Supercomputing, 2022. 78(1): p. 18-42.

[17] Najafizadeh, A., et al., Multi-objective Task Scheduling in cloud-fog computing using goal programming approach. Cluster Computing, 2022. 25(1): p. 141-165.

[18] Hayyolalam, V., et al., Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends. Concurrency and Computation: Practice and Experience, 2022. 34(5): p. e6698.

[19] Hayyolalam, V., et al., Exploring the state-of-the-art service composition approaches in cloud manufacturing systems to enhance upcoming techniques. The International Journal of Advanced Manufacturing Technology, 2019. 105(1-4): p. 471-498.

[20] Praveenchandar, J. and A. Tamilarasi, Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 2021. 12(3): p. 4147-4159.

[21] Dorsala, M.R., V. Sastry, and S. Chapram, Blockchain-based solutions for cloud computing: A survey. Journal of Network and Computer Applications, 2021. 196: p. 103246.

[22] Gabi, D., et al., Cloud customers service selection scheme based on improved conventional cat swarm optimization. Neural Computing and Applications, 2020: p. 1-22.

[23] Kritikos, K. and D. Plexousakis. Multi-cloud application design through cloud service composition. in 2015 IEEE 8th international conference on cloud computing. 2015. IEEE.

[24] Huo, Y., et al., Discrete gbest-guided artificial bee colony algorithm for cloud service composition. Applied Intelligence, 2015. 42(4): p. 661-678.

[25] Liu, Z.-Z., et al., Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition. Information Sciences, 2016. 326: p. 315-333.

[26] Huang, J., et al. QoS correlation-aware service composition for unified network-cloud service provisioning. in 2016 IEEE Global Communications Conference (GLOBECOM). 2016. IEEE.

[27] Karimi, M.B., A. Isazadeh, and A.M. Rahmani, QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm. The Journal of Supercomputing, 2017. 73(4): p. 1387-1415.

[28] Singh, A., D. Juneja, and M. Malhotra, A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. Journal of King Saud University-Computer and Information Sciences, 2017. 29(1): p. 19-28.

[29] Wang, S., et al., Towards green service composition approach in the cloud. IEEE Transactions on Services Computing, 2018. 14(4): p. 1238-1250.

[30] Jian, C., M. Li, and X. Kuang, Edge cloud computing service composition based on modified bird swarm optimization in the internet of things. Cluster Computing, 2019. 22(4): p. 8079-8087.

[31] Jin, H., et al., Eagle strategy using uniform mutation and modified whale optimization algorithm for QoS-aware cloud service composition. Applied Soft Computing, 2022. 114: p. 108053.

[32] Tao, F., et al., Correlation-aware resource service composition and optimal-selection in manufacturing grid. European Journal of Operational Research, 2010. 201(1): p. 129-143.

[33] Zhang, Y., et al., Long/short-term utility aware optimal selection of manufacturing service composition toward industrial internet platforms. IEEE Transactions on Industrial Informatics, 2019. 15(6): p. 3712-3722.

[34] Perumal, B. and A. Murugaiyan, A firefly colony and its fuzzy approach for server consolidation and virtual machine placement in cloud datacenters. Advances in Fuzzy Systems, 2016. 2016.

[35] Singh, U. and R. Salgotra, Synthesis of linear antenna arrays using enhanced firefly algorithm. Arabian Journal for Science and Engineering, 2019. 44(3): p. 1961-1976.

[36] Kashyap, N., A.C. Kumari, and R. Chhikara, Multi-objective Optimization using NSGA II for service composition in IoT. Procedia Computer Science, 2020. 167: p. 1928-1933.

[37] Sefati, S. and N.J. Navimipour, A qos-aware service composition mechanism in the internet of things using a hidden-markov-model-based optimization algorithm. IEEE Internet of Things Journal, 2021. 8(20): p. 15620-15627.

[38] Alsaryrah, O., I. Mashal, and T.-Y. Chung, Bi-objective optimization for energy aware Internet of Things service composition. IEEE Access, 2018. 6: p. 26809-26819.

[39] Sun, M., et al., Energy-efficient IoT service composition for concurrent timed applications. Future Generation Computer Systems, 2019. 100: p. 1017-1030.

Systematic Review of Deep Learning Techniques for Lung Cancer Detection

Mattakoyya Aharonu[1], R Lokesh Kumar[2]*
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

***Abstract*—Cancer is the leading cause of deaths across the globe and 10 million people died of cancer and particularly 2.21 million new cases registered besides 1.80 million deaths, according to WHO, in 2020. Malignant cancer is caused by multiplication and growth of lung cells. In this context, exploiting technological innovations for automatic detection of lung cancer early is to be given paramount importance. Towards this end significant progress has been made and deep learning model such as Convolutional Neural Network (CNN) is found superior in processing lung CT or MRI images for disease diagnosis. Lung cancer detection in the early stages of the disease helps in better treatment and cure of the disease. In this paper, we made a systematic review of deep learning methods for detection of lung cancer. It reviews peer reviewed journal papers and conferences from 2012 to 2021. Literature review throws light on synthesis of different existing methods covering machine learning (ML), deep learning and artificial intelligence (AI). It provides insights of different deep learning methods in terms of their pros and cons and arrives at possible research gaps. This paper gives knowledge to the reader on different aspects of lung cancer detection which can trigger further research possibilities to realize models that can be used in Clinical Decision Support Systems (CDSSs) required by healthcare units.**

Keywords*—Artificial intelligence; deep learning; lung cancer; lung cancer detection; machine learning

I. Introduction

Cancer is an abnormal growth of cells in human body. Lung cancer is leading cause of deaths across the world and there are different kinds of cancers according to WHO. They include breast cancer, lung cancer, colon and rectum cancer, skin cancer, prostate cancer, and stomach cancer. Cancer incidence in 2020 is high with 2.26 billion breast cancer cases, 2.21 million lung cancer cases, 1.93 million colon and rectum cancer, 1.41 million prostate cancer cases, 1.20 million skin cancer cases and 1.09 million stomach cancer cases. These statistics show the alarming nature of the disease across the globe. There are many risk factors for cancers. They include tobacco, alcohol, unhealthy diet, air pollution and physical activity. Some chronic diseases also play their role in causing cancer. The major challenge is to detect the lung cancer at early stage and increase the survival rate of lung cancer patients. In this context, technological innovations need to be understood towards cancer diagnosis automatically. With advancements in artificial intelligence (AI) based approaches such as machine learning and deep learning, it is important to consider them for automatic diagnosis of lung cancer. Since there are number of training samples available and they grow in future, supervised learning is found suitable for automatic lung cancer detection.

Different methods came into existence based on image processing, machine learning, deep learning, and artificial intelligence (AI). The existing methods are investigated in this paper through systematic review. Several ML based approaches such as [2], [3], [4], [5] and [9], to mention few, are some of the representative approaches where supervised learning is used for lung cancer detection. There are plenty of deep learning methods explored in [6], [7], [8], [10], [11] to mention few, that have different advanced learning models with optimizations for improving prediction performance. Our contributions in this paper are as follows. We have made systematic review of methods used for lung cancer detection. It is backed by a methodology and results of research questions. It throws light on various insights besides giving research gaps. The remainder of the paper is structured as follows. Section II reviews literature on existing lung cancer prediction methods. Section III presents research methodology used for systematic review. Section IV provides the results for different research questions. Section V concludes the paper and gives directions for future scope of the research.

II. Literature Review

This section reviews literature on deep learning methods used for lung cancer detection. It also covers machine learning methods, imaging techniques and performance metrics besides important research gaps.

A. Machine Learning Methods

Machine learning techniques with supervised learning are used for lung cancer detection. Bhatia et al. [2] proposed a methodology to diagnose lung cancer using CT images. Their methodology includes Random Forest (RF) and XGBoost methods along with ensemble learning approach. Radhika and Nair [4] explored different ML models such as Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM) and Naïve Bayes. Pradhan and Chawla [5] used ML models along with a novel architecture for realizing Medical Internet of Things (MIoT). Rehman et al. [9] used ML models like SVM and ANN to detect the disease from lung CT scans. Jenipher and Radhika [14] explored in feature extraction and selection with ML models for early prediction of lung cancer Banerjee et al. [16] studied Classification models like SVM, RF for prediction of early-stage lung cancer. Joshua et al. [17] investigated on different ML techniques suitable for image processing leading to lung cancer prognosis. Chaturvedi et al. [19] studied ML models for detection and classification of lung cancer using MRI, X-Ray, and CT-scan imagery. Raoof et al. [26] proposed ML based framework made up of SVM, ANN and so on for lung cancer research. Shanthi and Rajkumar [27]

*Corresponding Author

proposed a feature selection method based on Stochastic Diffusion Search (SDS) to improve ML performance in lung cancer classification.

Singh and Gupta [31] used X-Ray, CT and MRI imagery for lung cancer detection with the help of ML models. DICOM CT images are used by Dev et al. [34] with ML models to detect lung cancer. Saba [38] investigated on recent advancements on ML models for lung cancer research. They used different imaging techniques for proof of the concept. Thallam et al. [40] explored ML models like ANN, KNN, RF and SVM for early prediction of lung cancer. Both ML and IsomiR Expression methods are combined by Liao et al. [46] for cancer diagnosis. Pawar et al. [49] used ML models and image processing techniques to detect lung cancer. Chabon et al. [50] investigated on genomic features for early lung cancer detection in a non-invasive approach. Selvathi and AarthyPoornila [55] studied different ML models such as SVM and KNN for cancer research. Lalitha [60] used ML models for automatic detection of lung cancer. Gupta et al. [62] investigated on supervised ML techniques to detect cancers. Kumar and Rao [63] made research like that of [62].

Katiyar and Singh [66] compared ML models and their efficiency in cancer detection process. Wang et al. [67] focused on pathology analysis using lung cancer images using AI approaches. Houby [68] discussed about disease management techniques associated with ML. Gang et al. [69] proposed dimensionality reduction method along with deep learning to analyses chest X-Ray for detection of lung cancer. Mukherjee and Bohra [70] used ML approaches for disease prediction. Hussain et al. [71] presented different feature extraction methods to improve prediction capability of ML models in disease prediction. Bankar et al. [76] made symptom analysis with data-driven approach using ML techniques for early detection of lung cancer.

B. Deep Learning Methods

Deep learning methods are based on neural networks. They are used widely for processing images such as lung CT and MRI. Tekade and Rajeswari [6] used deep learning models such as VGG and U-net architecture for lung cancer classification. Shakeel et al. [7] proposed "Improved Profuse Clustering Technique (IPCT)" using CT images for lung cancer detection. Shakeel et al. [8] used improved deep neural network and ensemble learning for automatic detection of the disease. Das and Majumder [10] explored different methods linked to deep learning practices for lung cancer detection. Kalaivani et al. [12] used a CNN based model known as DenseNet with CT imagery for disease diagnosis. Shin et al. [13] used deep learning approach for Spectroscopic analysis towards lung cancer diagnosis. Ibrahim et al. [18] used CT scan images investigating on chest diseases such as Covid-19, lung cancer and pneumonia. Their methodology is based on CNN based pre-trained deep learning models such as ResNet152V2 and VGG 19. Cherukuri et al.,[21] the study looks upon clinical tomography exercises. Lakshmana Prabu et al. [22] proposed a deep learning-based approach based on techniques known as "Optimal Deep Neural Network (ODNN) and Linear Discriminate Analysis (LDA)." Selvathi and Poornima [23] proposed deep learning methods for medical data analysis using CT and MRI imagery with the notion of Region of Interest (ROI).

Elnakib and Amer [24] proposed a deep learning method based on VGG16 and AlexNet along with an optimization technique based on Genetic Algorithm (GA) to classify lung nodules from CT images. Wang et al. [25] presented a weakly supervised learning approach with deep learning for classifying lung images. They exploited Fully Convolutional Network (FCN) to realize an automated detection system. Liu et al. [28] explored deep reinforcement learning approaches towards lung cancer detection in presence of Medical IoT. Sajja et al. [29] investigated on deep transfer learning using CT images for diagnosis of lung cancer. Schwyzer et al. [30] used deep neural networks with CT images for lung cancer detection automatically. Ardila et al. [32] used 3D deep learning phenomena with end-to-end screening of CT images for lung cancer prognosis. Avanzo et al. [33] focused on combining deep learning with radiomics for efficient detection of lung cancer. Hashemzadeh et al. [35] investigated on automatic cancer screening applications using deep learning models. Kancherla and Mukkamala [36] proposed a novel methodology for lung cancer diagnosis. This methodology exploits features associated with nucleus segmentation. Xu et al. [37] explored serial medical imaging with deep learning for prediction of lung cancer.

Doppalapudi et al. [39] used deep learning approaches to predict period of lung cancer survival. Coudray et al. [41] proposed deep learning-based approach for classification and mutation prediction linked to lung cancer. Munir et al. [43] explored various methods in ML and deep learning for prognosis of different kinds of cancers. Nasrullah et al. [44] used deep learning techniques along with multiple strategies to detect lung nodules and classify them. Mhaske et al. [45] proposed a deep learning algorithm to analyse lung CT image to find the presence of cancer. Hua et al. [47] focused on lung nodule classification using deep learning approaches. Subramanian et al. [48] proposed a deep learning framework using CNN based methods such as AlexNet, LeNet and VGG16. Kumar and Bakariya [51] used deep learning methods to find the presence of malignant cancers in CT images. Kriegsmann et al. [52] used deep learning models to classify and differentiate non-small cell lung cancer from small cell lung cancer. Yang et al. [53] made a retrospective study of whole slide images with deep learning for multi-class classification of lung cancer. Similarly, Hosny et al. [54] made a cohort radiomics study with deep learning methods for lung cancer prognosis.

Jena et al. [56] proposed a hybrid model based on deep learning, named DGMM-RBCNN, for detection and classification of lung cancer. Pham et al. [57] proposed a two-step deep learning approach for lung cancer detection from histopathological images. Gordienko et al. [58] used chest X-Ray images along with deep learning for lung segmentation to diagnose lung cancer. Sun et al. [59] exploited ROI based approaches and automatic feature selection using deep learning for lung cancer diagnosis. Fang [61] made a hybrid approach using deep learning, transfer learning, GoogLeNet and the features linked to median intensity projections.

Ponnada and Srinivasu [64] proposed efficient CNN model based on deep learning for lung cancer prognosis. Cha et al. [65] used deep learning and chest radiographs to detect lung cancer. Hatuwal and Thapa [72] used Histopathological imagery with deep learning such as CNN for lung cancer prediction. Sungheetha et al. [73] made a comparative study of deep learning models while lung CT image segmentation is explored by Brahim et al. [74]. Salaken et al. [75] proposed a methodology to study low population dataset to extract features using deep learning for lung cancer diagnosis. Lustberg et al. [77] proposed automatic contouring technique with deep learning across the globe for clinical evaluation. Lung abnormality detection method is proposed in [78] for automatic lung cancer detection based on deep learning. Masood et al. [79] proposed a methodology for Pulmonary cancer detection with CT imagery. Bharati et al. [80] used X-Ray images to evaluate their hybrid deep learning method that combines VGG, CNN and data augmentation.

C. *Imaging Techniques*

Lung MRI and CT scan imagery are widely used for lung cancer diagnosis. Rehman et al. [9] used ML models like SVM and ANN to detect the disease from lung CT scans. Shakeel et al. [7] proposed “Improved Profuse Clustering Technique (IPCT)” using CT images for lung cancer detection. Kalaivani et al. [12] used a CNN based model known as DenseNet with CT imagery for disease diagnosis. CT imagery is used by Rahane et al. [15]. Joshua et al. [17] used both MRI and CT scan imagery in their research. Ibrahim et al. [18] used CT scan images investigating on chest diseases such as Covid-19, lung cancer and pneumonia. Chaturvedi et al. [19] studied ML models for detection and classification of lung cancer using MRI, X-Ray and CT-scan imagery. Riquelme et al. [20] used CT scans for lung cancer nodules classification. Lakshmana Prabu et al. [22] used lung CT images for detection of cancer. In [31] X-Ray, CT and MRI imagery are used for lung cancer detection. In [32] CT images and in [34] DICOM CT imagers are used in lung cancer research. Kadir and Gleeson [42] used advanced imaging techniques for lung cancer diagnosis.

D. *Performance Metrics*

Widely used performance metrics in the ML and deep learning-based approaches for lung cancer detection are summarized in the Table I.

TABLE I. WIDELY USED METRICS IN ML AND DEEP LEARNING RESEARCH LINKED TO LUNG CANCER DIAGNOSIS

Metric	Formula	Value range	Best Value
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	[0; 1]	1
Precision (p)	$\frac{TP}{TP+FP}$	[0; 1]	1
Recall (r)	$\frac{TP}{TP+FN}$	[0; 1]	1
F1-Score	$2*\frac{(p*r)}{(p+r)}$	[0; 1]	1

Precision refers to positive predictive value while the recall refers to true positive rate. F1-score is the harmonic mean of both precision and recall which is used to have a measure without showing imbalance while accuracy measure may show imbalance. These metrics are derived from confusion matrix shown in Fig. 1.

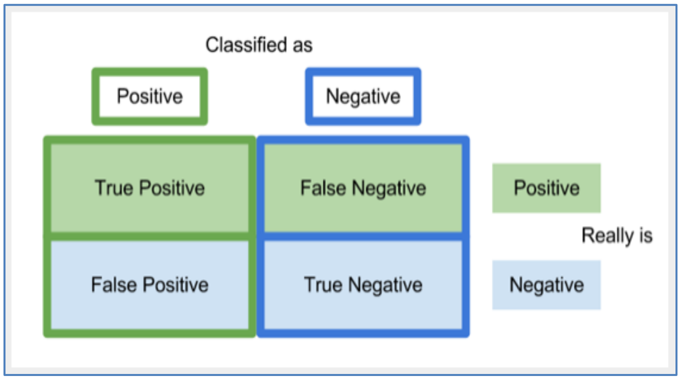


Fig. 1. Confusion matrix.

Based on the confusion matrix, the confusion matrix shows the measures like true positive (TP), false positive (FP), false negative (FN) and true negative (TN). These are determined by comparing result of ML algorithm when compared with the ground truth.

E. *Research Gaps*

With recent advances in deep learning, research has made a significant leap to help identify, classify, and quantify patterns in medical images. Particularly, improvements in computer vision inspired its use in medical image analysis such as image segmentation, image registration, image fusion, image annotation, computer-aided diagnosis and prognosis, lesion/landmark detection, and microscopic imaging analysis, to name a few. Particularly lung cancer is one of the problems that has attracted significant research. Many deep learning-based solutions came into existence. However, there is lot to do to have more accurate detection of lung cancer early. For instance, the research carried out in [22] has significant limitations. First, it has issues in detection of lung cancer early. Second, there is need for improving CNN architectures and cascade them besides making a pipeline with patient-level descriptive statistics for better prediction. Third, ensemble of classifiers could lead to further improvement in prediction of lung cancer.

III. RESEARCH METHODOLOGY

This section presents the research methodology for making systematic review of literature on deep learning techniques for lung cancer detection. It throws light on many aspects of the review process with details on preferred databases for articles, the process of research or conceptual framework, criteria for including or excluding research articles and distribution of articles used in the research.

A. Research Questions

This systematic review has synthesis of literature from 2012 to 2021 on deep learning methods used for lung cancer detection. It has potential to answer the following research questions.

Research Question 1: What are different deep learning methods used from 2012 to 2021 to diagnose lung cancer?

Research Question 2: What are different datasets used from 2012 to 2021 for the research of lung cancer detection?

Research Question 3: What are the different development platforms used for lung cancer detection from 2012 to 2021?

Research Question 4: What are the different performance metrics used for evaluation of lung cancer detection methods?

Research Question 5: What are the imaging techniques used for lung cancer detection from 2012 to 2021?

Research Question 6: What are the results obtained for lung cancer detection from 2012 to 2021?

These research questions help to ascertain answers to various questions in terms of deep learning methods, datasets used for the research, development platforms used for building models, performance metrics widely used, imaging techniques and results.

B. Data Sources

Research articles used in this paper are collected from different sources. Article selection process includes criteria for inclusion and exclusion of articles. The databases or digital libraries from which articles are taken are as follows.

- Wiley (https://onlinelibrary.wiley.com/)
- Google Scholar (https://scholar.google.com)
- Springer (https://www.springer.com/in)
- Elsevier (https://www.elsevier.com/en-in)
- IEEEXplore (https://ieeexplore.ieee.org/Xplore/home.jsp)

C. Search Process

Search for articles is carried out with different search phrases. The search applications associated with digital libraries are used to fine relevant peer-reviewed articles that satisfy inclusion and exclusion criteria. The collected articles include articles and conferences.

Table II shows search process in terms of different phrases used for finding suitable articles on lung cancer detection. It has provision to collect articles from different sources based on the search phrases provided. After collecting research articles, different criteria on inclusion and exclusion are used to filter out the articles further.

TABLE II. SEARCH PHRASES USED IN THE SEARCH PROCESS

Sr. No.	Search Phrase	Description
1	Machine learning techniques for lung cancer detection	This is used for finding lung cancer research based on machine learning approaches.
2	Deep learning techniques for lung cancer detection	This is used for finding lung cancer research based on deep learning approaches.
3	Artificial intelligence techniques for lung cancer detection	This is used for finding lung cancer research based on Artificial intelligence approaches.
4	Lung cancer detection techniques	It is used to find any methods used for lung cancer detection

D. Criteria for Article Selection Process

Inclusion and exclusion criteria for this research are defined as presented in Table III. The criteria are meant for leveraging quality of the articles used for the systematic review. Articles in English language that have been published between 2017 and 2021 are used for systematic review.

Table III shows the criteria used to select quality articles used in the systematic review. It reflects the state-of-the-art methods used for lung cancer detection.

TABLE III. CRITERIA USED FOR ARTICLE SELECTION PROCESS

SL. NO.	Inclusion Criteria	Exclusion Criteria	Justification
1	Article in English language	Articles other than English	English articles are preferred as there is neither time for translation nor availability of required tools for translation.
2	Lung cancer with either CT or MRI imaging	Datasets other than CT and MRI	Lung cancer detection with these imaging techniques is reliable.
3	Peer-reviewed	Non-peer reviewed	Quality of articles that are peer-reviewed is considered to be high.
4	Machine learning and deep learning	Non machine learning and non-deep learning methods	The machine learning and deep learning techniques are preferred due to their utility.

E. Research Process

We followed a specific research process of search, exclusion, and inclusion criteria of articles that is illustrated in Fig. 2. As many as 180 articles are collected using the search phrases.

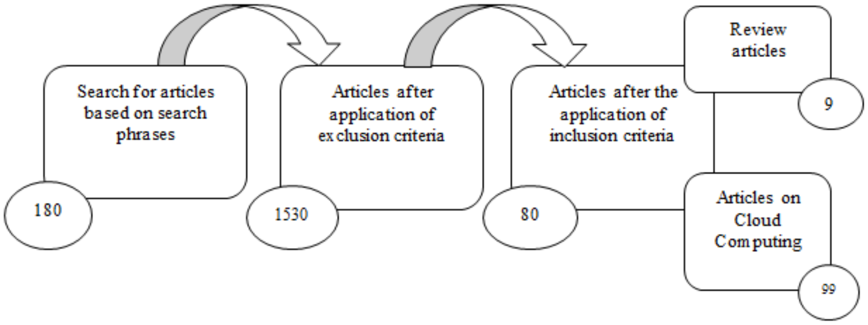

Search for articles based on search phrases
180
Articles after application of exclusion criteria
1530
Articles after the application of inclusion criteria
80
Review articles
9
Articles on Cloud Computing
99

Fig. 2. Methodology for article selection process.

Afterwards, the articles are subjected to exclusion criteria which have resulted in filtering of the articles to 153. Then the selection process continued with the inclusion criteria on the 153 articles. This process has eliminated 73 articles from the list and finally we got 80 articles for the systematic literature.

This article selection process has led to the removal of irrelevant articles that do not meet the selection criteria.

As presented in Fig. 3, the articles that satisfied inclusion and exclusion criteria are subjected to data analysis process. It is found that most of the articles belong to 2017 to 2021 publication years. From 2012 to 2016 only five articles are found to satisfy the selection quality process.

As presented in Fig. 4, the article distribution is provided in terms of percentage for each year. The highest number of articles belong to 2019 and 2020 years with 26% while 25% of articles are selected from 2018.

As presented in Fig. 5, IEEE is the publisher from which 11 journal papers and 14 conference papers are selected. From Elsevier publications 12 articles are selected while from Springer publications 10 articles are selected. The articles taken from Google Scholar are 21 while 5 are from MDPI.

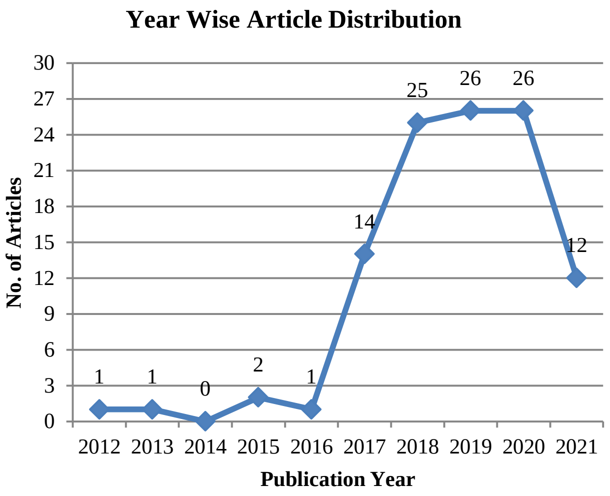


Fig. 3. Shows year wise distribution of published articles used for the review.

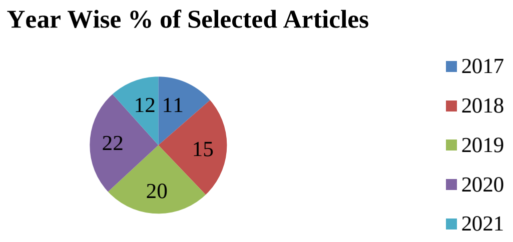


Fig. 4. Year wise % of selected articles.

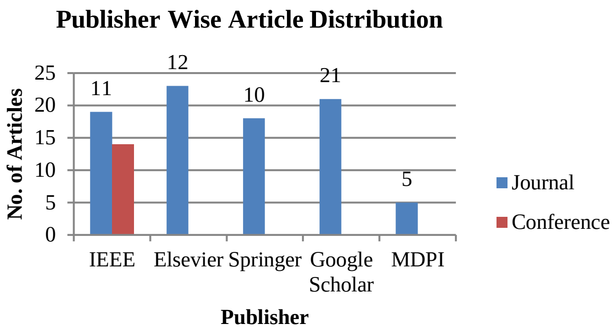


Fig. 5. Shows publisher wise selected article distribution.

As presented in Table IV, the articles used for the survey in this paper are provided and they belong to different journals and conferences from diversified and reputed publishers.

TABLE IV. SUMMARY OF RESEARCH ARTICLES USED IN REVIEW

Journal	References
IEEE	4, 6,9,10, 14, 15, 16, 21, 25, 26, 36.
Elsevier	3, 18, 28, 38, 39, 57, 68, 74, 77, 78, 79, 80.
Springer	1,2,8, 23, 27, 31, 33, 34, 56, 58.
ACS Publications	11, 13
MDPI	20, 43, 44, 52, 67
Google Scholar	29, 30, 32, 35, 37, 41, 42, 46, 47, 48, 49, 50, 51, 53, 54, 59, 60, 64, 65, 72, 73
IEEE Conference	40, 45, 55, 61, 62, 63, 66, 69, 70, 71, 75, 76, 85.
Other Conferences	5, 7, 12, 17, 19, 22, 24

IV. RESULTS OF THE RESEARCH

This section provides results containing answers for many research questions provided in Section III A.

A. Results of Research Question 1

Research Question 1: What are different deep learning methods used from 2012 to 2021 to diagnose lung cancer?

Most authors used Convolution Neural Network to detect the lung cancer from 2012 to 2021.

As presented in Table V, different deep learning methods used from 2012 to 2021 to diagnose lung cancer are provided.

TABLE V. RESULTS OF RESEARCH QUESTION 1

Reference	**Authors**	**Publisher**	**Year**	**Deep Learning Methods**
[1]	Asuntha and Srinivasan	Springer	2020	Convolutional Neural Network, FPSOCNN
[2]	Bhatia *et al.*,	Springer	2019	UNet and ResNet
[5]	Kanchan and Priyanka	Management Analytics	2020	Convolutional Neural Network and RNN and DBN
[6]	Ruchita and Rajeswari	IEEE	2018	Convolutional Neural Network
[7]	Shakeel *et al.*,	Other Conference	2019	Convolutional Neural Network and DNN
[8]	Mohamed *et al.*,	Springer	2020	Improved deep neural network (IDNN)
[10]	Susmita and Swanirbhar	IEEE	2020	Convolutional Neural Network
[12]	Kalaivani	Conference	2020	Convolutional Neural Network
[18]	Ibrahim *et al.*,	Elsevier	2021	Convolutional Neural Network
[24]	Ahmed and Hanan	Other	2020	Convolutional Neural Network
[25]	Wang *et al.*,	IEEE	2019	Convolutional Neural Network and FCN
[29]	Sajja *et al.*,	IIETA	2019	Convolutional Neural Network
[33]	Avanzo *et al.*,	Springer	2020	Convolutional Neural Network
[37]	Xu *et al.*,	Other	2019	Convolutional Neural Network and RNN
[39]	Doppalapudi *et al.*,	Elsevier	2021	Convolutional Neural Network, RNN and ANN
[42]	Timor and Fergus	Other	2018	Convolutional Neural Network
[43]	Munir *et al.*,	MDPI	2019	GANs, DANs, RNNs, CNNs, LTSM
[44]	Nasrullah *et al.*,	MDPI	2019	Convolutional Neural Network and R-CNN
[45]	Mhaske *et al.*,	IEEE	2019	Computer-Aided Diagnosis (CAD)
[46]	Hua *et al.*,	Other	2015	Computer-Aided Diagnosis (CAD)
[48]	Subramanian *et al.*,	Elsevier	2020	Convolutional Neural Network
[51]	Vinod and Brijesh	Medical Engineering & Technology	2021	Convolutional Neural Network
[52]	Kriegsmann *et al.*,	MDPI	2020	Convolutional Neural Network
[55]	Selvathi and AarthyPoornila	IEEE	2017	KNN, SVM, decision trees, and random forest
[56]	Jena *et al.*,	Springer	2021	DGMM-RBCNN
[58]	Gordienko *et al.*,	Springer	2019	Convolutional Neural Network

B. Results of Research Question 2

Research Question 2: What are different datasets used from 2012 to 2021 for the research of lung cancer detection?

As presented in Table VI, different datasets used from 2012 to 2021 for the research of lung cancer detection are provided. The data sets from (Lung Image Database Consortium) LIDC, LUNA dataset real time data, and Cancer Image Archive (CIA) etc. all the authors downloaded and used the datasets for their proposed different deep learning algorithms to analysis, classify and predict the lung cancer disease.

C. Results of Research Question 3

Research Question 3: What are the different development platforms used for lung cancer detection from 2012 to 2021. Most of the authors used Python and MATLAB for detect lung cancer.

As presented in Table VII, different development platforms used for lung cancer detection from 2012 to 2021 are provided. The authors used different computer programming languages and tools like Python, Java and MAT Lab for implementing their work.

TABLE VI. RESULTS OF RESEARCH QUESTION 2

Reference	Authors	Publisher	Year	Dataset Used
[1]	Asuntha and Srinivasan	Springer	2020	Real-time data set, B LIDC data set
[2]	Bhatia *et al*.,	Springer	2019	LIDC-IDRI dataset.
[4]	Radhika *et al*.,	IEEE	2019	Data world, UCI Machine Learning Repository
[5]	Kanchan and Priyanka	Management Analytics	2020	UCI repository dataset
[6]	Ruchita and Rajeswari	IEEE	2018	LUNA16 and Data Science Bowl 2017
[7]	Shakeel *et al*.,	Other Conference	2019	Cancer imaging Archive (CIA)
[8]	Mohamed *et al*.,	Springer	2020	cancer imaging archive (CIA)
[9]	Rehman *et al*.,	2021	2021	Chest CT scan image
[10]	Susmita and Swanirbhar	IEEE	2020	LUNA dataset
[25]	Wang *et al*.,	IEEE	2019	TCGA Dataset and SUCC Dataset
[29]	Sajja *et al*.,	IIETA	2019	Lung Image Database Consortium (LIDC) dataset
[44]	Nasrullah *et al*.,	MDPI	2019	LUNA16 and LIDC-IDRI
[45]	Mhaske *et al*.,	IEEE	2019	Lung Image Database Consortium (LIDC)dataset
[46]	Hua *et al*.,	Other	2015	Lung Image Database Consortium
[51]	Vinod and Brijesh	Medical Engineering & Technology	2021	LIDC
[58]	Gordienko *et al*.,	Springer	2019	JSRT, BSE-JSRT

TABLE VII. RESULTS OF RESEARCH QUESTION 3

Reference	Authors	Publisher	Year	Platform/Tool
[5]	Kanchan and Priyanka	Management Analytics	2020	Python
[6]	Ruchita and Rajeswari	IEEE	2018	Python
[7]	Shakeel *et al*.,	Other Conference	2019	MATLAB
[8]	Mohamed *et al*.,	Springer	2020	MATLAB
[9]	Rehman *et al*.,	2021	2021	MATLAB
[15]	Rahane *et al*.,	IEEE	2018	JAVA
[56]	Jena *et al*.,	Springer	2021	MATLAB

D. Results of Research Question 4

Research Question 4: What are different performance metrics used for evaluation of lung cancer detection method?

As presented in Table VIII, different performance metrics used for evaluation of lung cancer detection methods are provided. Performance metrics like Accuracy, sensitivity, specificity, f measure and ROC curves using the following metrics, the authors evaluate their developed models.

E. Results of Research Question 5

Research Question 5: What are the imaging techniques used for lung cancer detection from 2012 to 2021?

As presented in Table IX, different imaging techniques used for lung cancer detection from 2012 to 2021 are provided. The imaging techniques like Computer Tomography, X-Ray, and magnetic resonance imaging (MRI) provide different images related to Lung cancer and non-cancer images. The authors used those images as datasets for deep learning models.

F. Results of Research Question 6

Research Question 6: What are the results obtained for lung cancer detection from 2012 to 2021?

As presented in Table X, different performance metric like Accuracy, Specificity, F1 Score and area under the curve (AUC) values obtained by author's for lung cancer detection from 2012 to 2021 are provided.

TABLE VIII. RESULTS OF RESEARCH QUESTION 4

Reference	**Authors**	**Publisher**	**Year**	**Performance Metrics**
[1]	Asuntha and Srinivasan	Springer	2020	Accuracy,
[2]	Bhatia et al.,	Springer	2019	Accuracy
[4]	Radhika et al.,	IEEE	2019	Accuracy
[5]	Kanchan and Priyanka	Management Analytics	2020	accuracy, sensitivity, specificity, precision, F1-score, AUC, recall, and ROC
[6]	Ruchita and Rajeswari	IEEE	2018	Accuracy
[7]	Shakeel et al.,	Other Conference	2019	Accuracy, specificity, precision, F1-score, recall
[8]	Mohamed et al.,	Springer	2020	Accuracy, specificity, precision, recall and F-score.
[9]	Rehman et al.,	2021	2021	specificity, sensitivity, and accuracy
[10]	Susmita and Swanirbhar	IEEE	2020	specificity, sensitivity, and accuracy.
[12]	Kalaivani	Conference	2020	Accuracy
[14]	Nisha and Radhika	IEEE	2020	Accuracy
[17]	Joshua1 et al.,	Others	2020	ROC, AUC, Accuracy, Sensitivity and Specificity
[18]	Ibrahim et al.,	Elsevier	2021	Loss, AUC, precision, recall, and accuracy
[20]	Diego and Moulay	MDPI	2020	sensitivity (SE), specificity (SP), accuracy (ACC), precision (PPV), F1-score
[29]	Sajja et al.,	IIETA	2019	Sensitivity, Accuracy
[44]	Nasrullah et al.,	MDPI	2019	Accuracy
[45]	Mhaske et al.,	IEEE	2019	Accuracy
[48]	Subramanian et al.,	Elsevier	2020	Accuracy
[55]	Selvathi and AarthyPoornila	IEEE	2017	Accuracy
[56]	Jena et al.,	Springer	2021	Accuracy, sensitivity, specificity, F-measure MCC, and ROC curves
[60]	Lalitha. S	Springer	2021	Accuracy, Sensitivity, Specificity and Time consumption

TABLE IX. RESULTS OF RSEARCH QUESTION 5

Reference	Authors	Publisher	Year	Imaging Technique
[1]	Asuntha and Srinivasan	Springer	2020	Computed Tomography-Scan
[2]	Bhatia et al.,	Springer	2019	Computed Tomography-Scan
[6]	Ruchita and Rajeswari	IEEE	2018	Computed Tomography-Scan
[7]	Shakeel et al.,	Other Conference	2019	Computed Tomography-Scan
[8]	Mohamed et al.,	Springer	2020	Computed Tomography-Scan
[9]	Rehman et al.,	2021	2021	Computed Tomography-Scan
[10]	Susmita and Swanirbhar	IEEE	2020	Computed Tomography- image
[12]	Kalaivani	Conference	2020	Computed Tomography-Scan
[15]	Rahane et al.,	IEEE	2018	Computed Tomography-Scan
[17]	Joshua1 et al.,	Others	2020	CT Scan
[18]	Ibrahim et al.,	Elsevier	2021	Computed Tomography-Scan
[20]	Diego and Moulay	MDPI	2020	Computed Tomography-Scan
[29]	Sajja et al.,	IIETA	2019	Computed Tomography-Scan
[33]	Avanzo et al.,	Springer	2020	Computed Tomography-Scan
[37]	Xu et al.,	Other	2019	Computed Tomography-Scan
[44]	Nasrullah et al.,	MDPI	2019	Computed Tomography-Scan
[45]	Mhaske et al.,	IEEE	2019	Computed Tomography-Scan
[46]	Hua et al.,	Other	2015	Computed Tomography-Scan
[48]	Subramanian et al.,	Elsevier	2020	Computed Tomography-Scan
[51]	Vinod and Brijesh	Medical Engineering & Technology	2021	Computed Tomography-Scan
[52]	Kriegsmann et al.,	MDPI	2020	Computed Tomography-Scan
[56]	Jena et al.,	Springer	2021	Computed Tomography-Scan
[58]	Gordienko et al.,	Springer	2019	X-Ray, MRI
[60]	Lalitha. S	Springer	2021	Computed Tomography- Images

TABLE X. RESULTS OF RSEARCH QUESTION 6

Reference	Authors	Publisher	Year	Result (Accuracy)
[1]	Asuntha and Srinivasan	Springer	2020	Accuarcy-87.5%
[7]	Shakeel et al.,	Other Conference	2019	Accuracy-98.42%
[9]	Rehman et al.,	2021	2021	Accuracy - 93.00%
[12]	Kalaivani	Conference	2020	Accuracy - 90.85%
[18]	Ibrahim et al.,	Elsevier	2021	Accuracy-98.05% Specificity (SPC)-99.5%, F1 score - 98.24%, area under the curve (AUC)- 99.66%
[56]	Jena et al.,	Springer	2021	Accuracy-87.79%
[60]	Lalitha. S	Springer	2021	Accuracy -98.7%

V. CONCLUSION AND FUTURE WORK

In this paper, we made a systematic review of deep learning methods for detection of lung cancer. It reviews peer reviewed journal papers and conferences from 2012 to 2021. Literature review throws light on synthesis of different existing methods covering machine learning (ML), deep learning and artificial intelligence (AI). It provides insights of different deep learning methods in terms of their pros and cons and arrives at possible research gaps. This paper gives knowledge to the reader on different aspects of lung cancer detection which can trigger further research possibilities to realize models that can be used in Clinical Decision Support Systems (CDSSs) required by healthcare units. Besides, this systematic review could answer different research questions that may be of help to other researchers and readers. In future, we will work on filling certain research gaps identified in this paper.

REFERENCES

[1] Kancherla, K., & Mukkamala, S. Feature Selection for Lung Cancer Detection Using SVM Based Recursive Feature Elimination Method”. Lecture Notes in Computer Science, p168–176,2021.

[2] Bansal, Jagdish Chand; Das, Kedar Nath; Nagar, Atulya; et.al., [Advances in Intelligent Systems and Computing] Soft Computing for Problem Solving Volume 817 (SocProS 2017, Volume 2) || Lung

Cancer Detection: A Deep Learning Approach.10.1007/978-981-13-1595-4(Chapter 55), p699–705,2019.

[3] Ewelina Bębas;Marta Borowska;Marcin Derlatka;Edward Oczeretko;Marcin Hładuński;Piotr Szumowski;Małgorzata Mojsak; (2021). Machine-learning-based classification of the histological subtype of non-small-cell lung cancer using MRI texture analysis. Biomedical Signal Processing and Control, p1-8.

[4] PR, Radhika; Nair, Rakhi. A. S.; G, Veena. [IEEE 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) - Coimbatore, India (2019.2.20-2019.2.22)] 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) - A Comparative Study of Lung Cancer Detection using Machine Learning Algorithms., p1–4,2019.

[5] Pradhan, Kanchan; Chawla, and Priyanka. Medical Internet of things using machine learning algorithms for lung cancer detection. Journal of Management Analytics, p1–33,2020.

[6] Tekade, Ruchita; Rajeswari, K. [IEEE 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Pune, India (2018.8.16-2018.8.18)] 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Lung Cancer Detection and Classification Using Deep Learning., p1–5,2018.

[7] Mohamed Shakeel, P.; Burhanuddin, M.A.; Desa, Mohamad Ishak. "Lung Cancer Detection from CT Image Using Improved Profuse Clustering and Deep Learning Instantaneously Trained Neural Networks". Measurement, p1-26,2019.

[8] Shakeel, P. Mohamed; Burhanuddin, M. A.; Desa, Mohammad Ishak. "Automatic lung cancer detection from CT image using improved deep neural network and ensemble classifier". Neural Computing and Applications, p1-14,2020.

[9] Amjad Rehman; Muhammad Kashif; Ibrahim Abunadi; Noor Ayesha; . Lung Cancer Detection and Classification from Chest CT Scans Using Machine Learning Techniques. 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), p1-4,2021.

[10] Susmita Das; Swanirbhar Majumder; Lung Cancer Detection Using Deep Learning Network: A Comparative Analysis. 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), p1-6,2020.

[11] Zhiqiang Guo; Lina Xu;Yujuan Si;Navid Razmjooy;. Novel computer-aided lung cancer detection based on convolutional neural network - based and feature-based classifiers using metaheuristics. International Journal of Imaging Systems and Technology, p1-16,2021.

[12] N Kalaivani;N Manimaran;Dr. S Sophia;D D Devi. Deep Learning Based Lung Cancer Detection and Classification. IOP Conference Series: Materials Science and Engineering, p1-6,2021.

[13] Shin, Hyunku; Oh, Seunghyun; Hong, Soonwoo; Kang, Minsung.et.al., "Early-Stage Lung Cancer Diagnosis by Deep Learning-Based Spectroscopic Analysis of Circulating Exosomes". ACS Nano, p1-40,2020.

[14] V. Nisha Jenipher;S. Radhika. "A Study on Early Prediction of Lung Cancer Using Machine Learning Techniques. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), p1-6,2020.

[15] Rahane, Wasudeo; Dalvi, Himali; Magar, Yamini;et.al.,. [IEEE 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) - Coimbatore, India (2018.3.1-2018.3.3)] 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT) - Lung Cancer Detection Using Image Processing and Machine Learning HealthCare. , p1–5,2018.

[16] Banerjee, Nikita; Das, Subhalaxmi. [IEEE 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA) - Gunupur, India (2020.3.13-2020.3.14)] 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA) - Prediction Lung Cancerâ□ " In Machine Learning Perspective, p1–5,2020.

[17] Eali Stephen Neal Joshua1*, Midhun Chakkravarthy1 , Debnath Bhattacharyya. "An Extensive Review on Lung Cancer Detection Using Machine Learning Techniques: A Systematic Study. Revue d'Intelligence Artificielle. Vol. 34 (3), p351-359,2020.

[18] Dina M. Ibrahim;Nada M. Elshennawy;Amany M. Sarhan; Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases . Computers in Biology and Medicine, p1-13,2021.

[19] Pragya Chaturvedi;Anuj Jhamb;Meet Vanani;Varsha Nemade; . Prediction and Classification of Lung Cancer Using Machine Learning Techniques. IOP Conference Series: Materials Science and Engineering, p1-20,2021.

[20] Riquelme, Diego; Akhloufi, Moulay A. Deep Learning for Lung Cancer Nodules Detection and Classification in CT scans. AI, vol.1(1), p28–67, 2020.

[21] Naresh Cherukuri;Naga Raju Bethapudi;Venkata Sai Krishna Thotakura;Prasad Chitturi;CMAK Zeelan Basha;Raja Mani Mummidi;. Deep Learning for Lung Cancer Prediction using NSCLS patients CT Information. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), P1-6,2021.

[22] S.K., Lakshmanaprabu; Mohanty, Sachi Nandan; K., Shankar; N., Arunkumar; Ramirez, Gustavo. Optimal deep learning model for classification of lung cancer on CT images. Future Generation Computer Systems, p1-31,2018.

[23] Hemanth, Jude; Balas, Valentina Emilia. [Lecture Notes in Computational Vision and Biomechanics] Biologically Rationalized Computing Techniques for Image Processing Applications Volume 25 || Deep Learning Techniques for Breast Cancer Detection Using Medical Image Analysis. 10.1007/978-3-319-61316-1(Chapter 8), p159–186,2018.

[24] Elnakib, Ahmed; M. Amer, Hanan; E.Z. Abou-Chadi, Fatma. Early Lung Cancer Detection using Deep Learning Optimization. International Journal of Online and Biomedical Engineering (iJOE), 16(06), p1-13,2020.

[25] Wang, Xi; Chen, Hao; Gan, Caixia; Lin, Huangjing; Dou, Qi; Tsougenis, Efstratios; Huang, Qitao; Cai, Muyan; Heng, Pheng-Ann. Weakly Supervised Deep Learning for Whole Slide Lung Cancer Image Analysis. IEEE Transactions on Cybernetics, p1–13,2019.

[26] Raoof, Syed Saba; Jabbar, M A.; Fathima, Syed Aley. [IEEE 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) - Bangalore, India (2020.3.5-2020.3.7)] 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) - Lung Cancer Prediction using Machine Learning: A Comprehensive Approach., p108–115,2020.

[27] Shanthi, S.; Rajkumar, N. Lung Cancer Prediction Using Stochastic Diffusion Search (SDS) Based Feature Selection and Machine Learning Methods. Neural Processing Letters, p1-14,2020.

[28] Liu, Zhuo; Yao, Chenhui; Yu, Hang; Wu, Taihua. Deep reinforcement learning with its application for lung cancer detection in medical Internet of Things. Future Generation Computer Systems, p1-18,2019.

[29] Tulasi Krishna Sajja, Retz Mahima Devarapalli, Hemantha Kumar Kalluri. Lung Cancer Detection Based on CT Scan Images by Using Deep Transfer Learning. Traitement du Signal. Vol.36 (4), p339-344,2019.

[30] Schwyzer, Moritz; Ferraro, Daniela A.; Muehlematter, Urs J.; Curioni-Fontecedro, Alessandra; Huellner, Martin W.; von Schulthess, Gustav K.; Kaufmann, Philipp A.; Burger, Irene A.; Messerli, Michael. Automated Detection of Lung Cancer at Ultralow dose PET/CT by Deep Neural Networks - Initial results. Lung Cancer, p1-12,2018.

[31] Singh, Gur Amrit Pal; Gupta, P. K. Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans. Neural Computing and Applications, p1-15,2018.

[32] Ardila, Diego; Kiraly, Atilla P.; Bharadwaj, Sujeeth; Choi, Bokyung; Reicher, Joshua J.; Peng, Lily; Tse, Daniel; Etemadi, Mozziyar; Ye, Wenxing; Corrado, Greg; Naidich, David P.; Shetty, Shravya. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine, p1-24,2019.

[33] Avanzo, Michele; Stancanello, Joseph; Pirrone, Giovanni; Sartor, Giovanna. Radiomics and deep learning in lung cancer. Strahlentherapie und Onkologie, p1-9,2020.

[34] Ennifar, Eric. [Methods in Molecular Biology] Microcalorimetry of Biological Molecules Volume 1964 (Methods and Protocols) || Machine Learning Based Approach for Detection of Lung Cancer in DICOM CT Image., 10.1007/978-1-4939-9179-2(Chapter 15), p161–173,2019.

[35] Hadi Hashemzadeh;Seyedehsamaneh Shojaeilangari;Abdollah Allahverdi;Mario Rothbauer;Peter Ertl;Hossein Naderi-Manesh; . A combined microfluidic deep learning approach for lung cancer cell high throughput screening toward automatic cancer screening applications. Scientific Reports, p1-10,2021.

[36] Kancherla, Kesav; Mukkamala, Srinivas. [IEEE 2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) - Singapore, Singapore (2013.04.16-2013.04.19)] 2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) - Early lung cancer detection using nucleus segementation based features. , p91–95,2013.

[37] Xu, Yiwen; Hosny, Ahmed; Zeleznik, Roman; et.al.,. Deep Learning Predicts Lung Cancer Treatment Response from Serial Medical Imaging. Clinical Cancer Research, p1-11,2019.

[38] Saba, Tanzila. Recent advancement in cancer detection using machine learning: Systematic survey of decades, comparisons and challenges. Journal of Infection and Public Health, p1-15,2020.

[39] Shreyesh Doppalapudi;Robin G. Qiu;Youakim Badr;. Lung cancer survival period prediction and understanding: Deep learning approaches. International Journal of Medical Informatics, p1-12,2021.

[40] Chinmayi Thallam;Aarsha Peruboyina;Sagi Sai Tejasvi Raju;Nalini Sampath; . Early-Stage Lung Cancer Prediction Using Various Machine Learning Techniques. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), p1-8,2020.

[41] Coudray, Nicolas; Ocampo, Paolo Santiago; Sakellaropoulos, Theodore; Narula, Navneet; Snuderl, Matija; Fenyö, David; Moreira, Andre L.et.al., Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. Nature Medicine, p1-13,2018.

[42] Kadir, Timor; Gleeson, Fergus Lung cancer prediction using machine learning and advanced imaging techniques. Translational Lung Cancer Research, vol.7(3), 304–312,2018.

[43] Munir, Khushboo; Elahi, Hassan; Ayub, Afsheen; Frezza, Fabrizio; Rizzi, Antonello. Cancer Diagnosis Using Deep Learning: A Bibliographic Review. Cancers, vol.11(9), p1-36,2019.

[44] Nasrullah, Nasrullah; Sang, Jun; Alam, Mohammad S.; Mateen, Muhammad; Cai, Bin; Hu, Haibo. Automated Lung Nodule Detection and Classification Using Deep Learning Combined with Multiple Strategies. Sensors, vol.19(17), p1-19,2019.

[45] Mhaske, Diksha; Rajeswari, Kannan; Tekade, Ruchita. [IEEE 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA) - Pune, India (2019.9.19-2019.9.21)] 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA) - Deep Learning Algorithm for Classification and Prediction of Lung Cancer using CT Scan Images., p1–5,2019.

[46] Liao, Zhijun; Li, Dapeng; Wang, Xinrui; Li, Lisheng; Zou, Quan. Cancer Diagnosis Through IsomiR Expression with Machine Learning Method. Current Bioinformatics, vol.13(1), p57–63,2018.

[47] Yu-Jen Chen, Yu-Jen; Hua, Kai-Lung; Hsu, Che-Hao; Cheng, Wen-Huang; Hidayati, Shintami Chusnul . Computer-aided classification of lung nodules on computed tomography images via deep learning technique. OncoTargets and Therapy, p1-8,2015.

[48] R. Raja Subramanian, R. Nikhil Mourya, V. Prudhvi Teja Reddy, B. Narendra Reddy, Srikar Amara. . Lung Cancer Prediction Using Deep Learning Framework. International Journal of Control and Automation. Vol.13 (3), p154-160,2020.

[49] Vikul J. Pawar1 , Kailash D. Kharat2 , Suraj R. Pardeshi3 , Prashant D. Pathak.. Lung Cancer Detection System Using Image Processing and Machine Learning Techniques. International Journal of Advanced Trends in Computer Science and Engineering. Vol.9 (4), p5956-5963,2020.

[50] Chabon, Jacob J.; Hamilton, Emily G.; Kurtz, David M.et,al . Integrating genomic features for non-invasive early lung cancer detection. Nature, p1-34,2020.

[51] Vinod Kumar;Brijesh Bakariya;. Classification of malignant lung cancer using deep learning. Journal of Medical Engineering &amp; Technology, p1-10,2021.

[52] Kriegsmann, Mark; Haag, Christian; Weis, Cleo-Aron; Steinbuss, et.al; Deep Learning for the Classification of Small-Cell and Non-Small-Cell Lung Cancer. Cancers, vol. 12(6), p1-15,2020.

[53] Huan Yang;Lili Chen;Zhiqiang Cheng;Minglei Yang;Jianbo Wang;Chenghao Lin.et.al.,;. Deep learning-based six-type classifier for lung cancer and mimics from histopathological whole slide images: a retrospective study. BMC Medicine, p1-14, 2021.

[54] Hosny, Ahmed; Parmar, Chintan; Coroller, Thibaud P.;et.al.. Deep learning for lung cancer prognostication: A retrospective multi-cohort radiomics study. PLOS Medicine, vol.15(11), p1-25,2018.

[55] Selvathi, D.; AarthyPoornila, A. [IEEE 2017 International Conference on Signal Processing and Communication (ICSPC) - Coimbatore, India (2017.7.28-2017.7.29)] 2017 International Conference on Signal Processing and Communication (ICSPC) - Performance analysis of various classifiers on deep learning network for breast cancer detection. , p359–363,2017.

[56] Sanjukta Rani Jena;S. Thomas George;D. Narain Ponraj; Lung cancer detection and classification with DGMM-RBCNN technique . Neural Computing and Applications, p1-17,2021.

[57] Pham, Hoa Hoang Ngoc; Futakuchi, Mitsuru; Bychkov, Andrey; Furukawa, Tomoi; Kuroda, Kishio; Fukuoka, Junya. Detection of lung cancer lymph node metastases from whole-slide histopathological images using a two-step deep learning approach. The American Journal of Pathology, p1-38,2019.

[58] Hu, Zhengbing; Petoukhov, Sergey; Dychka, Ivan; He, Matthew. [Advances in Intelligent Systems and Computing] Advances in Computer Science for Engineering and Education Volume 754 || Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer. , 10.1007/978-3-319-91008-6(Chapter 63), p638–647,2019.

[59] Sun, Wenqing; Zheng, Bin; Qian, Wei. Automatic Feature Learning Using Multichannel ROI Based on Deep Structured Algorithms for Computerized Lung Cancer Diagnosis. Computers in Biology and Medicine, p1-15,2017.

[60] S. Lalitha. An automated lung cancer detection system based on machine learning algorithm. Journal of Intelligent &amp; Fuzzy Systems, p1-10,2021.

[61] Tiantian Fang. A Novel Computer-Aided Lung Cancer Detection Method Based on Transfer Learning from GoogLeNet and Median Intensity Projections. IEEE International Conference on Computer and Communication Engineering Technology (CCET), p1-5,2018.

[62] Gupta, Madhuri; Gupta, Bharat. [IEEE 2018 Second International Conference on Computing Methodologies and Communication (ICCMC) - Erode, India (2018.2.15-2018.2.16)] 2018 Second International Conference on Computing Methodologies and Communication (ICCMC) - A Comparative Study of Breast Cancer Diagnosis Using Supervised Machine Learning Techniques., p997–1002,2018.

[63] M.Siddardha Kumar;K. Venkata Rao;. Prediction of Lung Cancer Using Machine Learning Technique: A Survey. 2021 International Conference on Computer Communication and Informatics (ICCCI), p1-5,2021.

[64] Venkata Tulasiramu Ponnada, S.V. Naga Srinivasu. Efficient CNN for Lung Cancer Detection. International Journal of Recent Technology and Engineering (IJRTE). Vol.8 (2), p1-5,2019.

[65] Min Jae Cha, MD, Myung Jin Chung, MD, PhD, Jeong Hyun Lee, MD, and Kyung Soo Lee, MD. Performance of Deep Learning Model in Detecting Operable Lung Cancer with Chest Radiographs.vol.34 (2), p1-6,2019.

[66] Katiyar, Preeti; Singh, Krishna. [IEEE 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) - Noida, India (2020.2.27-2020.2.28)] 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) - A Comparative study of Lung Cancer Detection and Classification approaches in CT images. , p135–142,2020.

[67] Wang, Shidan; Yang, Donghan M.; Rong, Ruichen; et.al.. Artificial Intelligence in Lung Cancer Pathology Image Analysis. Cancers, vol.11(11), p1-16,2019.

[68] El Houby, Enas M.F. A survey on applying machine learning techniques for management of diseases. Journal of Applied Biomedicine, p1-10,2018.

[69] Gang, Peng; Zhen, Wang; Zeng, Wei;et.al. [IEEE 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI) - Xiamen, China (2018.3.29-2018.3.31)] 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI) - Dimensionality reduction in deep learning for chest X-ray analysis of lung cancer. , p878–883,2018.

[70] Swati Mukherjee;S. U. Bohra. Lung Cancer Disease Diagnosis Using Machine Learning Approach. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), p1-5,2020.

[71] Hussain, Lal; Aziz, Wajid; Saeed, Sharjil;et.al. [IEEE 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) - New York, NY, USA (2018.8.1-2018.8.3)] 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) - Automated Breast Cancer Detection Using Machine Learning Techniques by Extracting Different Feature Extracting Strategies. , p327–331,2018.

[72] Bijaya Kumar Hatuwal1 , Himal Chand Thapa. Lung Cancer Detection Using Convolutional Neural Network on Histopathological Images. International Journal of Computer Trends and Technology. Vol.68 (10), p21-24,2020.

[73] Dr. Akey Sungheetha and Dr. Rajesh Sharma R. Comparative Study: Statistical Approach and Deep Learning Method for Automatic Segmentation Methods for Lung CT Image Segmentation. Journal of Innovative Image Processing. Vol.2 (4), p187-193,2020.

[74] Ait Skourt, Brahim; El Hassani, Abdelhamid; Majda, Aicha. Lung CT Image Segmentation Using Deep Neural Networks. Procedia Computer Science, vol.127, p109–113,2018.

[75] Salaken, Syed Moshfeq; Khosravi, Abbas; Khatami, Amin; et.al. [IEEE 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) - Windsor, ON, Canada (2017.4.30-2017.5.3)] 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) - Lung cancer classification using deep learned features on low population dataset. , p1–5,2017.

[76] Atharva Bankar;Kewal Padamwar;Aditi Jahagirdar; Symptom Analysis using a Machine Learning approach for Early Stage Lung Cancer . 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), p1-5,2020.

[77] Lustberg, Tim; van Soest, Johan; Gooding, Mark; Peressutti, Devis; Aljabar, Paul; van der Stoep, Judith; van Elmpt, Wouter; Dekker, Andre (2017). Clinical evaluation of atlas and deep learning based automatic contouring for lung cancer. Radiotherapy and Oncology, p1-6.

[78] Bhandary, Abhir; Prabhu, G. Ananth; Rajinikanth, V.;et.al. Deep-Learning Framework to Detect Lung Abnormality – A study with Chest X-Ray and Lung CT Scan Images. Pattern Recognition Letters, p1-11,2019.

[79] Masood, A., Sheng, B., Li, P., Hou, X., Wei, X., Qin, J., & Feng, D.. Computer-Assisted Decision Support System in Pulmonary Cancer detection and stage classification on CT images. Journal of Biomedical Informatics, vol. 79, p117–128,2018.

[80] Bharati, S., Podder, P., &Mondal, M. R. H. Hybrid deep learning for detecting lung diseases from X-ray images. Informatics in Medicine Unlocked, vol.20, p1-14,2020.

Question Classification in Albanian Through Deep Learning Approaches

Evis Trandafili, Nelda Kote, Gjergj Plepi
Faculty of Information Technology
Polytechnic University of Tirana
Tirana, Albania

***Abstract*—In recent years, there is growing interest in intelligent conversation systems. In this context, Question Classification is an essential subtask in Question Answering systems that determines the question type, therefore, also the type of the answer. However, while there is abundant research for English, little research work has been carried out for other languages. In this paper we deal with classification of questions in the Albanian language which is considered a complex Indo-European language. We employ both machine learning and deep learning approaches on a large corpus in Albanian based on the six-class TREC dataset with approximately 5000 questions. Experiments with and without stop-words show that the impact of stop-words is significant in the accuracy of the classifier. Extensive comparison of algorithms for the task of question classification in Albanian show that deep learning algorithms outperform conventional machine learning approaches. To the best of our knowledge this is the first approach in literature for classifying questions in Albanian and the results are highly comparable to English.**

Keywords—Question classification; deep learning; BiLSTM; transformer; RoBERTa; Albanian corpus; natural language processing

I. Introduction

Large amounts of unstructured text data are generated every day resulting in a continuously increasing demand for information retrieval. Intelligent conversational agents that use Artificial Intelligence (AI) algorithms can facilitate user interaction with computer systems, conduct a conversation and answer natural language questions. In this context, Natural Language Processing (NLP), as a branch of AI, deals with automatic comprehension, interpretation and manipulation of natural language. However, natural language is complex and implies many different forms of communication facets leading to the need to interpret not only narrative or confirmative phrases but also questioning ones. Therefore, question answering systems go beyond the simple retrieval of relevant documents and aim at generating answers in natural language [1], [2].

Question Classification (QC) is the core component of a QA system that directly affects the retrieved or generated answer. QC deals with the assignment of question labels based on the corresponding answer type [3], [4]. This process is usually modelled as a text classification problem employing Machine Learning and Deep learning approaches. A lot of research works has been done to develop intelligent systems for QC in different languages and the recent performance of state of the art QA systems is impressive. However, QA systems lack in performance on low resource languages and create linguistic barriers that hinder the free flow of knowledge, business and communication.

In this paper we deal with question classification for a low resource language like Albanian through modelling this task as a classification problem. An Albanian corpus composed of approximately 5000 questions based on Text REtrieval Conference (TREC) Question Classification dataset with 6 classes of questions has been created. This corpus has been used with three traditional machine learning algorithms Support Vector Machines, Random Forest and Logistic Regression. For each classification algorithm we have explored the use of different approaches to extract information from raw text data, the classical approach tf-idf, FastText embeddings and the RoBERTa pre-trained language model. In addition, the corpus has been used in other experiments that employe three deep learning models, BiLSTM with and without attention, Transformer and RoBERTa. The overall evaluation of the models has been assessed using accuracy, precision, recall and F-score.

The experimental results show that stop-words have a high impact on the classification accuracy. Experimental comparison of different models shows that deep learning algorithms outperform the traditional machine learning approaches. Finally, although a first attempt for Albanian, our approach is comparable with the same approaches for English, leading to promising results for building question answering systems in Albanian and similar low resource languages.

The content of the paper is organized as follows: In Section II we cover Question Classification background and related works. In Section III we present the technologies and software libraries that we used in our experimental environment. In Section IV the Albanian dataset is introduced and its structure is explained in detail. Moreover, in Sections V and VI we introduce learning models, experimental design and analyze the results. Finally, we conclude with Section VII.

II. Related Work

There has been relatively little research done in the field of question classification tasks in the Albanian language. The first paper in this field treated this task as a traditional classification problem. The Albanian question corpus was used to evaluate the performance of three question classification models, which utilized the Support Vector Machine (SVM), Logic Regression

(LR) and Random Forest (RF) algorithms. The SVM model achieved the highest accuracy of 75.7% using FastText, while the use of RoBERTa resulted in a slightly lower accuracy of 0.6% [5]. The question-answering system proposed for the Albanian language in [6] does not implement the question classification component. The system has three modules: the first module preprocesses and indexes the documents, the second module analyzes the question, and the third module retrieves passages and extracts answers.

The current state of the question classification task in the Albanian language is still in its early stages, and further research and development are needed to enhance the performance of classification models. Nonetheless, the available approaches show promising results. Therefore, we will focus on existing studies related to languages other than Albanian.

The three components of question answering systems are question analysis, information retrieval, and answer analysis. The initial step of the question analysis component is question classification, which aims to determine the type of question and, consequently, the type of answer required. The question analysis component and its question classification step play an important role in the question answering systems as it can identify the correct class of a question and as a result restrict the possible answers [3, 4].

In question answering systems, the question type is a key factor that determines which type of questions a system can answer. These systems can be factoid, which answer questions related to facts, or non-factoid, which answer questions like mathematical calculations [7].

Previous work on the question classification problem can be analyzed from different aspects, such as the question classification categories, classification methods and algorithms, features, and so on. The answer to a question depends on the type of question, and questions can be classified into a category based on their common linguistic properties. Furthermore, their answers can also share common linguistic properties [8].

Numerous research studies have focused on developing artificial intelligence systems for question classification in various languages, including English [9], German [10], Italian [11], French [12], Spanish [13], and others.

The TREC [14] dataset, a popular resource for question classification tasks, is widely used in English and has also been translated into various other languages. This dataset contains a large collection of labeled questions and has been instrumental in the development and evaluation of many questions classification models. It is a dataset of 6,000 English questions, categorized into a main classification schema with six labels (human, description, abbreviation, entity, location, and numeric value,) and a fine-grained schema for each of the main categories. In [15], two classes (yes-no-explain and list) were added to the main taxonomy to improve question classification accuracy. The "yes-no-explain" class is used for questions that require a yes or no answer with an accompanying explanation. On the other hand, the "list" class is used for questions that have answers from a predefined list. An additional 250 questions were also added to the dataset. In [16], a Portuguese version of the TREC corpus was proposed.

The DISEQuA Corpus [17], a multilingual question answering corpus that uses a 7-class label classification schema (location, date, object, measure, person, organization, and other) is another widely used corpus. Additionally, paper [18] proposes a much larger taxonomy of 180 classes, making it one of the biggest taxonomies used in QA systems for question classification.

The SVM algorithm is frequently utilized in question classification tasks. In [19], an Arabic version of the TREC dataset is introduced and employed to train a two-stage classification model. The model involves using either the SVM, RF, or ME algorithm, followed by a CNN neural network. The machine learning algorithm is utilized to predict the primary class of the question in the first phase, and then the CNN is used in the second phase to predict the subclass. The SVM algorithm with CNN achieved the best performance, with an accuracy of 89%.

The use of top-words and dependency relations as features with the SVM algorithm in paper [20] resulted in an accuracy of 93.4% in the question classification task using the TREC dataset. Furthermore, the proposed solution in [21] to use syntactic and semantic analysis with SVM algorithms improves the performance of the model compared to the state-of-the-art models. The proposed hybrid method in [22] using the SVM algorithms with semantic and lexical feature extraction results in an accuracy of 96% on the TREC dataset.

The grammatical-based framework proposed in [23] performs better when using the J48 algorithm than SVM in factoid question classification approach, achieving an accuracy of 95.8%. The three main features of this approach are grammatical features, domain-specific features, and patterns.

In [24], the author proposed a Question Classification approach for Italian based on word embedding with sub-word information and Convolutional Neural Networks. The proposed approach is tested on the TREC dataset of questions in English and the Italian translated version, by using advanced vectors learned in an unsupervised manner using the skipgram model and character-based information to initialize the word embeddings. The Italian model achieved an accuracy of 80.42% using FastText, which is comparable to the English model accuracy of 80.16%.

In [25], the author proposed two models based on LSTM networks for question classification. These models were trained and evaluated on the heterogenous TREC and USC dataset, which features a two-level hierarchy annotation schema. The first model was used solely for predicting the main class of the question, while the second model added sub-class prediction to the question classification. The models achieved high accuracy values of 91.20% for the main class and 82.20% for the sub-class using 1000 h dimensions. The authors concluded that these types of networks are highly effective in the question classification task.

In [26], the author proposed the use of a BERT-based model for pregnancy question classification. Two attention mechanisms were evaluated: an additional layer on top of

BERT and the built-in self-attention mechanism of BERT, which resulted in better classification accuracy than the traditional models. The model that used an additional layer on top of BERT achieved the highest accuracy of 88%.

The new approach, which leverages data augmentation to create extra training instances, is presented in [27]. The effectiveness of this approach is evaluated on the TREC question classification datasets using pre-trained models such as BERT, RoBERTa, and DeBERTa. The results indicate that the need for labeled instances is reduced by up to 81.7%, achieving a new state-of-the-art classification accuracy of 98.11% on the TREC dataset.

III. Technologies and Software Libraries

Learning models used in this paper are trained and tested using Python programming language. Python is the most preferred language for computer scientists and researchers working in the field of NLP because of the versatility to use API-s and libraries that enhance performance growth [28].

As a programming environment we used the Jupyter Notebook. It is a free, open source, web based computational notebook that supports over 100 programming languages even if its name is a reference of three core programming languages, Julia, Python and R. Python is the most popular programing language with Jupyter notebook [29]. The notebook has two main components: The first component consists of front-end cells that hold text or code and are executed independently. This is very important for time consuming operations like dataset manipulations. The second component is the back-end kernel that executes the code inside the cells. Kernels can also be run on remote servers.

For our experimental environment, we installed Visual Studio IDE which fully supports work in Jupyter Notebook and Python. In the subsections below we present libraries and modules imported in Jupyter Notebook.

A. Pandas

Pandas is a mature Python library with a stable API used for data analysis [30]. Pandas library is used to deal with the .csv files where the dataset with questions is kept. Pandas offers build in functions that facilitate text manipulation. We used DataFrame, a two-dimensional data structure, to create the .csv file that contains both, questions in English, their equivalent in Albanian and the respective question tag.

B. Scikit-learn (Sklearn)

Ease of use and its computational efficiency make Scikit-learn (Sklearn) the most used Python library for supervised and unsupervised Machine Learning algorithms. It’s task-oriented interface enables easy comparisons of different machine learning algorithms for a given dataset [31].

In our Question Classification task, we used the implementation of Support Vector Machine (SVM), Random Forest and Logistic Regression from Scikit-learn to build Question Classification models [5]. The main object in our supervised learning is the Scikit-learn estimator that implements a *fit* method with two arguments: an array with data and an array with labels. After calculating model learned parameters in *fit* method, we use the *predict* method with a single question input to predict/classify the question. By using the *score* method of the estimator, we compute the accuracy by default.

C. PyTorch

PyTorch is a Python framework which is very popular in deep learning research community. It assembles usability together with performance. PyTorch has proved its overall speed on several common benchmarks. It supports code as a model, offers an easy debugging, is efficient with other libraries and supports hardware accelerators [32]. PyTorch provides an array-based programming model accelerated by GPUs. NLP applications are successfully used with PyTorch because of the tensors which allow the use of GPUs to perform complex computational calculations with significantly improved performance.

D. Hugging-Face

Hugging Face Transformer is a neural network architecture by Google Brain based on attention mechanisms to draw global dependencies between input and output. It outperformed traditional encoder-decoder architectures in translation tasks [33]. Hugging Face is a great source of pre-trained machine learning models based on several transformer architectures like BERT [34], RoBERTa [35], BART [36], etc.

IV. Dataset Preparation

The Albanian questions dataset used in this paper is built by translating into Albanian the six-class fact-based questions dataset from Text REtrieval Conference (TREC) [14], originally with 5452 training examples and 500 test examples. The six-class tags used to identify the type of answer are shown in the Table I below.

TABLE I. Six-Class Tags in Questions Dataset

Question tag	Abbreviation	Description
0	DESC	Description and abstract concepts
1	ENTY	Entities
2	ABBR	Abbreviation
3	HUM	Human beings
4	NUM	Numeric values
5	LOC	Locations

Raw translation from Albanian language was achieved using the library translate-api in python. Google Translate is the most used online translation service, therefore we used its interface for translating the dataset from English to Albanian. However, taking in consideration that Albanian language has a limited translation support, after the automatic translation we checked the dataset manually. Two main issues were noticed:

- The translation was not very accurate.
- Some questions lost their relations with the respective tags.

Therefore, we revised the dataset manually by correcting translations and by removing questions considered not appropriate for use in the dataset. As a result, the final dataset is composed of 4694 questions in Albanian. In Table II we show several examples of questions translated from English using the automatic translation.

The numbers of questions in each of the six classes in questions' dataset is shown in Fig. 1. We can easily observe that the class ABBR is highly unbalanced as the number of questions classified as abbreviations is merely 8% of the number of questions in other classes.

The problem of class imbalance is very popular in real world classification datasets. Questions asking for explanation of abbreviations are infrequent in our everyday life, as a result they are rarely found in written texts compared to other types of questions. This is clearly reflected even in questions' dataset. In this situation the classification risks to ignore the ABBR class.

The solution that we used to learn from the unbalanced questions dataset is the data augmentation technique. We manually increased the number of sentences in the ABBR class by creating new sentences through exchanging some words of the question with their synonyms.

TABLE II. EXAMPLES OF TRANSLATED QUESTIONS

Question tag	English Question	Albanian Question
DESC	How do you match a name to a social security number?	Si shoqërohet një emër me një numër të sigurimeve shoqërore?
ENTY	What's the only work by Michelangelo that bears his signature?	Cila është e vetmja vepër nga Michelangelo që mban firmën e tij?
ABBR	What is the abbreviation of the company name "General Motors"?	Cili është shkurtimi i emrit të kompanisë "General Motors"?
HUM	Who killed Gandhi ?	Kush e vrau Gandin?
NUM	How long does it take for your blood to make one complete trip through the body?	Sa kohë duhet për gjakun tuaj të bëjë një udhëtim të plotë përmes trupit?
LOC	What country is the world 's largest importer of cognac?	Cili vend është importuesi më i madh në botë i konjakut?

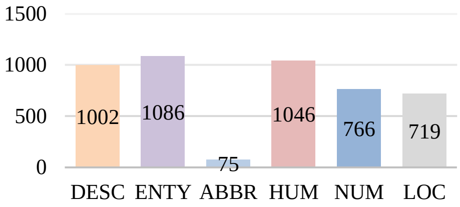


Fig. 1. Class distribution of questions.

Moreover, the most recommended pre-processing method that showed improvement in text classification tasks is stop-word removal. Stop-words are a set of commonly used words in a natural language that hold unimportant information in NLP applications. Their removal allows the model to learn from the most important words. However, the authors in [37] showed that stop-word removal does not always improve accuracy. Text Classification performance should always be fine-tuned by performing evaluations and determining the best combinations of pre-processing methods. As a pre-processing step, we used stop-word removal. We utilized the Spark-NLP [38] python library, which supports stop-word removal for Albanian language. A total of 223 frequently used words with low level information in Albanian language are removed from the dataset. Some examples of stop-words in Albanian are: "*ai*", "*ajo*", "*çfarë*", "*cili*", "*dhe*", "*unë*", etc. To test if stop-word removal is beneficial in terms of accuracy for our question classification model, we preserve a copy of the dataset, prior to stop word removal. The latter is also used for training the model and its accuracy is compared with the accuracy from models trained with the pre-processed dataset. We explore the effects of pre-processing with stop-words in Experiment 2 in Section VI.

In conventional Machine Learning algorithms, the words in the dataset are represented as numeric vectors or word embeddings in order for them to be fed as input to the models. In this paper we applied FastText [39], a popular neural network model used to generate word embeddings by splitting the word in n-grams. Due to its generalization capabilities and the ability to compute word embeddings for out-of-vocabulary words, FastText has shown good accuracy for small datasets and for low resource languages. In [5] we used tf-idf, FastText and RoBERTa to represent word embedding fed to Machine Learning models and the results showed that FastText outperformed in terms of accuracy.

V. LEARNING MODELS AND EXPERIMENTAL SETUP

We model the question classification task as a classification problem. If we are given a question u composed by a sequence of words $x = \{x_1, \cdots, x_n\}$, the classification model should predict the label y, by computing the probability $p\ (y \mid x)$. To effectively deal with this problem, we compared the accuracy of several classification algorithms divided in two main categories:

a) Classical machine learning algorithms: Support Vector Machines, Random Forest and Logistic Regression with tf-idf, FastText and RoBERTa as vector representations. We used the results obtained in [5].

b) Deep learning algorithms: *BiLSTM* [40] is a bidirectional LSTM which takes as input a sequence of vector representations for each word in the question/sentence and outputs the class/label of the question.

Transformer [33] is a new and powerful neural networks architecture developed by Google. In the classification context, it utilizes a similar approach with BiLSTM but using a different and more powerful architecture.

RoBERTa [35] is a transformer model pre-trained on large volumes of raw text data. For the Albanian language it is pre-trained using Wikipedia corpus. In our work, we used the RoBERTa pretrained Albanian tokenizer. RoBERTa transformer model takes as input a sequence of indexes from sentence words acquired from the tokenizer and outputs an array of base size 768 for each part of the sequence. The arrays are then unified, and the average value is calculated and fed to a feed-forward network with two layers and an activation function, in our case RELU.

Moreover, the deep learning models that we used for training have different numbers of learnable parameters, frequently used to measure the classification performance. In Table III, we show the number of parameters in each model.

TABLE III. NUMBER OF PARAMETERS FOR EACH MODEL

Model	No of Parameters
BiLSTM	3.8 M
BiLSTM with attention	4.8 M
Transformer	4 M
RoBERTa	83.7 M

RoBERTa is the model with the highest number of parameters, 83.7M, while BiLSTM is the model with the least number of parameters. If we assess sequence-to-sequence models, BiLSTM with attention has the highest number of parameters while Transformer and BiLSTM have comparable number of parameters.

Classical Machine Learning and Deep Learning models are trained on a computer with a 16 cores CPU and a high-performing graphics processing unit, GPU NVIDIA A100-PCIE 40GB.

The classical Machine Learning algorithms were evaluated using 5-fold cross validation. Each model spent approximately 1-2 minutes to build.

On the other hand, to build deep learning models we performed the 70/10/20 train/validation/test splitting technique. In Table IV the distribution of classes during the split is shown.

TABLE IV. OVERVIEW OF TRAIN/VALIDATION/TEST

Split	Total	DESC	ENTY	ABBR	HUM	NUM	LOC
Train	3191	691	739	51	706	540	464
Validation	564	116	131	6	133	85	93
Test	939	195	216	18	207	141	162

To avoid overfitting, an early stopping method is implemented in the validation set. The models were fine-tuned for 100 epochs and a batch size of 64. Furthermore, we applied the Adam optimizer [41] as the most recommended optimizer in reducing loss and improving accuracy. When training BiLSTM, BiLSTM with attention and Transformer models, we applied $1e-3$ learning rate. Whereas, for RoBERTa model we used $1e-4$ learning rate [35]. Each of deep learning models spent 15-20 minutes when trained in the beforementioned computer hardware. Deep learning models require powerful processing units in order to ensure efficiency and reduce time consumption.

VI. EXPERIMENTAL EVALUATION

In this section we present experimental design and evaluation of Albanian question classification models. We describe the objectives and outcomes of each experiment and highlight performance variations between classical machine learning algorithms, with deep learning models. Classification performance is evaluated using the common metrics of accuracy, precision, recall and F-score.

Experiment 1: Question Classification using the classical machine learning algorithms.

The implementation of FastText for vector representation is shown to increase the classification performance in all the classical machine learning models used for classification [5]. In Table V we show the performance of Support Vector Machine, Random Forest, and Logistic Regression where FastText is used to generate word embeddings. Support Vector Machine proves to be the best performing algorithm among Random Forest and Logistic Regression, with scoring 75.7% for accuracy and 80.0% for F1-score which is not satisfactory if we compare it with state-of-the-art Question Classification models for languages other than Albanian. However, classical Machine Learning algorithms are not time-consuming and require less powerful processing units.

TABLE V. CLASSIFICATION PERFORMANCE IN THE PRE-PROCESSED ALBANIAN DATASET

Models (Albanian)	Accuracy	F1-score macro	Precision	Recall
Support Vector Machine (SVM)	**75.7%**	**80.0%**	**71.6%**	**73.9%**
Logistic Regression (LogReg)	74.0%	79.6%	64.9%	66.1%
Random Forest (RnFor)	70.8%	78.6%	62.5%	64.9%

Experiment 2: Deep learning classification models using the preprocessed Albanian dataset.

The main goal of this experiment is to show the effects of preprocessing on question classification task in Albanian. As discussed in Section IV, as a preprocessing step we chose the stop-word removal. The most frequent Albanian words were removed from the questions dataset. The same Deep Learning models were trained using the preprocessed Albanian dataset. The results are shown in Table VI. RoBERTa proves to be the best performing model compared to BiLSTM, BiLSTM with attention and Transformer.

TABLE VI. DL CLASSIFICATION PERFORMANCE IN THE PRE-PROCESSED ALBANIAN DATASET

Models (Albanian)	Accuracy	F1-score macro	Precision	Recall
BiLSTM	80.8%	78.5%	81.7%	76.6%
BiLSTM + Attention	81.5%	81.0%	86.2%	78.2%
Transformers	85.3%	84.0%	85.1%	83.0%
RoBERTa	**87.6%**	**86.5%**	**85.5%**	**88.2%**

Experiment 3: Deep learning classification models using the Albanian dataset without prior preprocessing.

Questions are sentences with a specific structure that contains a question word which is classified a stop-word. The motivation behind this experiment is to measure the effect of stop-words in question classification in Albanian. The same models as in Experiment 2 were trained and tested on the Albanian questions dataset without removing stop-words from the dataset. The evaluation metrics of the models are shown in Table VII. RoBERTa outperforms the classification performance of other Deep Learning architectures.

In Fig. 2 we analyze the results of Experiment 2 and Experiment 3. Except from the Transformer, that is not significantly affected by stop-word removal, the other models show meaningful performance improvement when the stop-words are preserved in the dataset. When the stop-words were not removed, the accuracy of RoBERTa increased by 3.5%.

TABLE VII. CLASSIFICATION PERFORMANCE WITHOUT PRE-PROCESSING ALBANIAN DATASET

Models (English)	Accuracy	F1-score macro	Precision	Recall
BiLSTM	82.5%	81.3%	85.2%	79.2%
BiLSTM + Attention	83.2%	82.1%	83.0%	81.4%
Transformers	84.4%	82.3%	81.3%	83.4%
RoBERTa	**91.1%**	**90.0%**	**90.0%**	**90.2%**

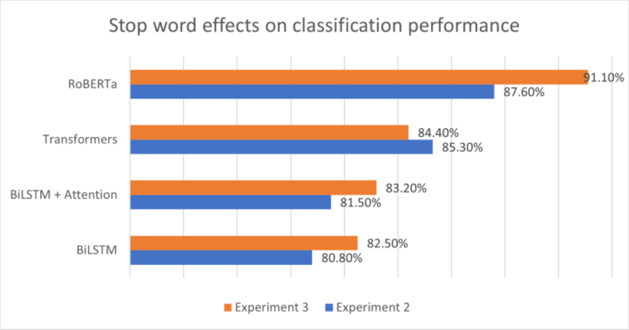


Fig. 2. Accuracy of DL models in experiments 2 and 3.

Experiment 4: Question Classification of deep learning models using the original TREC-6 dataset in English.

The main objective of this experiment is to determine a benchmark for the question classification performance. Albanian is an understudied language and the pretrained models are based on raw data. Furthermore, the automatic translation used to create the Albanian dataset, may negatively affect the performance of our model. English is a computationally rich and very well studied language. In order to assess if we have a decrease in performance due to low resources in Albanian language, we will define the state-of-the-art performance by training the models with the TREC dataset in English.

In Table VIII, we show the evaluation metrics calculated on our Deep Learning models trained with the original TREC-6 dataset. We show an increase in performance by adding attention to BiLSTM model. Furthermore, we confirm the superiority of Transformer in comparison to BiLSTM. The best performing classification is achieved using the RoBERTa pre-trained model with a benchmark from F-measure scoring 95.1%.

TABLE VIII. DL MODELS TRAINED WITH TREC-6 DATASET

Models (English)	Accuracy	F1-score macro	Precision	Recall
BiLSTM	81.8%	81.7%	80.9%	82.8%
BiLSTM + Attention	83.2%	83.2%	86.2%	81.3%
Transformers	86.0%	86.4%	88.1%	85.2%
RoBERTa	**94.0%**	**95.1%**	**95.0%**	**95.3%**

Experiment results show that Support Vector Machines model performs better compared to Random Forest and Logistic Regression, but its classification performance is still poor, with F1-score of 80%.

Among the deep learning models tested with the Albanian corpus dataset, RoBERTa performs better compared to BiLSTM and Transformer with an accuracy of 87.6%. We must state that all deep learning models performed better than SVM, Random Forest and Logistic Regression.

Stop word removal pre-processing task, applied on the Albanian questions dataset, shows a decrease in performance of the all the deep learning classifiers. Question classification with RoBERTa has an accuracy of 87.7% when trained with the dataset where the stop-words were previously removed, and the accuracy improved to 91.1% when trained with the dataset where the stop-words were not removed. We conclude that stop-words should not be removed from Albanian corpus prior to question classification.

The benchmark of 95.1% for the accuracy obtained by RoBERTa using the original TREC dataset in English shows that the performance of the question classification system in Albanian can be improved by working with the Albanian language from a computational linguistic perspective.

VII. CONCLUSION AND FUTURE WORK

Question Classification is a fundamental part of a Question Answering system. In this paper we have modeled the Question Classification problem in an Albanian language corpus using the state-of-the-art Deep Learning models and architectures. To the best our knowledge there is no previous research work addressing the problem of question classification in Albanian language using recent deep learning approaches. We have employed an Albanian questions dataset to train classification models based on classical machine learning algorithms and more recent deep learning approaches. The experiments, both with and without stop-words, demonstrate that the presence of stop-words significantly affects the accuracy of the classifier. Moreover, the comparison between algorithms indicates that deep learning algorithms outperform conventional machine learning approaches.

As future work, we intend to expand the size of the dataset and also increase its overall quality by adding linguistic expertise. In addition, we plan to analyze the impact of the length of the questions on the classifier performance. Finally, it would be interesting to investigate language alignment approaches in order to exploit well-established machine translation algorithms in the process of classification.

REFERENCES

[1] Plepi, J., Kacupaj, E., Singh, K., Thakkar, H. and Lehmann, J., "Context Transformer with Stacked Pointer Networks for Conversational Question Answering over Knowledge Graphs," in European Semantic Web Conference, LNISA,volume 12731, 2021.

[2] Kacupaj, E, Plepi, J., Singh, K., Thakkar, H. and Lehmann, J., "Conversational Question Answering over Knowledge Graphs with Transformer and Graph Attention Networks," in 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 850–862, ACL, 2021.

[3] Alanazi, S. S., Elfadil, N., Jarajreh, M. and Algarni, S., "Question Answering Systems: A Systematic Literature Review," International Journal of Advanced Computer Science and Applications, vol. 12, no. 3, pp. 495-502, 2021.

[4] Sati, A. B. B., Ali, M. A. S. and Abdou, Sh. M., "Arabic Text Question Answering from an Answer Retrieval Point of View: a survey," International Journal of Advanced Computer Science and Applications, vol. 7, no. 7, pp. 478-484, 2016.

[5] Kote, N., Trandafili, E. and Plepi, Gj., "Question Classification for Albanian Language: An Annotated Corpus and Classification Models," in Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC, Tirana, 2022.

[6] Trandafili, E., Mece, E., Kica, K. and Paci, H., "A Novel Question Answering System for Albanian Language.," in Proceedings of EIDWT 2018, Tirana, Albania, 2018.

[7] Kodra, L. and Kajo Meçe, E., "Question Answering Systems: A Review on Present Developments, Challenges and Trends," International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, pp. 217-224, 2017.

[8] Cortes, E. G., Woloszyn, V., Binder, A., Himmelsbach, T., Barone, D. and Moller, S., "An Empirical Comparison of Question Classification Methods for Question Answering Systems," in Proceedings of the 12th LREC, Marseille, 2020.

[9] A. Mohasseb, M. Bader-El-Den and M. Cocea, "Domain Specific Grammar based Classification for Factoid Questions," in Proceedings of 5th International Conference on Web Information Systems and Technologies, Vienna, Austria, 2019.

[10] A. Davidescu, A. Heyl, S. Kazalski, I. Cramer and D. Klakow , "Classifying German Questions According to Ontology-Based Answer Types," in Advances in Data Analysis, Berlin, 2007.

[11] M. Pota, M. Esposito and G. De Pietro, "Convolutional Neural Networks for Question Classification in Italian Language," in The 16th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET_17), Japan, 2017.

[12] A.-L. Ligozat, "Question Classification Transfer," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 2013.

[13] M. A. Garcıa Cumbreras, L. A. Urena Lopez and F. Martınez Santiago, "BRUJA: Question Classification for Spanish.Using Machine Translation and an English Classifier," in EACL 2006 Workshop on Multilingual Question Answering - MLQA06, 2006.

[14] X. Li and D. Roth, "Learning Question Classifiers," in COLING 2002: The 19th International Conference on Computational Linguistics, 2002.

[15] D. Metzler and W. B. Croft, "Analysis of Statistical Question Classification for Fact-based Questions," Information Retrieval, vol. 8, no. 3, p. 481–504, 2005.

[16] Â. Costa, T. Luís, J. Ribeiro and A. Crist, "An English-Portuguese parallel corpus of questions," in Proceedings of LREC'12, Istanbul, Turkey, 2012.

[17] B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Peñas, V. Peinado, F. Verdejo and M. de Rijke, "Creating the DISEQuA Corpus: A Test Set for Multilingual Question Answering," in Comparative Evaluation of Multilingual Information Access Systems, 2003.

[18] E. Hovy, U. Hermjakob and D. Ravichandran, "A Question/Answer Typology with Surface Text Patterns," in HLT '02: Proceedings of the second international conference on Human Language Technology Research, 2002.

[19] A. Aouichat, M. S. Hadj Ameur and A. Geussoum, "Arabic question classification using support vector machines and convolutional neural networks," in Natural Language Processing and Information Systems, 2018.

[20] S. Xu, G. Cheng and F. Kong, "Research on Question Classification for Automatic Question Answering," in 2016 International Conference on Asian Language Processing (IALP), 2016.

[21] T. Hao, W. Xie, Q. Wu, H. Weng and Y. Qu, "Leveraging question target word features through semantic relation expansion for answer type classification," Knowle dge-Base d Systems, vol. 133, pp. 43-52, 2017.

[22] Y. Liu, X. Yi, R. Chen, Z. Zhai and J. Gu, "Feature extraction based on information gain and sequential pattern for English question classification," The Institution of Engineering and Technology, vol. 12, no. 6, pp. 520-526, 2018.

[23] A. Mohasseb, M. Bader-El-Den and M. Cocea, "Classification of factoid questions intent using grammatical features," ICT Express, vol. 4, no. 4, pp. 239-242, 2018.

[24] M. Pota and M. Esposito, "Question Classification by Convolutional Neural Networks Embodying Subword Information," in International Joint Conference on Neural Networks (IJCNN), 2018.

[25] G. Di Gennaro, A. Buonanno, A. Di Girolamo, A. Ospedale and F. Palmieri, "Intent Classification in Question-Answering Using LSTM Architectures," in Progresses in Artificial Intelligence and Neural Systems. Smart Innovation, Systems and Technologies, vol 184., Singapore, Springer, 2020, pp. 115-124.

[26] X. Luo, H. Ding, M. Tang, P. Gandhi, Z. Zhang and Z. He, "Attention Mechanism with BERT for Content Annotation and Categorization of Pregnancy-Related Questions on a Community Q&A Site," in 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, 2020.

[27] C. Mallikarjuna and S. Sivanesan, "Question classification using limited labelled data," Information Processing & Management,, vol. 59, no. 6, 2022.

[28] S. Raschka, J. Patterson and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," Information 11, vol. 4, no. 193, 2020.

[29] J. M. Perkel, "Why Jupyter is data scientists' computational notebook of choice," Nature, vol. 563, no. 7729, pp. 145-146, 2018.

[30] W. McKinney, "Pandas: a foundational Python library for data analysis and statistics.," Python for high performance and scientific computing, vol. 14, no. 9, pp. 1-9, 2011.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapea, M. Brucher, M. Perrot and É. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825-2830, 2011.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steine, L. Fang, J. Bai and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32, 2019.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is All you Need.," in Advances in neural information processing systems 30, 2017.

[34] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of NAACL-HLT 2019, 2019.

[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[36] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[37] Y. HaCohen-Kerner, D. Miller and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," PLoS ONE, vol. 15, no. 5, 2020.

[38] V. Kocaman and D. Talby, "Spark NLP: Natural Language Understanding at Scale," Software Impacts, vol. 8, 2021.

[39] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information.," Transactions of the Association for Computational Linguistics, vol. 5, p. 135–146, 2017.

[40] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in In Proceedings of the IEEE International Joint Conference on Neural Networks(IJCNN), Montreal, QC, Canada, 2005.

[41] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in International Conference on Learning Representations (ICLR), San Diego, 2015.

An Efficient Source Printer Identification Model using Convolution Neural Network (SPI-CNN)

Naglaa F. El Abady[1], Hala H. Zayed[2], Mohamed Taha[3]
Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha, Egypt[1, 2, 3]
School of Information Technology and Computer Science (ITCS), Nile University, Egypt[2]

Abstract*—Document forgery detection is becoming increasingly important in the current era, as forgery techniques are available to even inexperienced users. Source printer identification is a method for identifying the source printer and classifying the questioned document into one of the printer classes. According to what we know, most earlier studies segmented documents into characters, words, and patches or cropped them to obtain large datasets. In contrast, in this paper, we worked with the document as a whole and a small dataset. This paper uses three techniques dependent on CNN to find the document source printer without segmenting the document into characters, words, or patches and with small datasets. Three separate datasets of 1185, 1200, and 2385 documents are used to estimate the performance of the suggested techniques. In the first technique, 13 pre-trained CNN were tested, and they were only used for feature extraction, while SVM was used for classification. In the second technique, a pre-trained neural network is retrained using transfer learning for feature extraction and classification. In the third technique, CNN is trained from scratch and then used for feature extraction and SVM for classification. Many experiments are done in the three techniques, showing that the third technique gives the best result. This technique achieved 99.16%, 99.58%, and 98.3% accuracy for datasets 1, 2, and 3. The three techniques are compared with some previously published papers, and found that the third technique gives better results.

***Keywords*—Document forgery; source printer identification (SPI); convolution neural network (CNN); transfer learning (TL); support vector machine (SVM)**

I. Introduction

Investigating and analyzing digital evidence to identify the details of a crime is known as digital forensics. To find, gather, and examine digital evidence, digital forensics uses a set of specialized tools and methods [1]. In the last ten years, the use of digital documents has exploded. These digital papers may include images of official contracts, bills, checks, and other documents. Maintaining a digital document to a paper copy is easier, cheaper, and more effective, but security is a challenge.

Printed documents as disputed or questioned evidence are fairly common in forensic investigations. Due to the rise in these situations and the greater usage of printers in document creation than handwritten documents, printer inspection has become a crucial requirement in questioned document analysis in recent years. Additionally, numerous printers have been involved in the widespread forgery of printed papers during the past 20 years. In these situations, it is crucial for the investigators to decide the type of printer used and to create a connection between the disputed document and the stated printer.

Personal computers, scanners, and printers can produce forged documents such as certificates, agreements, identity cards, lottery tickets, etc. Modern printers have such high resolution that it is difficult for normal persons to differentiate forged documents from real ones. Traditional approaches use chemical techniques to detect forgeries in printed documents [2],[3]. These procedures need laboratory tools and a specialist to evaluate the samples. Additionally, these methods take a long time and risk damaging the printed paper. Digital techniques, in contrast, use a reference scanner to turn printed papers into their digital equivalent. Using digital approaches for source printer identification makes distinguishing between documents printed on various printers possible. Since all the analysis is done digitally, it is quicker and more automatic.

The two basic digital techniques for detecting source printers are extrinsic (active) and intrinsic (passive). Finding extrinsic signatures, such as watermarks, digital signatures, printer serial numbers, and printing dates, is known as active research. It's so time-consuming and expensive that using it is practically impossible. On the other side, passive characterizes the printer by identifying intrinsic features. Prior research typically used the following techniques to extract statistical features from printed documents: Feature extraction techniques include the Discrete Wavelet Transform (DWT), Local Binary Pattern (LBP), Key Printer Noise Features (KPNF), Gray-Level Co-Occurrence Matrix (GLCM), Speeded Up Robust Features (SURF), Oriented FAST Rotated and BRIEF (ORB), Histogram of gradient (HOG), spatial filters, and others[4], [5], [14]–[16], [6]–[13]. The forensic classification systems adopt support vector machine (SVM), Random Forest (RF), and ensemble techniques. However, feature extraction, feature selection, and classification operations in the abovementioned approaches require much professional human participation.

Additionally, to obtain results that may be generalized, the entire procedure must be repeated multiple times using a random selection of training and testing samples. A branch of artificial intelligence is machine learning (AI) [7]. In general, machine learning attempts to recognize the structure of data and fit that data into models that are helpful and clear. Convolution Neural Network (CNN) is an artificial neural network suggested [17]. Its network structure for shared weights is comparable to simulations of genuine biological brain networks. This feature has the potential to simplify and decrease the number of parameters in the network model. Instead of using the complicated feature extraction and data reconstruction steps used in the conventional identification approach, CNN can use the image directly as the input. Convolutional, activation, pooling, and fully connected layers make up most of a typical CNN. Activation layers are used to enable nonlinear mapping, which enhances feature maps' capacity for expression. Fully connected layers are employed as the classifier and final output layers for classification tasks [18].

Deep learning's success is strongly connected to large amounts of data. A lack of training data can seriously affect the performance of deep learning models. Transfer learning was introduced to resolve this problem. It has many advantages, including reducing training time and improving neural network performance [19]. There are two approaches to implement transfer learning: feature extraction and fine-tuning. The pre-trained network is used as any other feature extractor when extracting features. In contrast to feature extraction, a new, fully connected head is built and layered over the base architecture when fine-tuning.

Deep learning needs a large number of datasets to give more accuracy. In this research, the problem of the large dataset is solved by using transfer learning. The transfer learning CNN was used to save time in creating a model from scratch, training, and small datasets. Transfer learning CNN is divided into two types: the first used a pre-trained model such as (AlexNet [20], VGGNet-16 [21], VGGNet-19, GoogleNet [22], ResNet-18[23], ResNet-50, ResNet-101, Inceptionv3 (24], SqueezeNet[25], XceptionNet [26], DarkNet-19[27], DarkNet-53, ShuffleNet [28]) for feature extraction, and the result was classified using other machine learning classification techniques. The second used a pre-trained model but changed the last layers. An SPI-CNN technique is developed to extract deep features that are fitted into an SVM for classification to get higher performance than with an SPI-CNN model alone. Three different data sets—the first with 10 printers and 1185 documents, the second with 20 printers and 1200 documents, and the third with 30 printers and 2385 documents—were used to test all of the earlier models. The following are some of the contributions of this work:

- The techniques are tested on 30 printers, whereas all previous studies only used a maximum of 20 printers.
- Training new CNN (SPI-CNN) from scratch adapted to this application. Despite their simplicity, neural networks have proven to be extremely successful in producing good results across all datasets.
- The proposed techniques work on a whole document without segmenting it into characters, words, or patches, which speeds up processing.
- An efficient pre-processing stage that combines histogram equalization and gamma correction is implemented, significantly improving the model's performance and increasing accuracy.

The following sections make up the entire paper. Section II briefly describes the related work for classifying the source printer of a printed text document. Section III contains a description of the specifics of our proposed approach. The efficiency of the proposed approach is investigated using detailed experiments. The proposed approach's description and outcomes have been explored in Section IV. Lastly, conclusions from this effort are presented in Section V.

II. Related Work

There are several procedures for detecting document manipulation. Most of these procedures detect the source of the printer to determine the types of printers used in the printing process [16]. The problem of source printer classification has received plenty of attention in the earlier decade [29]. This section will review the most common methods for authenticating a document and confirming that a legal printer printed it.

Mikkilineni et al. [4] introduced a printer identification process that uses an SVM classifier. They studied the impact of font size, font type, paper type, and printer age. Their printer identification technique works for various font sizes, paper types, and printer ages when those variables are constant. A novel color laser printer forensic algorithm is presented by [5]. It is based on an SVM classifier and noisy texture analysis. To estimate invisible noises, two filters are used: the Wiener filter and the 2D DWT filter. The noise texture is then analyzed using the GLCM. The machine classifier is trained and tested using 384 statistical features collected from the data. The proposed method achieves 99.3%, 97.4%, and 88.7% accuracy for brand, toner, and model recognition. In [6], the authors presented a method for detecting document forgeries based on Distortion Mutation of Geometric Parameters (DMGP) with translation and rotation distortion parameters. Both Chinese and English documents can be examined using this method. It can investigate documents based on separate characters. It is strong to JPEG compression and works well with documents of low resolution. The GLCM and DWT were utilized for texture feature extraction to examine the Chinese printed source to determine the impact of different output devices [7]. The feature selection techniques are used to choose the best feature subset, and an SVM is used to determine the source model of the documents. The average experimental results achieve a 98.64% identification rate, which is 1.27% higher than the previously known approach of GLCM. Many important statistical features, such as the Spatial filters, LBP, the Wiener filter, GLCM, the Gabor filter, DWT, Haralick, and SFTA features, are calculated using image processing techniques and data exploration techniques [8]. The highest rate of identification is achieved by the LBP method. It is considered superior to other methods in its various characteristics. In [9], presented a technique for analyzing the relationship between digital printers and printed Chinese characters. An SVM-based classification and feature selection decision fusion are used. The most significant features are methodically selected from GLCM, DWT, spatial, Wiener, and Gabor filters. The identification accuracy rate of the GLCM method gets the maximum rate compared with other approaches. In [10] presents a set of characteristics for describing geometric distortions at the text-line level. Experiments on 14 printers showed that the suggested system outperforms the current state-of-the-art method based on geometric distortion. It provides substantially higher accuracy when working with a limited training size constraint. A classifier trained using one page, one printer, one font, three different fonts, and 14 printers had an average classification accuracy of 98.85%. In [11], a proposed system utilized all printed letters simultaneously to identify the source printer from scanned images of printed documents. All printed letters, as well as local texture pattern-based features, are classified by a single classifier. The method was tested on a public dataset of 10 printers, and a new dataset of 18 printers scanned at 600 and 300 dpi resolution and produced in four different fonts. The authors of [12] identified the document source printer using a passive technique. Some feature extraction approaches have been deployed, such as Key Printer Noise Features (KPNF), Speeded Up Robust Features (SURF), Orientated FAST rotated, and BRIEF. Three classification procedures are considered for the classification job: k-NN, Random Forest, and Decision Tree. The majority vote was for these three classification techniques. Combining ORB, KPNF, and SURF with an RF classifier and adaptive boosting approach yielded the best accuracy of 95.1%. Printer identification using GLCM is presented in [13]. A feature vector is created by extracting a set of features from each character for each letter "e" in the document. Each feature vector is then classified using a 5-Nearest-Neighbor (5-NN) classifier. With training, this approach is unaffected by font type or size, although cross-font and cross-size testing yielded mixed results. A separate 5-NN classifier block for each character would be required to classify a document using all its characters, not only "e"s. The classifier becomes more complex as a result. Techniques for color and picture documents produced by inkjet printers must also be researched. A text-independent method for adequately describing source printers using deep visual Features has been applied [14]. Through transfer learning on a pre-trained CNN, the system could recognize 1200 papers from 20 different printers, including 13 laser and 7 inkjet printers. Solutions to learn discriminant-printing patterns directly from the provided data were found by Anselmo in [15]. This enabled him to reject any past beliefs about the distinctive printing artefacts of each printer. Results of the experiments demonstrated that the technique works better than its existing counterparts and is robust to noisy data. In [16], a novel technique is proposed based on SURF, Oriented Fast Rotated, and BRIEF feature descriptors. The Random Forest, Naive Bayes, k-NN, and other classifiers combinations were used for classification. The model could correctly classify the questioned papers and assign them to the relevant printer. The accuracy was 86.5% using a combination of Naive Bayes, k-NN, Random Forest classifiers, a straightforward majority voting system, and adaptive boosting techniques. A text-independent algorithm for detecting document forgeries based on source printer identification SPI is suggested by [30]. The image is divided into the top, middle, and bottom sections. The feature extraction algorithms HOG and LBP are employed. Classification approaches such as decision trees, k-NN, SVM, random forests, bagging, and boosting are considered for printer identification. The AdaBoost classifier achieves 96% classification accuracy, which is the highest.

III. Proposed Techniques

In this research, three distinct techniques are proposed for SPI and classifying the questioned document into one of the printer types. The proposed techniques are worked with the document as a whole and a small dataset. In the first technique, pre-trained CNN models with transfer learning for feature extraction are used. Feature maps can be extracted from any layer to train a classical classifier. It classifies the output using an SVM classifier, which means that the SoftMax layer of a CNN model is replaced with such an SVM. In the second technique, pre-trained CNN models are adjusted via transfer learning, which involves replacing the final fully connected or learnable layer of a CNN model with a new fully connected layer equal to the number of classes in the datasets. The third technique utilizes a convolutional neural network (CNN) model to resolve the SPI problem. The suggested framework (SPI-CNN) has the ability to dynamically learn and extract printer features. The SPI-CNN and a support vector machine (SVM) classifier used in this work were trained using various datasets. The datasets description, pre-processing, and details of CNN models are covered in the subsections below.

A. Datasets Description

To test the models, three different datasets are employed. The public dataset by Khanna et al. [31] consists of 20 printers (13 laser printers and 7 inkjet printers). There are a total of 60 pages distributed to each printer. All of a printer's documents are unique. Contracts, invoices, and scientific publications are the three types of documents that are included in the dataset. The second dataset includes printed documents and extracted characters from 10 printers. English and Portuguese documents are printed on each printer. The dataset is freely available on [32]. The third dataset, which comprises 30 printers, was created by combining the first and second datasets. Details of three different datasets used in training and testing are shown in Table I.

TABLE I. Details of Three Different Datasets used in Training and Testing

	Dataset_1 (Khanna et al., 2007)	**Dataset_2 (Ferreira et al., 2015)**	**Dataset_3**
Name	Anselmo Ferreira dataset	Khanna dataset	Hyper dataset
Number of printers	10	20	30
Number of documents	1185	1200	2385
Resolution	4922x6530	3312x4677	3312x467 and 4922x6530
Format	.tiff	.png	.tiff and .png

B. Pre-Processing

The pre-processing phase is utilized in the training and testing phases. For the pre-processing step, there are three methods: Histogram Equalization (HE), Gamma Correction, and resizing image. Histogram equalization (HE) [33], [34], helps normalize image grey-scale values and improve brightness discrimination between foreground and background images. The histogram function is written as (1)

$$H(f) = \frac{C(f) - C(f)_{min}}{(W \times H) - C(f)_{min}} \times (G_L - 1) \quad (1)$$

Where, $H(f)$ signifies the histogram function of the image, $C(f)$ identifies the cumulative function, $C(f)_{min}$ denotes the minimum non-zero value of the cumulative distribution function, $W \times H$ gives the image's number of pixels, and G_L defines the number of grey levels utilized.

Gamma correction is a nonlinear process to manage an image's overall brightness. Translating the values of the input intensity image to new values, improves the image's contrast. The Gamma is obtained by (2),

$$N_i = G(\alpha) \times (O_i)^{\frac{1}{\gamma}} \qquad (2)$$

Where, N_i stands for the new intensity value, O_i for the old intensity value, $G(\alpha)$ stands for the gray stretch parameter utilized to linearly scale the outcome on the image of [0, 255], and γ stands for the positive constant. Gamma can have any value between 0 and 1; infinite mapping is linear when it is 1. When less than 1, Gamma is weighed in terms of greater output values. The mapping is weighted toward lower output values if Gamma exceeds 1. Fig. 1 shows three Different Gamma corrections. Finally, the input image is resized to match each model's input size because each CNN model has an input size.

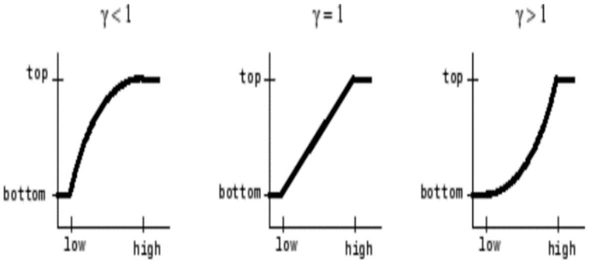


Fig. 1. Plots showing three different gamma correction settings.

C. The First Proposed Technique

Transfer learning (TL) for pre-trained models is a method that is suggested in this section for SPI. TL is used to prevent deep learning defects, which speeds up training and improves training outcomes. The pre-trained network with TL is considerably more comfortable and faster than the one trained from the start. Instead of building and training a new network, which requires millions of images, the system may quickly learn different jobs utilizing pre-trained deep networks. To transfer the learning capabilities to our application, transfer learning is employed rather than creating and training a deep learning model from scratch. 13 well-known pre-trained CNN models were used in this method to extract features (AlexNet, VGG-16, VGG-19, GoogleNet, DarkNet-19, DarkNet-53, ResNet-18, ResNet-50, ResNet101, SqueezeNet, XceptionNet, shuffleNet, inceptionv3). The obtained features are classified using SVM. In order to extract learned features from printer images, the pre-trained VGG-16 CNN model is used. The features are taken from one of the CNN layers and used to train an SVM classifier. Fig. 2 displays the model that extracts features using feature extraction and then classifies them using an SVM classifier.

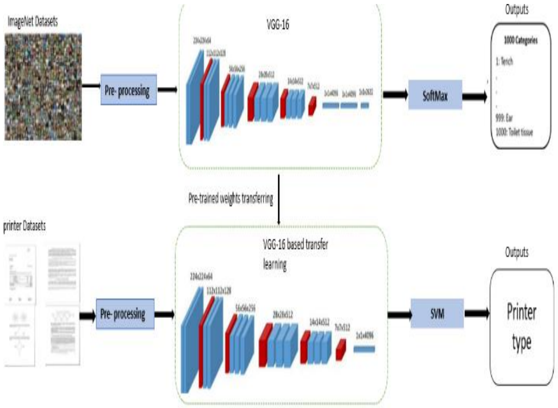


Fig. 2. Illustration of the first proposed SPI model based on transfer learning via extracting features and classifying using an SVM classifier.

D. The Second Proposed Technique

Transfer learning is based on using a CNN model that has been pre-trained and its weights that have been trained on enough data [35]. You can save time by using a pre-trained CNN model rather than creating a CNN from scratch, which requests a large, labeled dataset and lots of computational resources. While being preserved in other layers, the weights of the pre-trained CNN model are tuned in some. The higher layers of a pre-trained CNN (like DarkNet-53), initially designed for printer classification, are swapped out for the dense layer(s) in the proposed SPI approach to make the CNN compatible with SPI. In this technique, after the scanned documents for each dataset have been pre-processed. The number of classes in the current classification task is modified in every last FC layer neuron of pre-trained ConvNets ((AlexNet, VGG-16, VGG-19, GoogleNet, DarkNet-19, DarkNet-53, ResNet-18, ResNet-50, ResNet101, SqueezeNet, XceptionNet, shuffleNet, inceptionv3). With a very small learning rate of 0.0001 and 16 different batch sizes, the Adam Optimizer, also known as the Adaptive Learning Rate Algorithm, is employed to fine-tune the network. It is more efficient and less memory intensive. A model using transfer learning through fine-tuning is shown in Fig. 3.

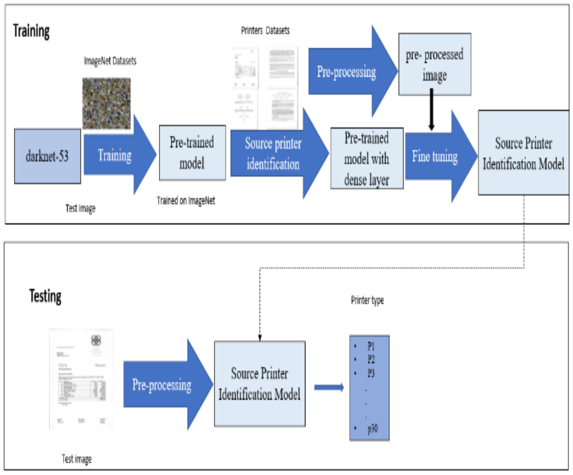


Fig. 3. Illustration of the second proposed SPI model based on transfer learning via fine-tuning in deep CNN.

E. The Third Proposed Technique (SPI_CNN)

The third technique suggests the CNN model as a solution to the SPI problem. The suggested framework (SPI-CNN) can dynamically learn and feature extraction for printers. This method uses a support vector machine (SVM) as the classifier and the SPI-CNN as the feature extraction technique.

1) SPI-CNN for features extraction: Four distinct models (7, 10, 13, and 17 layers) are used in SPI to choose the model with the highest degree of accuracy. Table II provides information about the various SPI-CNN models that were employed. Section B explains that all the dataset's scanned documents have been pre-processed. Following the pre-processing of all documents, the SPI-CNN model is applied, as shown in Fig. 4.

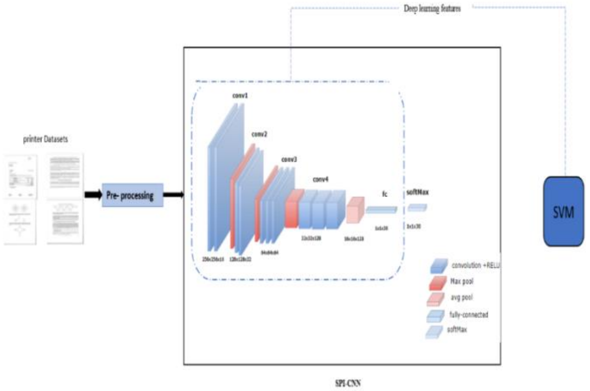


Fig. 4. The architecture of the third proposed SPI-CNN model.

TABLE II. DETAILS OF DIFFERENT MODELS OF CNN

Model_1	Model _2	Model _3	Model _4

Input Layer [256 256]
Conv1(5x5,16)
ReLU-Layer
AvgPool1(2x2)
Dropout Layer (0.5)
Fully Connected (num class)
SoftMax Layer
	Input Layer [256 256]
	Conv1(5x5,16)
	ReLU-Layer
	AvgPool1(2x2)
	Conv2(5x5,32)
	ReLU-Layer
	AvgPool2(2x2)
	Dropout Layer (0.5)
	Fully Connected (num class)
	SoftMax Layer
		Input Layer [256 256]
		Conv1(5x5,16)
		ReLU-Layer
		AvgPool1(2x2)
		Conv2(5x5,32)
		ReLU-Layer
		AvgPool2(2x2)
		Conv3(5x5,64)
		ReLU-Layer
		AvgPool3(2x2)
		Dropout Layer (0.5)
		Fully Connected (num class)
		SoftMax Layer
			Input Layer [256 256]
			Conv1(5x5,16)
			ReLU-Layer
			AvgPool1(2x2)
			Conv2(5x5,32)
			ReLU-Layer
			AvgPool2(2x2)
			Conv3(5x5,64)
			ReLU-Layer
			AvgPool3(2x2)
			Conv4(5x5,128)
			ReLU-Layer
			AvgPool4(2x2)
			Dropout Layer (0.5)
			Fully-Connected (num class)
			SoftMax Layer

The SPI-CNN is made up of layers arranged as follows:

- A multi-layer neural network is made up of different combinations of convolutional layers with a kernel size of 5 x 5 and (16,32,64,128) number of filters.
- A 2x2 kernel size average-pooling layer is used to aggregate the generated feature maps.
- Using a dropout layer with a probability of 0.5, we generate more robust features by randomly omitting various subsets throughout training.
- The final dense layer, which used a SoftMax function and various output neurons depending on the dataset, served as the classifier.
- ReLu was utilized as the activation function in each convolutional layer to learn complex functional mappings.

2) Classification with SVM: Although SoftMax succeeds in classification, current research has shown that the SVM classifier increases classification accuracy [36]. The SVM classifier in the current investigation replaced the SoftMax layer. To train the SVM, the outputs from the layer before (FC) are used as features. After training, it applies an SPI using the features gathered from the testing image.

IV. EXPERIMENTS AND DISCUSSIONS

This section describes the experimental technique and analyses the results. All of the earlier techniques were tested on three distinct data sets: the first with 10 printers and 1185 documents, the second with 20 printers and 1200 documents, and the third with 30 printers and 2385 documents. Evaluation of the first proposed technique's performance is discussed in Section 4.1, along with the performances of the second and third proposed techniques, which are considered in Sections 4.2 and 4.3, respectively. Finally, a discussion and comparison of the three methods to other techniques are provided.

The performance of the proposed techniques is estimated using accuracy metrics [37], [38], [39]. The accuracy is obtained using the following equation (3). It is defined as the percentage of perfectly classified images, where TP: True Positive, FN: False Negative, FP: False Positive, and TN: True Negative.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (3)$$

The suggested techniques were tested on a DELL PC using the following configuration implemented in MATLAB R2021b: Windows 11 64-bit, Intel(R) Core (TM) i7-11800H @ 2.30GHz, 6GHz GPU, 16GB RAM. Several tests were run to evaluate how well the suggested techniques worked.

F. Performance of the First Proposed Technique

The proposed technique is tested against 13 different pre-trained CNN models) AlexNet, VGG-16, VGG-19, GoogleNet, DarkNet-19, DarkNet-53, ResNet-18, ResNet-50, ResNet101, SqueezeNet, XceptionNet, shuffleNet, and inceptionv3 (with three distinct datasets. The number next to the model's name indicates its depth; thus, the models chosen are various, with varying depth sizes. The tests were carried out on both randomly selected 20% data as a test (i.e., 80% as the training set). When using pre-trained CNN as feature extractors and an SVM for classification, VGG-16 claims a maximum classification rate of 82.6% for dataset 1, 87.15% for dataset 2, and 86.67% for dataset 3. The accuracy of feature extraction and classification using pre-trained CNN and SVM with three different datasets is shown in Fig. 5.

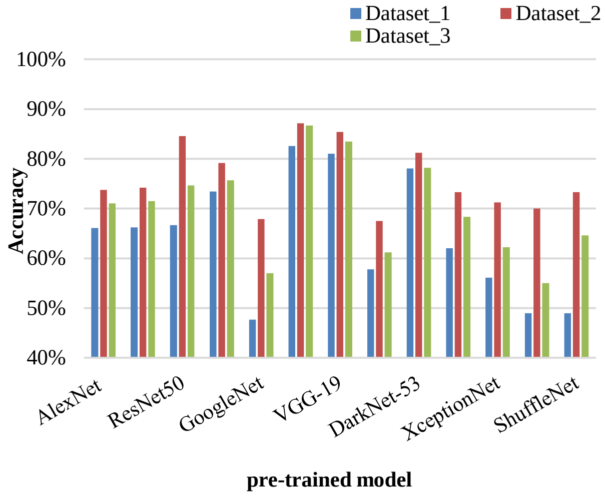


Fig. 5. The feature extraction and classification accuracy using pre-trained CNN and SVM with three different datasets (the first proposed model).

G. Performance of the Second Proposed Technique

As section D indicates, we use three separate datasets to fine-tune 11 pre-trained CNN models (AlexNet, VGG-16, VGG-19, GoogleNet, DarkNet-19, DarkNet-53, ResNet-18, ResNet-50, ResNet101, SqueezeNet, and shuffleNet). Deep transfer learning CNN architectures were used in this method to transfer learning weights, which reduced training time, mathematical calculations, and hardware resource utilization. Each dataset is split up into two parts: 20% for testing and 80% for training. The network architecture is as follows: The batch size for the 2D-CNN training was 16 samples. The Adam optimizer with a learning rate of 0.0001 was used. DarkNet-53 achieved a maximum classification rate of 98.31% for dataset 1, 97.5% for dataset 2, and 97.9% for dataset 3. Fig. 6 illustrates the performance of pre-trained CNN models during fine-tuning.

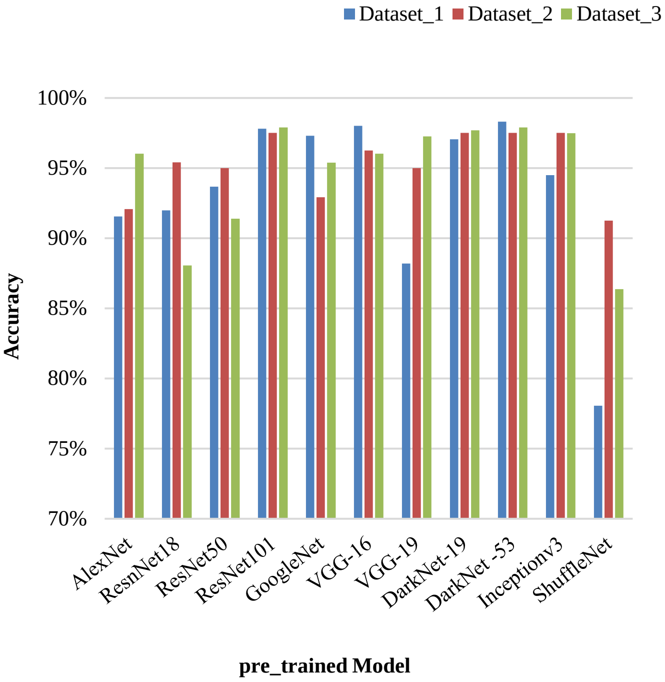


Fig. 6. The accuracy of pre-trained CNN via fine-tuning before and after pre-processing (the second proposed model).

H. Performance using the Third Proposed Technique

Three separate datasets were used to train the four CNN models that are shown in Table II. Neural network models with 1, 2, 3, and 4 convolutions were developed for comparison. Each dataset is split into 20% for testing and 80% for training. Following is the network architecture: 16 samples were used in the batch size for the 2D-CNN training. The Adam optimizer with a 0.0001 learning rate was employed. Four CNN models are used to train and classify each scanned document of a sample. Fig. 7 displays the accuracy attained by each SPI-CNN model using various datasets. Using model _4(SPI-CNN), which consists of four convolution layers, the average accuracy was 96.23% for dataset 3, 93.33% for dataset 2, and 96.2% for dataset 1. Fig. 8 displays the accuracy attained by each CNN+SVM model using a different dataset. Using model _4, which consists of four convolution layers, the average accuracy for datasets 1, 2, and 3 was 99.16%, 99.58%, and 98.32%, respectively. Results indicate that model 4 has the highest accuracy rate for SPI-CNN with SoftMax and SPI-CNN with SVM. Additionally, compared to its original configuration, the SVM classifier increased SPI-CNN accuracy by about 3%.

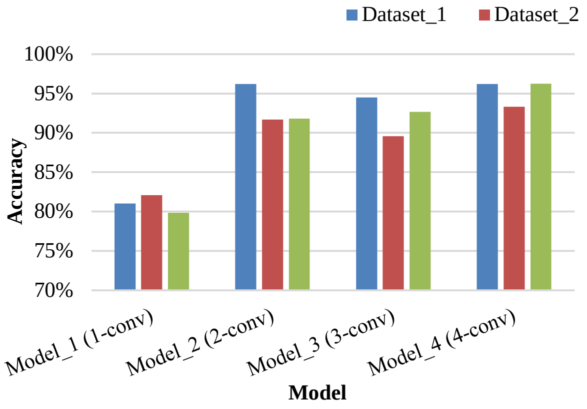


Fig. 7. Performance of SPI_CNN model.

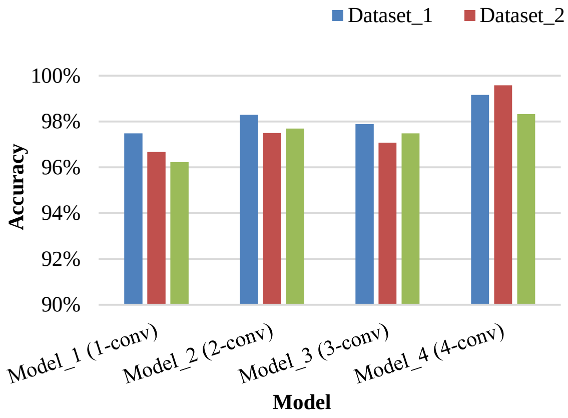


Fig. 8. Performance of SPI_CNN with SVM.

I. Discussion and Comparison

This section compares and discusses the results of three techniques using different datasets. Without dividing the document into letters, words, or patches and using only small datasets, three different CNN were trained to recognize the SPI. The first method was trained using SVM and used simply for feature extraction. The second had been trained in feature extraction and classification techniques. For feature extraction and classification, the third is trained completely from scratch. Because its parameters were tuned to extract features from printers document rather than other images, the third technique extracted features more effectively than the others. As illustrated in Fig. 9, our third proposed technique (SPI-CNN) outperforms [15], [40], [4], and [32] on both textural and deep learning features. As shown in Fig. 10, our third proposed technique (SPI-CNN) outperforms [37], [14], [12], [12], and [30] on dataset 2 of 20 printers and 1200 documents for both textural and deep-learned features. Fig. 11 compares the outcomes of the three proposed techniques for the data set 3 of 30 printers and 2385 documents. The prior outcomes lead us to the conclusion that the third model performs better than any previous method.

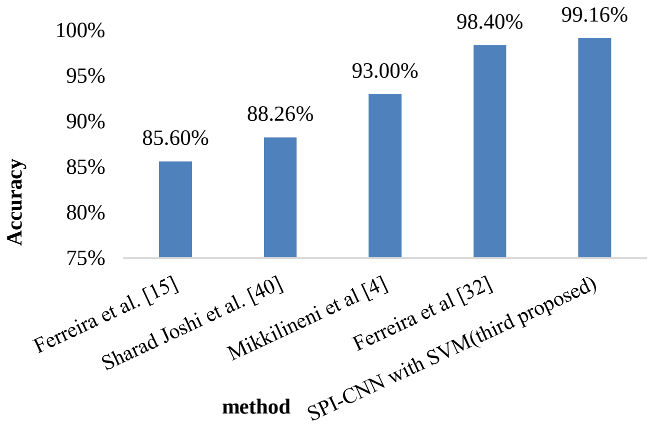


Fig. 9. Comparison of the third proposed technique with previous work on the dataset _1 of 10 printers.

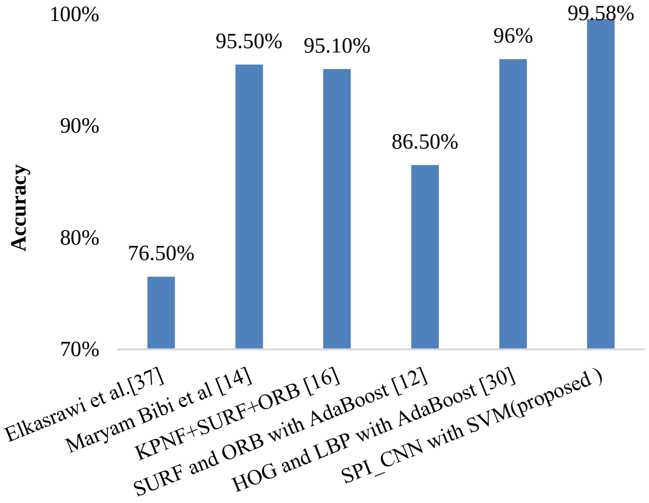


Fig. 10. Comparison of the third proposed technique with previous work on the dataset _2 of 20 printers.

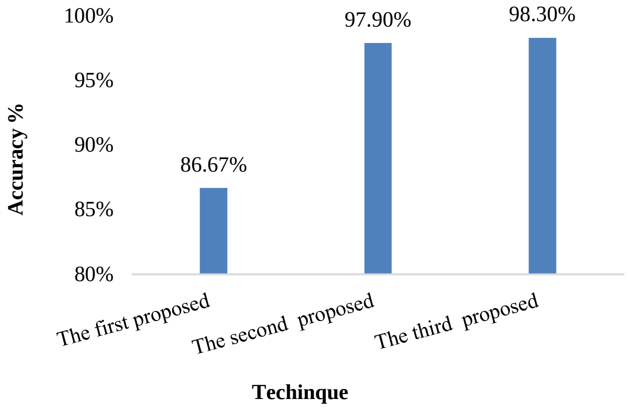


Fig. 11. Comparison of the three proposed techniques on the dataset _3 of 30 printers.

V. Conclusion

Three different techniques with CNN are proposed in this research to determine the printer's source. Although much research on source printer identification has been proposed, they have all been analyzed using distinct datasets and experimental setups. As earlier mentioned, several researchers use isolated characters in a text-dependent framework for experimental purposes. This paper uses CNNs to identify the source printer without segmenting the document into characters, words, or patches and with small datasets. An efficient pre-processing stage that combined histogram equalization and gamma correction was implemented, significantly improving the model's performance and increasing accuracy. The techniques are tested on a large number of 30 printers, whereas all previous studies only used a maximum of 20 printers. This paper trains three different CNN models on three separate datasets to determine the most accurate model. Transfer learning is used in the first technique for 13 pre-trained CNN models. These models serve as feature extractors, while SVM serves as a classifier. VGG-16 with SVM produces the best results. We tried 11 pre-trained models in the second technique but fine-tuned them by retraining each model and altering the last fully connected (The learning) layer. The fine-tuned DarkNet-53 achieves maximum classification rates. New CNN (SPI-CNN) from scratch adapted to this application in the third technique. The trained model was then used for feature extraction instead of SoftMax, and SVM was utilized as a classifier. Despite their simplicity, neural networks have proven to be extremely successful in producing good results across all datasets. The accuracy of the SPI-CNN model was 96.2%, 93.33%, and 96.23% for datasets 1, 2, and 3, respectively. For datasets 1, dataset 2, and dataset 3, the SPI-CNN-SVM model achieved 99.16%, 99.58%, and 98.3% accuracy, respectively. Based on the outcomes of the three techniques, we find that SPI-CNN with SVM is more accurate than the other two models. Additionally, the SVM classifier increased SPI-CNN accuracy by about 3% compared to its original configuration. With some previously published papers, the three techniques found that the third technique gives better results.

VI. Future Work

Future work will focus on discovering novel techniques to increase the accuracy of printer source identification. Try k-fold cross-validation as well rather than 20-80 validation. Identify forgeries in handwritten documents by looking at the type of ink used and the signature.

References

[1] J. A. Lewis, "Forensic document Eeamination: fundamentals and current trends", Elsevier, Oxford. 2014.

[2] A. Braz, M. López-López, and C. García-Ruiz, "Raman spectroscopy for forensic analysis of inks in questioned documents", Forensic Sci. Int., vol. 232, no. 1–3, pp. 206–212, 2013.

[3] P. C. Chu, B. Y. Cai, Y. K. Tsoi, R. Yuen, K. S. Y. Leung, and N. H. Cheung, "Forensic analysis of laser printed ink by X-ray fluorescence and laser-excited plume fluorescence", Anal. Chem., vol. 85, no. 9, pp. 4311–4315, 2013.

[4] A. K. Mikkilineni et al., "Printer forensics using SVM techniques", Int. Conf. Digit. Print. Technol., pp. 223–226, 2005.

[5] J. H. Choi, H. Y. Lee, and H. K. Lee, "Color laser printer forensic based on noisy feature and support vector machine classifier", Multimed. Tools Appl., vol. 67, no. 2, pp. 363–382, Nov. 2013.

[6] S. Shang, N. Memon, and X. Kong, "Detecting documents forged by printing and copying", EURASIP J. Adv. Signal Process., vol. 2014, no. 1, pp. 1–13, 2014.

[7] M. J. Tsai, J. S. Yin, I. Yuadi, and J. Liu, "Digital forensics of printed source identification for Chinese characters", Multimed. Tools Appl., vol. 73, no. 3, pp. 2129–2155, Oct. 2014.

[8] M. J. Tsai and I. Yuadi, "Digital forensics of microscopic images for printed source identification", Multimed. Tools Appl., vol. 77, no. 7, pp. 8729–8758, Apr. 2018.

[9] M. J. Tsai, C. L. Hsu, J. S. Yin, and I. Yuadi, "Digital forensics for printed character source identification", Proc. - IEEE Int. Conf. Multimed. Expo, vol. 2016–August, Aug. 2016.

[10] H. Jain, S. Joshi, G. Gupta, and N. Khanna, "Passive classification of source printer using text-line-level geometric distortion signatures from scanned images of printed documents", Multimed. Tools Appl., vol. 79, no. 11–12, pp. 7377–7400, Mar. 2020.

[11] S. Joshi and N. Khanna, "Single classifier-based Passive system for pource Printer classification using local texture features", IEEE Trans. Inf. Forensics Secur., vol. 13, no. 7, pp. 1603–1614, Jul. 2018.

[12] S. Gupta and M. Kumar, "Forensic document examination system using boosting and bagging methodologies", Soft Comput., vol. 24, no. 7, pp. 5409–5426, Apr. 2020.

[13] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. C. Chiu, J. P. Allebach, and E. J. Delp III, "Printer identification based on graylevel co-occurrence features for security and forensic applications", Secur. Steganography, Watermarking Multimed. Contents VII, vol. 5681, p. 430, Mar. 2005.

[14] M. Bibi, A. Hamid, M. Moetesum, and I. Siddiqi, "Document forgery detection using printer source identification-a text-independent approach", in 2019 International Conference on Document Analysis and Recognition Workshops, ICDARW 2019, 2019, vol. 2019–January, pp. 7–12.

[15] A. Ferreira et al., "Data-Driven Feature Characterization Techniques for Laser Printer Attribution", IEEE Trans. Inf. Forensics Secur., vol. 12, no. 8, pp. 1860–1873, 2017.

[16] M. Kumar, S. Gupta, and N. Mohan, "A computational approach for printed document forensics using SURF and ORB features", Soft Comput., vol. 24, no. 17, pp. 13197–13208, Sep. 2020.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", in Proceedings of the IEEE, 1998, vol. 86, no. 11, pp. 2278–2323.

[18] X. Yu, J. Wang, Q. Q. Hong, R. Teku, S. H. Wang, and Y. D. Zhang, "Transfer learning for medical images analyses: A survey", Neurocomputing, vol. 489, pp. 230–254, 2022.

[19] D. Larsen-Freeman, "Transfer of learning transformed", Lang. Learn., vol. 63, no. SUPPL. 1, pp. 107–129, 2013.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Adv. Neural Inf. Process. Syst., vol. 2, pp. 1097–1105, 2012.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., 2015.

[22] C. Szegedy et al., "Going deeper with convolutions", Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07–12–June, pp. 1–9, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016–Decem, pp. 770–778.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016–Decem, pp. 2818–2826, 2016.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer

parameters and <0.5MB model size", pp. 1–13, 2016, [Online]. Available: http://arxiv.org/abs/1602.

[26] F. Chollet, "Xception: Deep learning with depthwise separable convolutions", in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, vol. 2017–Janua, pp. 1800–1807.

[27] A. Redmon, J.; Farhadi, "YOLO9000: Better, faster, stronger", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, p. 7263–7271.

[28] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: an extremely efficient convolutional neural network for mobile devices", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.

[29] P. J. Chiang et al., "Printer and scanner forensics", IEEE Signal Process. Mag., vol. 26, no. 2, pp. 72–83, 2009.

[30] N. F. El Abady, M. Taha, and H. H. Zayed, "Text-independent algorithm for source printer identification based on ensemble learning", Comput. Mater. Contin., vol. 73, no. 1, pp. 1417–1436, 2022.

[31] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification using sensor pattern noise", Secur. Steganography, Watermarking Multimed. Contents IX, vol. 6505, p. 65051K, Feb. 2007.

[32] A. Ferreira, L. C. Navarro, G. Pinheiro, J. A. do. Santos, and A. Rocha, "Laser printer attribution: Exploring new features and beyond", Forensic Sci. Int., vol. 247, pp. 105–125, Feb. 2015.

[33] W. K. Pratt, "Digital Image Processing", vol. 19, no. 3. 1994.

[34] Anjani Suputri Devi D and Satyanarayana Ch, "An efficient facial emotion recognition system using novel deep learning neural network-regression activation classifier", Multimed. Tools Appl., vol. 80, no. 12, pp. 17543–17568, 2021.

[35] W. Tao, M. Al-Amin, H. Chen, M. C. Leu, Z. Yin, and R. Qin, "Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing", Procedia Manuf., vol. 48, pp. 926–931, 2020.

[36] Y. Tang, "Deep Learning using support vector machines", Accessed: Jan. 07, 2023. [Online]. Available: http://code.google.com/p/cuda-convnet.

[37] S. Elkasrawi and F. Shafait, "Printer identification using supervised learning for document forgery detection", Proc. - 11th IAPR Int. Work. Doc. Anal. Syst. DAS 2014, pp. 146–150, 2014.

[38] Y. Zhang, J. Goh, L. L. Win, and V. Thing, "Image region forgery detection: A deep learning approach", undefined, vol. 14, pp. 1–11, 2016.

[39] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-Class classification: an Overview", pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756.

[40] S. Joshi, M. Lomba, V. Goyal, and N. Khanna, "Augmented data and improved noise residual-based CNN for printer source identification", in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr. 2018, pp. 2002–2006.

The Cross-Cultural Teaching Model of Foreign Literature Under the Application of Machine Learning Technology

Jing Lv
School of Humanities and International Education, Xi’an Peihua University,
Xi’an 710125, Shaanxi, China

***Abstract*—As globalization spreads, so does the world's ethnic makeup, leading to a surge in cultural diversity that has become a major issue for the educational systems of all countries. Many western countries advocate for cross-cultural education (CCE) as a means of dealing with cultural variety and promoting trust, tolerance, and interaction between individuals of different backgrounds. The way to achieve this goal is to work toward solving the issue while also fostering greater national unity. One of the newest developments in international education is the concept of CCE, which has also given rise to a whole new area of research within the subject of education. Too much time has passed since students were actively engaged in the learning process, and the new curriculum reform has seriously harmed the traditional approaches to teaching foreign literature. As a result, the proposed research has shown that around half of all education is dedicated to the FL cross-cultural teaching paradigm. Chinese students' data were first gathered for this study and divided into two groups: Control Class (CC) and Experimental Class (EC). The performance of the students in both groups is then forecasted using the extreme gradient boosting (XGBoost) technique, which is based on machine learning. Then, we use an optimization method known as the Flower Pollination Algorithm (FPA) to improve XGBoost's prediction performance. According to the descriptive findings, students who adhere to the suggested teaching strategy show more learning interest than those who adhere to existing strategies.**

Keywords—Cross cultural education; foreign literature; extreme gradient boosting (XGBoost) algorithm; flower pollination algorithm (FPA); control class (CC)

I. Introduction

In order for Chinese students to achieve the English proficiency levels needed by their curriculum, a totally new piece of software known as the English displaying mode was developed [1]. It is possible that pupils' own levels of originality and academic success would rise if they were given extended periods of time on the computer to engage in speaking and listening activities [2]. It is possible that teachers might aid pupils in learning a language by introducing them to online reading, writing, and translation materials. Educators and students alike may anticipate an increase in opportunities to use multimedia in class as more and more resources become available for this purpose [3]. Multimedia courses in community nursing that make use of a digital library of teaching resources may improve students' interest in and recall of course content. There is strong evidence that incorporating YouTube into the classroom enhances students' ability to jhioijjj/iooooooooklglhk[poiuyt86867 6 565eeryyyyyyyyyyyyyyr7fdhjrfjtttkttt9tyeerse v sASZ z zjunderstand and remember what they read [4]. Students who saw the video on YouTube fared better than those who did not. The author learns why high school students study English online via a survey she administers to them. What makes an activity effective in meeting the requirements of the learner and fostering development is the "input and interaction" process. Here the author explores how music education might benefit from "smart classrooms" and other types of multimedia instruction. Today's multimedia productions rely significantly on the composition, editing, and synthesis capabilities of music software [5]. Since technology advancement has made it possible for students of any age and from any location to build their own learning plans, teachers need to rethink their responsibilities in the classroom. Additionally, students should be confident presenting and defending their views in class utilizing a number of methods [6]. Teachers may adapt their classes to their students' needs with the use of multimedia resources. As part of their education, students are expected to do their own study on topics that pique their curiosity or puzzle them, and to provide their own unique takes on the ideas they unearth. To some extent, computers may one day be used in lieu of human teachers because of the individualized attention they can provide to each student. Because not all pupils at a given school will have the same linguistic or cultural background, it is essential that schools have access to materials like this one [7]. A student like A may learn the same amount in three days whereas it could take the teacher a whole week. Student B may solve the issue or review it by clicking the mouse many times if necessary [8].

Internet of Things (IoT) advances has allowed educators to provide students with more engaging and valuable online learning opportunities, many of which include collaborative software and assignments. While it's true that learning English might be difficult, each individual should first take stock of their unique situation and objectives before making any broad assumptions. It is typical practice to survey educators prior to making major purchases like new textbooks or sets of worksheets [9]. To accommodate today's diverse student populations and make up for the time that might have been spent together in class if all students had the same previous knowledge and skills, new textbooks are being developed. The most valuable function [10] is that users may set playback to begin at a time and location of their choosing. The high quality of the provided resources, the ease with which they may be communicated, and the possibility of presenting them all at once are just a few of the many benefits of this approach. As a means to better serve users with varying network speeds, service providers may provide customers granular control over which resources are duplicated, while researchers may look at normal resource flows simultaneously [11]. Multimedia lectures might be very helpful for college students engaged in a wide range of digital building projects. Students may watch and take part in the lectures from anywhere with an internet connections, and professors are freed up to focus on other aspects of the curriculum [12]. Since there are a lot of great learning resources available online, yet text-based materials might be monotonous, streaming media technologies may be useful in the self-study sector. Visitors from prestigious institutions in the academic or corporate sector often deliver talks on campus, and they are almost always streamed live online [13]. If there are too many people, it may become awkward. With the various multimedia education tools now accessible, professors may now reach students all over the globe in real time via online streaming of lectures. The school should explore recording extracurricular activities like sports and cultural presentations to make them available to students online [14]. In order to connect with people from all walks of life, a command of the English language is required. Speed is of the essence in today's globalized world [15]. Learning English might make people more educated, tolerant, and successful. Learners of English should make it a goal to improve their essay writing skills. A student's writing skills may be judged not just by their vocabulary and grammatical understanding, but also by their sentence structure, discourse, and ability to think coherently [16]. As a result, teachers need to pay close attention to their pupils' language abilities while crafting assessments of their progress. Perhaps this explains why today's pupils seem to be more attentive in class than ever before.

Educators' support of students' growth in written English proficiency is more important than ever [17]. Only a small number of the available short English works are appropriate for usage in a classroom where instructors physically check student work. This technique will be less effective and more prone to erroneous conclusions if stricter grading standards and longer essay lengths are used [18]. Manual grading is also very susceptible to prejudice since it's impossible to know how a reader from a different section of the globe or cultural background would interpret the same work. Due to technological developments, Automated Essay Scoring (AES) software is now often used in the classroom [19]. Though it has its constraints, this technology effectively probes and evaluates textual content. Computerized candidate evaluation is more efficient and cost-effective than human evaluation. Professors should focus more on actual teaching and cutting-edge research rather than spending time grading papers [20]. Machines excel at routine work. Students may either use the analysis results to methodically correct their own work (for things like spelling and grammar mistakes) or provide suggestions for further evaluation questions. Students may get inspiration for individual words, phrases, and even whole essays when perusing the study findings. The research findings have already been implemented in the design and analysis of AES systems. In spite of its lack of depth and precision, this study does touch on a number of important topics [21-24].

Contributions of the Study is as follows:

- To improve the effectiveness of foreign literature learning by Chinese students, this study presented Cross-Cultural Teaching Model (CCTM).
- Using the XGBoost prediction model, the perceptions and performance of Chinese students, provided with CCTM -based foreign literature learning, were intelligently predicted.
- To enhance the student evaluation performance of XGBoost model, Flower Pollination Algorithm was employed for XGBoost model optimization.
- The effectiveness of CCTM in foreign literature learning was compared with existing foreign literature teaching models to emphasize the potency of CCTM in foreign literature learning.

II. Related Works

Saeid Ghalehbani's (2022) research looked at how recasts and cues affected the vowel and consonant pronunciation accuracy of English as foreign language (EFL) students. Based on their scores on a pilot version of the KET, 89 primary school students were chosen to participate in the study and were randomly divided into three groups: the control group, the recast group, and the prompt group [25]. Before receiving treatment, each group took a custom-made pronunciation test developed by the researchers and tested with a sample group. The three groups were all given the same quantity of instruction time and content. People in the remake group were given recast, and those in the prompt group were given prompt. Participants in the control group, however, were not given any form of corrective input. Participants were given a post-treatment pronunciation test that was similar to the test they did before treatment. A multivariate analysis of covariance (ANCOVA) was performed. The outcomes showed that both recast and prompt greatly improved EFL students' consonant and vowel pronunciation accuracy. Furthermore, the findings revealed that the impact of recast and prompt on EFL students' consonant and vowel pronunciation accuracy was statistically indistinguishable.

The current research by Wei Zhang (2020) aimed to test how various types of corrective feedback (recasts and clarification requests) can differentially affect the suprasegmental development of English intonation [26]. A total of 102 EFL learners took part in the study. All participants, with the exception of the control group (n=34), got five treatment sessions intended to get them to focus on and practice the target feature in real-world discourse, and their untarget like output was corrected through recasts or clarification requests. Seven intonation features, including words/IP, pause, anacrusis, lengthing, pitch reset, improper tonicity, and tone selection, were analyzed acoustically using pre- and post-test measures of learned and untrained occurrences. The findings indicated that 1) recasts are more effective than clarification requests in helping EFL learners develop English intonation at a suprasegmental level, and that 2) recasts may lead learners to establish, reinforce, and generalize their new phonological knowledge of English intonation that they practiced during the treatments.

It is timely and prudent to provide a scholarly work that focuses on synthesizing and presenting the current state of affairs in the field of corrective feedback (CF) for the development of L2 pronunciation, as interest in this topic has been growing quickly in recent years. Existing descriptive studies show that both teachers and students value the provision of CF as an important part of L2 pronunciation development, particularly when the errors in question impede effective communication. In recent years, researchers in both classroom and laboratory contexts have begun to examine the pedagogical benefits of CF with a specific emphasis on pronunciation through the use of quasi-experimental studies employing a pretest-posttest design.

Even though the findings suggest that pronunciation-focused CF aids in the development of both segmental and suprasegmental accuracy, there appears to be a great deal of individual variability in how well such CF techniques work. [27] The potentials of CF for teaching pronunciation are greatest (a) when L2 learners have sufficient phonetic knowledge, conversational experience, and perceptual awareness of target sounds; (b) when CF provides model pronunciation forms (e.g., recasts rather than prompts); and (c) when the target of instruction concerns communicatively important and salient features.

Nearly half of the world's population uses a mobile phone regularly, a number that has grown by 20% in the past year compared to the last global population survey. Researchers performed a global survey to determine the average monthly data usage of mobile devices, and found that a single user generated 45 exabytes of data in just one month. Nowadays, e-commerce businesses are starting to see data consumption and analytics as one of their most essential requirements. Individuals' future signature or activity can be forecast with the help of such gathered data. The average calculation and quantity of data to be collected for five billion users looks to be much more challenging if 45 terabytes of data can be stored for a single user. It appears that a conventional computer system would have trouble processing this volume of input, even more so than the human working concept. For researchers in the fields of machine learning and artificial intelligence, accumulating enough data to make accurate predictions based on a user's behavior is a crucial first step. The expectations for academic assessment and the responsibilities of both teachers and students are laid out in this essay by Yang Q (2022). Most people were unaware of the online education paradigm even before the pandemic. Since it's difficult to hold traditional, in-person classes, internet education has been slow to catch on [28]. In an effort to increase student-teacher communication and provide more flexible learning options, nearly sixty percent of nations are attempting to transition their education systems to online models. Big data is one of the IT industry's technological revolutions that gained traction after the cloud computing disaster. In this paper, we suggest using a support vector machine (SVM) to analyze ESL pedagogy and then compare it to the more conventional method of fuzzy analysis. According to the findings, the suggested model is 5% more accurate than the current algorithm, at 98%. Farhad Tabandeh's (2019) research compared the effectiveness of two methods for teaching English lax vowels to Persian EFL students: focus-on-form (FonF; explicit instruction followed by focused tasks) and focus-on-forms (FonFS; explicit instruction followed by controlled exercises). Forty-eight students voluntarily participated in a 6-hour course, with 17 receiving FonF, 16 receiving FonFS, and 15 participating in theme-based talks without attention to the target vowels as the experimental group, comparison, and control, respectively. Auditory measurements of tongue position were used to determine the phonetic accuracy of learners' pronunciations elicited in controlled read-aloud and spontaneous image description tasks (i.e., formant 1 [F1] for the height and formant 2 [F2] for the backness of the tongue). While the results showed that both methods of instruction led to substantial gains in phonetic accuracy (i.e., adjusting F1/F2 values) in the controlled task, only the FonF methodology was successful in the spontaneous task, showing particularly large effects in the delayed posttest [29]. Nothing got better for the control group. Results suggest that there may be significant advantages to teaching FonF to EFL students, particularly in terms of helping them acquire more native-like pronunciations in spoken English. Finally, the article discusses how these results can be applied in the classroom.

The purpose of this study by Siti Salina Mustakim was to better understand how teachers of the Contemporary Children's Literature Program at the upper primary level handle the subject. This paper uses classroom observations and interviews as research tools to compare and contrast the methods used by five ESL teachers of Year 5 students and to explore the various difficulties these educators encounter when instructing literature [30]. The preliminary results of the methods indicated that there was little effort made to incorporate literature concepts into the classroom. Educators lacked originality and relied heavily on the CDC's premade lessons. Because the school did not supply the necessary material, the Pre, While, and Post-Reading strategy was not fully implemented. Nonetheless, there is a lot of hope that the initiative will help students improve their language skills in the classroom. By analyzing how educator’s approach teaching and learning in the classroom, this research adds to the literature on authentic education.

The purpose of the study by Ruzbeh Babaee (2014) is to give credence to literature as a significant tool for training rudimentary linguistic abilities like speaking, listening, reading, and writing [31]. In order to make readers aware of the primary reasons why language instructors are suggested to use literary texts in their classrooms, it is important to emphasize the reasons for using literature in language classes and the major factors for choosing appropriate kinds of literary texts in such classes. The benefits of different types of literature to language instruction, as well as some of the challenges that teachers of languages encounter when using literature as a teaching tool, are also taken into account here.

Because of their potential to help students move beyond the referential and into the representational realms of language, reading diaries are commonly used in the instruction of foreign languages. In this article, Ochoa Delarriva (2015) presents findings from a qualitative study of reading logs created by upper-level students in an English language teacher education program in Argentina [32]. The students were required to keep the logs as part of their coursework in a literature course. The Authors were able to reflect on the usefulness and applicability of reading logs and the need to encourage diversity in literature teaching in English language teacher education by categorizing them according to the various literature teaching models used in the program.

III. Methodology

This section discusses in detail about the cross-cultural teaching model of foreign literature under the application of machine learning technology. Fig. 1 depicts the schematic representation of the proposed teaching model. 1307 Chinese students participated in the current research. All of the applicants are sophomores at one of three different Chinese high schools in Lu'an City. They may have been following a standardized curriculum and syllabus set out by the "People's Republic of China Ministry of Education" (Li 2020). Table I displays participant information in detail.

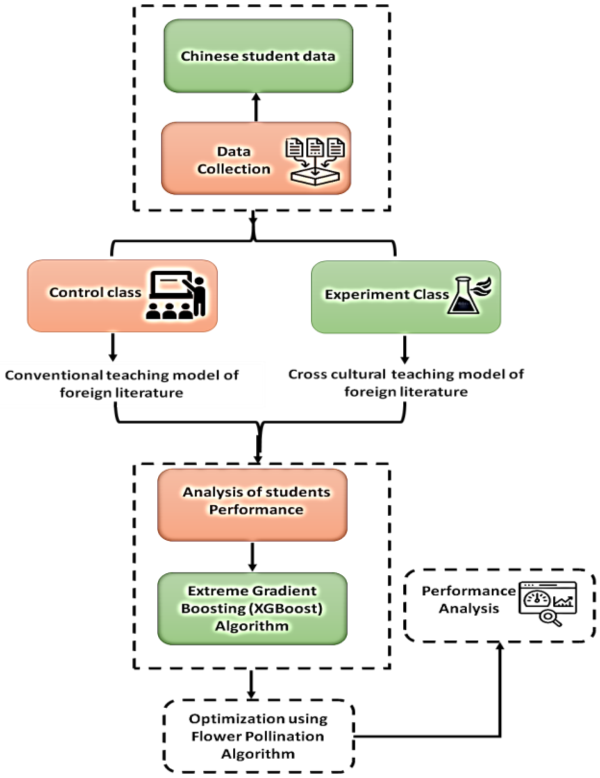

Chinese student data
Data Collection
Control class
Experiment Class
Conventional teaching model of foreign literature
Cross cultural teaching model of foreign literature
Analysis of students Performance
Extreme Gradient Boosting (XGBoost) Algorithm
Performance Analysis
Optimization using Flower Pollination Algorithm

Fig. 1. Schematic representation of the proposed teaching model.

TABLE I. Participants Characteristics

School	Total participants	Male	Female	Mean Age
A	829	399	430	16.92
B	154	94	60	16.64
C	324	179	145	16.72
Total	1307	672	635	16.83

A. *Student Categorization*

Among 1307 participants, 50 percent of the participants i.e. 653 students were in the control group and the remaining students were in the experimental group. CC students follow conventional teaching method of foreign literature whereas EC students follow cross cultural teaching method of foreign literature.

B. *Conventional Teaching Method*

The most prevalent teaching style used in schools all around the globe is teacher-centered instruction and lectures. Teachers that use this style of instruction choose activities and provide resources that are suitable for the students' abilities in foreign literature learning. Here, a curriculum establishes the foreign literature material, and all students study the same subjects at the same time. Only a little amount of information on foreign literature, chosen by the instructor or the school library, is available to students. The majority of the time, academic subjects is discrete and unrelated to one another. These learners recite material exactly and sometimes engage in critical information analysis. A little amount of attention is paid to how to use ideas or facts in conventional teaching model based foreign literature learning. Foreign literature learning takes place in an often-quiet classroom according to conventional teaching (CT) model. This CT model can be efficient particularly for

- Disseminating information that is hard to get elsewhere.
- Quickly presenting the facts.

However, there are various difficulties with this kind of instruction in foreign literature learning, including:

- Not all students learn best by listening.
- Maintaining students' attention is often challenging.
- The strategy typically calls for little to no critical thinking.
- It assumes that all students learn in the same impersonal manner.

C. *Cross Cultural Teaching Method*

Since China joined the "World Trade Organization," there has been an uptick in demand for people with foreign language skills in all industries. At the same time, however, more strict and detailed requirements for English proficiency and diverse abilities have been imposed. As globalization has accelerated in recent years, China and other nations have cooperated and competed more often, and interactions between Chinese civilization and other cultures have been booming [25]. China desperately needs individuals with cross-cultural aptitude given the history of mutual learning between Chinese and other cultures. Most pupils cease to be really interested in conventional education. Therefore, in this study, we suggest a cross-cultural teaching strategy.

There are three definitions for cross-cultural education: first, it relates to the simultaneous teaching of foreign literature based on multiple cultures in the environment where various cultures coexist; second, foreign literature students who have exposed to a specific cultural background are exposed to another cultural environment with varied customs for foreign literature education. Third, it's about making an intentional effort to bring together people from different cultural backgrounds in the classroom so that students of literature from other countries may learn about other cultures' languages, traditions, and values. An understanding of English is frequently necessary when analyzing works written in other languages. From the perspective of a student's capacity for intercultural communication, CCTM-based foreign literature learning seeks to strengthen intercultural communication abilities and encourage both intercultural communication and cultural adaptation, thereby establishing a highly robust intercultural communication theory.

Multiculturalism's protection of the cultural rights of ethnic minorities enables cross-cultural education to embrace and comprehend cultural differences, which is essential for successfully addressing the problem of cultural diversity. The essential notions of intercultural education are understood and seen differently by different academics due to differences in language habits between nations and experts. As a result of its explosive growth, cross-cultural studies is now acknowledged as an interdisciplinary field of applied social science and cultural education research devoted to the examination of the causes and consequences of variation in social structures, institutions, and individuals in response to their context. To face the problems of this age, a new kind of education based on understanding and appreciating other cultures has emerged. Human rights, the end of discrimination, and the cultivation of talent are all areas where intercultural training and civic education overlap. The decline of connectedness and synchronization strategies grounded in local communities all contribute to the pressing need to include people of different ethnicities and backgrounds into the framework of cross-cultural civic community in the postmodern era. International political and economic relations sparked the idea of cross-cultural competency. Thus, intercultural competency studies continue to be conducted with a focus on implementation. As individuals continue to investigate cross-cultural encounters, the foundations of what it means to be culturally competent grow. From this, it develops its own theory of intercultural communication and a model of intercultural proficiency.

Teaching FL with a cross-cultural focus allows institutions to better prepare their students for the global workforce by helping them develop the critical thinking skills necessary to effectively communicate and collaborate across cultural boundaries. However, due to China's late entry into the area, numerous issues remain in the implementation of FL courses at the tertiary level. Teachers that use a speculative approach to FL education must alter their students' conceptual frameworks in order to successfully implement cross-cultural speculative FL instruction. First, the teacher needs to have a deep understanding of the significance of cross-cultural speculative ability, and base their teaching work on this understanding; second, the teacher needs to have a complete understanding and mastery of the connotation of cross-cultural thinking ability, and use this as the starting point and the foothold to do all types of teaching work effectively; third, the teacher needs to have a deep understanding of the significance of cross-cultural sensitivity, and base their teaching work on this understanding; Fourth, educators ought to abandon the notion of conventional opposites in time, stop seeing FL culture through "tinted glasses," assess it objectively, scientifically, and dialectically, and take advantage of what other nations have to offer.

IV. Analysis of Students' Performance

A. *Machine Learning based Extreme Gradient Boosting (Xgboost) Algorithm*

The "Extreme Gradient Boosting Method (XGBoost)" was used to analyze the students' performance in both groups. Given a collection of student information denoted by the notation V={(h_w, t_w)} where h_w represents the performance profile of students and t_w is their associated binary label, Assuming that the XGBoost model is made up of R decision trees, the objective function may be found by using the equation (1):

$$\hat{t}_w = \sum_{r=1}^{R} l_r(h_w), l_r \in L \quad (1)$$

Where L is the space of regression trees, and for each l_r there is a corresponding tree whose performance scores are also stored in L. Equation (2) describes the loss function used to forecast student performance.

$$F(l_r) = \sum f\,(\hat{t}_w, t_w) + \sum \Omega\left(l_y\right) \quad (2)$$

For predicting student performance, the first term is the derivative of a loss function, F, which measure the difference between the real output $\hat{t}_w$ and the anticipated output t_w. The second is a regularisation term, which devalues the prediction's complexity to prevent over-fitting. In the case where Ω and $\hat{t}_w$ can be represented as:

$$\hat{y}_w^{(y)} = \hat{y}_w^{(y-1)} + l_y(h_w) \quad (3)$$

$$\Omega(l) = \gamma Y + \frac{1}{2}\|w\|^2 \quad (4)$$

where, Y represents the total number of leaf nodes and w represents the performance grade. From this, we may derive that

$$F\left(l_y\right) \approx \sum_{g-1}^{Y}(g_w) i_g + \frac{1}{2}\left(\sum_{w \in w_g} x_w + \lambda\right) i_g^2] + \gamma Y (5)$$

where, g_w and x_w represent the first- and second-order gradient statistics of the loss function in predicting student performance, respectively. Parameters γ and λ are constants used to adjust the level of regularisation. They help prevent prediction models from becoming too specific.

B. *Optimization using Flower Pollination Algorithm (FPA)*

The "Flower Pollination Algorithm" (FPA) is used to enhance the performance of XGBoost in making predictions. The inspiration for the invention of it was taken by Xin-She Yang in 2012 from the flow pollination experienced by flowering plants. Multi-objective optimization is now a part of FPA's capabilities. In the sake of simplicity, we will employ the following four rules:

- It has been shown that pollen-carrying insects follow trajectories that are compatible with L'evy flights, suggesting that biotic and cross-pollination may be thought of as a kind of global pollination (Rule 1).

- For local pollination, abiotic pollination and self-pollination are used (Rule 2).
- "Flower constancy," which pollinators like insects may acquire, is equivalent to a reproduction probability that scales with the degree of similarity between two blooms (Rule 3).
- The interaction or switching between local and global pollination may be controlled by a switch probability p [0, 1], slightly slanted toward local pollination (Rule 4).

In order to improve the accuracy of the algorithm's prediction of students' performance, it first generates a random initial population and evaluates it. Before a new student's performance can be calculated, the pollination type must be established using a fixed probability p (K4). That is, we select a random number k between 0 and 1, and if k is less than u, then both global pollination and flower constancy (K1 and K3) may occur as follows:

$$h_w^{y+1} = h_w^y + \gamma F\left(h_w^y - gbest\right) \qquad (6)$$

Where student w's performance at time y is denoted by h_w^y, the current best performance is denoted by gbest, is a scaling factor, and F is a step size derived from Lévy flight, as follows:

$$F(a,s) \sim \frac{\lambda\Gamma(\lambda)sin\left(\frac{\pi\lambda}{2}\right)}{\pi}\frac{s}{a^{1+\lambda}}, |a| \to \infty \quad (7)$$

where () is the standard gamma distribution with index, s is a variable used to adjust the distribution's tail amplitude (s=1 in the proposed FPA), and a big step function (a>>a0>0) is built using the necessary nonlinear transformation as:

$$a = \frac{P}{|D|^{\lambda-1}} \qquad (8)$$

When P and D are two random student samples chosen from a Gaussian normal distribution with mean equals 0, and standard deviations σ_P and σ_d:

$$P \sim \left(0, \sigma_p^2\right), D \sim (0, \sigma_d^2) \qquad (9)$$

$$\sigma_P = \left[\frac{\Gamma(1+\lambda)}{\lambda\Gamma((1+\lambda)/2)} \cdot \frac{sin\left(\frac{\pi\lambda}{2}\right)}{2^{(\lambda-1)/2}}\right], \sigma_d = \qquad (10)$$

Since σ_P and σ_d cannot be selected individually for any value of, the standard deviation σ_d is fixed at 1. If, however, k is larger than u, then the local pollination and flower constancy (K2 and K3) are carried out as follows:

$$h_w^{y+1} = h_w^y + \varepsilon\left(h_g^y - h_r^y\right) \quad (11)$$

where h_w^y and h_r^y are two student results chosen at random and $\varepsilon \in [0,1]$ is a random integer. Next, the current best performance is updated, and the search iterations are restarted until the termination criterion is met.

V. Results and Discussions

The investigation of student's performance and perceptions regarding cross-cultural teaching model (CCTM) based foreign literature learning is conducted using machine learning algorithm. The learning performance and perceptions of students being exposed to the suggested CCTM was evaluated using XGBoost prediction model optimized by FPA (XGBoost-FPA). To state the effectiveness of CCTM based foreign literature learning, we compared the suggested model with existing teaching models in foreign literature learning. The existing teaching models in foreign literature learning considered for the comparative investigation are Conventional Teaching (CT) model, Zoom Web Conference System (Zoom WCS) [33], Translanguaging + Task-based Language Teaching (TL+TBLT) model [34], Content and language integrated learning (CLIL) model [35], Flipped Learning (FL) model [36]. This study aimed at exploring the Chinese students' perceptions of using cross-cultural Teaching model in foreign literature learning in terms of Perceived Usefulness, Perceived Ease of Use, and student acceptance [37]. In addition, the learning performance of students being exposed to the suggested CCTM was evaluated in terms of student's overall performance, learning ability, and student engagement.

A. *Perceived Usefulness*

The degree to which a student thinks that utilising CCTM for studying foreign literature would improve their academic performance is known as student's perceived usefulness. Fig. 2 shows the comparative investigation of student's perceived usefulness for various teaching models in foreign literature learning. It is noted that student's perceived usefulness for CCTM based foreign literature learning was higher than that for existing teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. Higher student's perceived usefulness, observed for CCTM, using XGBoost-FPA illustrates that CCTM is highly useful for Chinese students who are learning foreign literature.

B. *Perceived Ease of Use*

One of the most important variables in determining whether students would adopt CCTM-based foreign literature learning is student's perceived ease of use. It is defined as how simple it is for students to access CCTM-based instruction in foreign literature. Fig. 3 shows the comparative investigation of student's perceived ease of use for various teaching models in foreign literature learning. From Fig. 3 and Table II, it is observed that student's perceived ease of use for CCTM based foreign literature learning (predicted using XGBoost-FPA) was higher than that for existing teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. This indicates that CCTM is easily accessible by Chinese students for learning foreign literature.

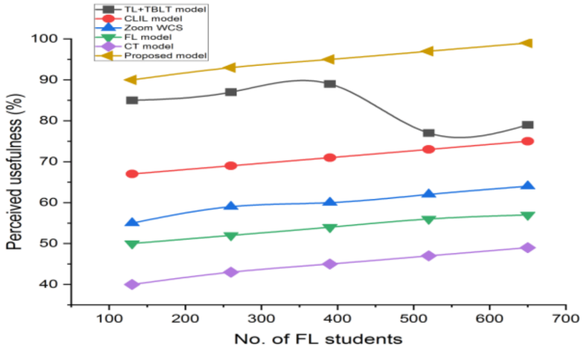


Fig. 2. Analysis of student's perceived usefulness for various foreign literature teaching models.

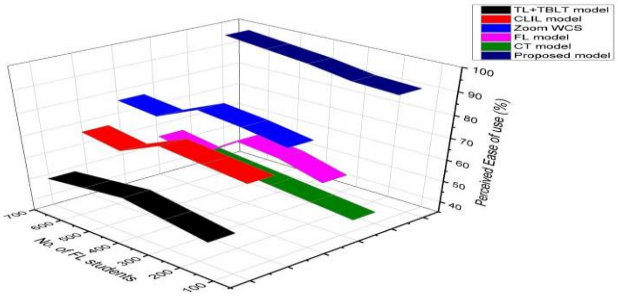


Fig. 3. Analysis of student’s perceived ease of use for various foreign literature teaching models.

TABLE II. COMPARATIVE EVALUATION OF STUDENT’S PERCEPTIONS REGARDING VARIOUS FOREIGN LITERATURE TEACHING MODELS

Student’s Perception Variables	TL+TBLT model	CLIL model	Zoom WCS	FL model	CT model (Control Group)	CCTM (Proposed model)
Perceived Usefulness (%)	79	75	64	57	49	99
Perceived Ease of Use (%)	52	69	79	58	45	97
Student’s Acceptance (%)	69	79	59	57	48	97

C. Student’s Acceptance

Student’s acceptance is defined as the degree to which a student is satisfied with the CCTM based foreign literature learning. Fig. 4 shows the comparative investigation of student’s acceptance for various teaching models in foreign literature learning. Chinese students receiving CCTM based foreign literature learning reported to have higher acceptance level compared to that for existing teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. The findings depict that Chinese students are highly satisfied with CCTM based foreign literature learning.

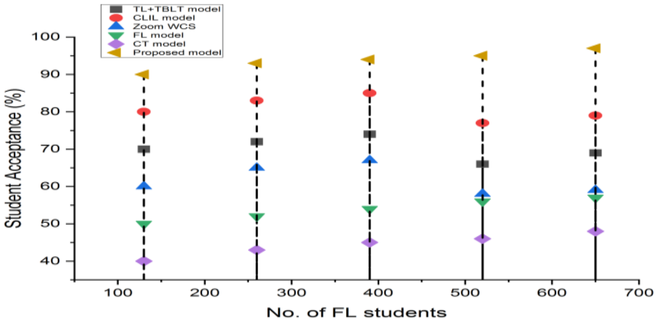


Fig. 4. Analysis of student’s acceptance for various foreign literature teaching models.

D. Student’s Academic Performance

Student’s academic performance means the extent to which a student has attained knowledge on foreign literature through CCTM. Fig. 5 shows the comparative investigation of student’s academic performance in various teaching models in foreign literature learning. It is noted from Table III that academic performance of students receiving CCTM based foreign literature learning was higher than that for existing teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. Higher student’s academic performance observed for CCTM (as predicted by XGBoost-FPA) shows that CCTM enhances the efficiency of foreign literature education.

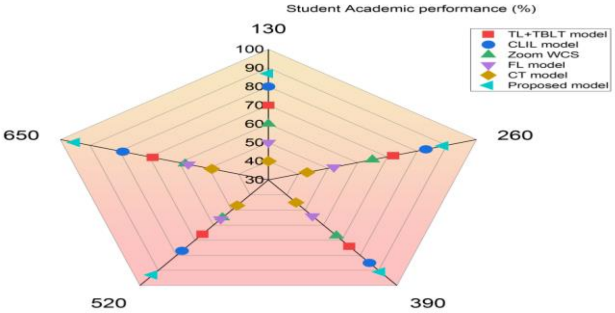


Fig. 5. Analysis of student’s academic performance in various foreign literature teaching models.

TABLE III. COMPARATIVE EVALUATION OF STUDENT’S PERFORMANCE REGARDING VARIOUS FOREIGN LITERATURE TEACHING MODELS

Student’s Performance Variables	TL+TBLT model	CLIL model	Zoom WCS	FL model	CT model (Control Group)	CCTM (Proposed model)
Student’s Academic Performance (%)	69	79	58	57	49	95
Student’s Engagement (%)	85	74	64	53	39	96
Student’s Learning Ability (%)	69	79	54	64	46	98

E. Student’s Engagement

The term "student engagement" refers to the extent to which a student participates in class discussions and other collaborative tasks. Fig. 6 shows the comparative investigation of student’s engagement in various teaching models in foreign literature learning. It is noted that student’s engagement in CCTM based foreign literature learning (predicted by XGBoost-FPA) was higher than that for existing teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. The results showed that Chinese students are highly engaged in CCTM based foreign literature classroom.

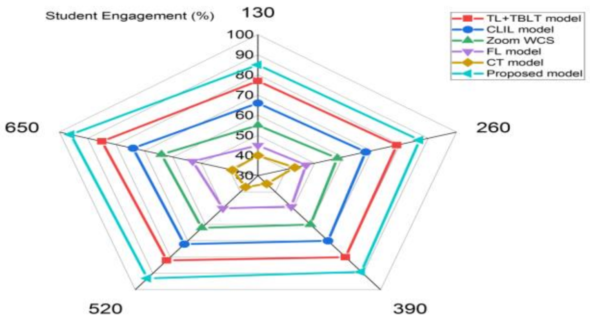


Fig. 6. Analysis of student's engagement in various foreign literature teaching models.

F. *Student's Learning Ability*

Student's learning ability is defined as the student's capacity to comprehend and understand foreign literature. Fig. 7 indicates the comparative investigation of student's learning ability in various teaching models in foreign literature learning. It is noted that student's learning ability in CCTM based foreign literature learning (predicted by XGBoost-FPA) was higher than that for existing teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. This proved that CCTM based foreign literature learning improves the ability of Chinese students to learn foreign literature.

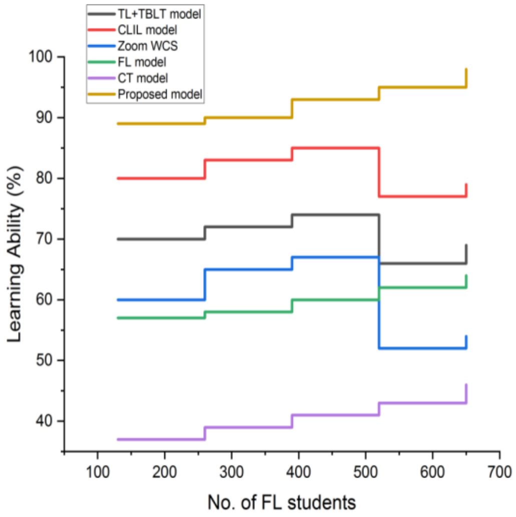


Fig. 7. Analysis of student's learning ability in various foreign literature teaching models.

VI. Conclusion

This research presents the cross-cultural teaching model (CCTM) for foreign literature learning by Chinese students. The performance and perceptions of Chinese students regarding CCTM based foreign literature learning were automatically estimated by XGBoost prediction model which was optimized by FPA (XGBoost-FPA). The effectiveness of CCTM in foreign literature learning was confirmed by comparing with existing foreign literature teaching models like CT model, TL+TBLT model, CLIL model, Zoom WCS, and FL model. The findings indicated that students exposed to CCTM based foreign literature learning are reported to have acceptance level of 97%, learning ability of 98%, academic performance of 95%, engagement level of 96%. Moreover, student's perceived usefulness and perceived ease of use for CCTM based foreign literature learning were found to be 99% and 97%. Chinese students are highly satisfied with the suggested CCTM based foreign literature learning. These results confirm that CCTM can improve student's foreign literature learning ability, make them involved in foreign literature learning. Future research may resolve the shortcomings of this study. First of all, the study's student sample size is small, which might restrict how broadly the findings can be applied to ensure the effectiveness of CCTM. Future research with a bigger sample size could provide results that add to the CCTM's potency in foreign literature learning.

References

[1] M. Polat, "Teachers' attitudes towards teaching English grammar: a scale development study," International Journal of Instruction, vol. 10, no. 4, pp. 379–398, 2017.

[2] O. Aljohani, "Does teaching English in Saudi primary schools affect students' academic achievement in Arabic subjects?" Advances in Language and Literary Studies, vol. 7, no. 1, pp. 214–225, 2015.

[3] Philominraj, D. Jeyabalan, and C. Vidal-Silva, "Visual learning: a learner centered approach to enhance English language teaching," English Language Teaching, vol. 10, no. 3, p. 54, 2017.

[4] R. Kabooha and T. Elyas, "(e effects of YouTube in mul- timedia instruction for vocabulary learning: perceptions of EFL students and teachers," English Language Teaching, vol. 11, no. 2, p. 72, 2018.

[5] Muslem and M. Abbas, "(e effectiveness of immersive multimedia learning with peer support on English speaking and reading aloud," International Journal of Instruction, vol. 10, no. 1, pp. 203–218, 2017.

[6] L. M. D. A. Santos, M. S. E. Kadri, R. Gamero, and T. Gimenez, "Teaching English as an additional language for social participation: digital technology in an immersion programme," Revista Brasileira de Lingu´ıstica Aplicada, vol. 18, no. 1, pp. 29–55, 2018.

[7] Zou, "Research on college English teaching model based on multimedia and network," DEStech Transactions on Social Science Education and Human Science, pp. 19–21, icsste, 2017.

[8] R. Gooch, K. Saito, and R. Lyster, "Effects of recasts and prompts on L2 pronunciation development: teaching English/ ɹ/to Korean adult EFL learners," System, vol. 60, no. 60, pp. 117–127, 2016.

[9] Finger, G. Creativity, visualization, collaboration and communication. In M. Henderson & G. Romeo (Eds.), "Teaching and digital technologies: Big issues and critical questions" (pp. 89–103). Cambridge University Press, 2015.

[10] Finger, G., Romeo, G., Lloyd, M., Heck, D., Sweeney, T., Albion, P., & Jamieson-Proctor, R. (2015). "Developing graduate TPACK capabilities in initial teacher education programs: Insights from the Teaching Teachers for the Future Project," The Asia-Pacific Education Researcher, 24(3), 505–513.

[11] D.Forbes and K.Rinehart, "Digital learning: Critical perspectives and lifelong possibilities," In M. Hill & M. Thrupp (Eds.), The professional practice of teaching in New Zealand. South Melbourne, VIC, Australia: Cengage, 2019.

[12] G.Gay, "Culturally responsive teaching: Theory, research and practice," New York, NY: Teachers College Press, (3rd ed.) 2018.

[13] M. Gui, M. Fasoli and R.Carradore, "Digital well-being developing a new theoretical tool for media literacy research," Italian Journal of Sociology of Education, vol.9, no.1, pp.155–173, 2017.

[14] T. Hadziristic, "The state of digital literacy in Canada: A literature review. Toronto, Canada: Brookfield Institute for Innovation Entrepreneurship," Google Scholar Harasim, L. (2012). Learning theory and online technologies. New York, NY: Routledge, 2017.

[15] L.Ilomäki, S.Paavola, M.Lakkala and A.Kantosalo, "Digital competence–an emergent boundary concept for policy and educational

research," Education and Information Technologies, vol.21, no.3, pp.655–679, 2016.

[16] M.Judge, "A case study analysis of Introducing iPads in a Portugese School under The Erasmus+ Micool Project," In Paper presented at the International Association for Development of the Information Society (IADIS) International Conference on Educational Technologies, Sydney, Australia (pp. 1–8). ICEduTech, 2017.

[17] S. Price and P. A. Flach, "Computational support for academic peer review: a perspective from artificial intelligence," Communications of the ACM, vol. 60, no. 3, pp. 70–79, 2017.

[18] J. Lemley, S. Bazrafkan, and P. Corcoran, "Deep learning for consumer devices and services: pushing the limits for machine learning, artificial intelligence, and computer vision," IEEE Consumer Electronics Magazine, vol. 6, no. 2, pp. 48–56, 2017.

[19] J. P. Davis and W. A. Price, "Deep learning for teaching university physics to computers," American Journal of Physics, vol. 85, no. 4, pp. 311-312, 2017.

[20] K. Gamage, "The pedagogical application of the grammar translation method as an effective instructional methodology in teaching English as a second language," Open Access Library Journal, vol. 7, no. e6913, pp. 1–10, 2020.

[21] M. Ben El Moudden, "The integration of games in teaching English as a foreign language in the classroom: moulay ismail university as a case study," International Journal of Language and Literary Studies, vol. 3, no. 1, pp. 208–229, 2021.

[22] N. Sidash, M. Roganova, V. Domina, L. Victorova, M. Roganov, and V. Miroshnichenko, "Pedagogical consciousness formation of future university educators in the process of teaching English," Universal Journal of Educational Research, vol. 8, no. 4, pp. 1202–1211, 2020.

[23] L. R. Sharma and R. N. Khanal, "Effectiveness of lecture and discussion teaching methods in teaching English language through figures of speech," Journal of Advanced Academic Research, vol. 6, no. 1, pp. 1–17, 2019.

[24] Y. Alhasov, A. Verbytska, and T. Kolenichenko, "Teaching English to adult learners within extracurricular activities at university: barriers and motivation factors," Advanced Education, vol. 7, no. 15, pp. 12–19, 2020.

[25] Saeid Ghalehbani, "The Comparative Effect of Recasts and Prompts on EFL Learners' Vowel and Consonant Accuracy," Language Teaching Research Quarterly, vol. 28, pp. 31–52, 2022.

[26] Wei Zhang, "Effects of recasts, clarification requests on suprasegment development of English intonation," Porta Linguarum, vol.35, enero, pp. 311-325, 2021

[27] Saito, K. (in press). Corrective feedback and the development of L2 pronunciation. In H. Nassaji & E. Kartchava (Eds.), The Cambridge handbook of corrective feedback in language learning and teaching. Cambridge, UK: Cambridge University Press,2019.

[28] Yang Q. Analysis of English Cultural Teaching Model Based on Machine Learning. Comput Intell Neurosci. Vol.2022, pp.7126758, 2022.

[29] Farhad Tabandeh et al., "Differential Effects of FonF and FonFS on Learning English Lax Vowels in an EFL Context," The Journal of AsiaTEFL, vol.16, no.2, pp.499-515, 2019.

[30] Siti Salina Mustakim, "Teacher's Approaches in Teaching Literature: Observations of ESL Classroom," Malaysian Online Journal of Educational Science, vol. 2, no. 4,2014.

[31] Ruzbeh Babaee, "Significance of Literature in Foreign Language Teaching," International Education Studies, vol. 7, no. 4, 2014.

[32] O.Ochoa Delarriva, E. A. Basabe, "Reading logs and literature teaching models in English language teacher education," HOW, vol.22, no.2, pp.37-53, 2015.

[33] C.Li, "A positive psychology perspective on Chinese EFL students' trait emotional intelligence, foreign language enjoyment and EFL learning achievement," Journal of Multilingual and Multicultural Development, vol.41, no.3, pp.246-263, 2020.

[34] G.Bui, and K.W.Tai, "Revisiting functional adequacy and task-based language teaching in the GBA: insights from translanguaging," Asian-Pacific Journal of Second and Foreign Language Education, vol.7, no.1, pp.1-14, 2022.

[35] H.A. Alfadda, and H.S.Mahdi, "Measuring students' use of zoom application in language course based on the technology acceptance model (TAM)," Journal of Psycholinguistic Research, vol.50, no.4, pp.883-900, 2021.

[36] L.Van Mensel, P.Hiligsmann, L.Mettewie, and B.Galand, "CLIL, an elitist language learning approach? A background analysis of English and Dutch CLIL pupils in French-speaking Belgium," Language, Culture and Curriculum, vol.33, no.1, pp.1-14, 2020.

[37] G.Lee, and A.Wallace, "Flipped learning in the English as a foreign language classroom: Outcomes and perceptions," TESOL quarterly, vol.52, no.1, pp.62-84, 2018.

Meta Heuristic Fusion Model for Classification with Modified U-Net-based Segmentation

Sri Laxmi Kuna[1], Dr. A.V. Krishna Prasad[2], Suneetha Bulla[3]
Research Scholar, Department of CSE, Koneru Lakshmaiah Educational Foundation, Amaravathi, India[1]
Research Supervisor, Department of CSE, Koneru Lakshmaiah Educational Foundation, Amaravathi, India[2, 3]

***Abstract*—General cause of diabetes mellitus is Diabetic Retinopathy (DR), which outcomes in lesions on the retinas that impair vision. If it is not detected in time, the result is severe blindness issues. Regrettably, there is no treatment for DR. Early diagnosis and treatment of DR can greatly lower the risk of visual loss. In contrast to computer-aided diagnosis methods, the manual diagnosis of DR using retina fundus images is more time-consuming effort, and high cost as well, as it is highly prone to error. Deep learning has emerged as one of the most popular methods for improving performance, particularly in the classification and analysis of medical images. Therefore, a deep structure-based DR detection and severity classification has been demonstrated for treating the DR with the usage of fundus images. The major aim of this developed technique is to classify the severity level of the retinal region of the human eye from the fundus images. At first, the required retinal fundus images are collected from the standard benchmark data sources. Secondly, image enhancement techniques are applied to the collected fundus images to improve the quality of images. Thirdly, the abnormality segmentations are carried out by using the optic disc removal process using active contouring model and then, the regional segmentation is done via the Modified U-Net method. Finally, the segmented image is subjected to the hybrid classifier network named a Hybrid Soft Attention-based DenseNet with Multi-Scale Gated ResNet (HSADMGR Net) for classifying the retinal fundus images and finding the severity level of the retinal images with higher accuracy. Furthermore, the parameters present inside the hybrid classifier network are optimized with the help of implemented Multi-Armed Bandits Groundwater Flow Algorithm (MABGFA). The test results regarding the developed deep structure-based DR model are validated with the existing DR detection and classification approaches by using different performance measures.**

Keywords*—Diabetic retinopathy segmentation and classification model; multi-armed bandits groundwater flow algorithm; hybrid soft attention-based DenseNet with multi-scale gated ResNet; modified U-Net-based segmentation

I. INTRODUCTION

One of the major issues facing people nowadays is diabetes. DR is one of the blindness diseases and it is complicated by diabetes. It has varying degrees of severity and it is hard to predict the DR in its early condition [9]. The retina, which converts light into the electric signal that is processed to produce an image, is impacted by DR. A network of abnormalities found in the retina gives nutrients to the retina. Abnormalities are damaged as a result of diabetes [10]. At that time, the retina does not contain a blood supply. This has an impact on retinal health and ultimately skews a person's vision [11]. Background retinopathy refers to the initial DR stage. Diabetes affects the blood vessels in the early stage, which causes sight loss [12]. It's possible for the vessels to swell slightly and leak blood, fluid, or proteins. Ophthalmologists or skilled graders manually identify DR, which is costly [13]. Lesions caused by diabetic retinopathy are typically believed to be reversible and the illness can only grow more slowly [14].

Convolutional methods frequently use a DL architecture with millions of learnable parameters [15]. Train a large amount of data suffers from the over fitting issue. Although, very effective laser therapy does not completely cure vision impairment and highlights the necessity of an efficient screening model [16]. Despite this fact, the screening of diabetic retinopathy is successful. An unacceptable amount of diabetes patients do not have the annual eye exams that are advised by guidelines [17]. One of the most significant issues faced by ophthalmologists and diabetes providers today is identifying at-risk patients and enabling them to get therapeutic therapies in a timely manner [18]. DR continues to be the cause of acquired vision loss despite the availability of healthcare facilities and the development of several methods that improve early detection. Maintaining such a register will probably be difficult because of the difficulty in obtaining precise [19], frequently updated lists from primary care practitioners. Due to these difficulties as well as the anticipated size and breadth of such a project, federal funding would probably be needed and would be expensive [20] for the construction of a national diabetes registry that allows for broad population coverage. The long-term advantages must be underlined, however, as they are of the opinion that such a register would significantly lower the rate of blindness among diabetes patients [21].

Due to their capacity to extract and learn the most discriminative features at the pixel level, convolutional neural networks (CNNs), a subset of deep learning, produced effective deep models for DR grading [22,23]. In this study, we create a method for the automatic segmentation and classification of retinal fundus picture using a convolutional neural network based on the U-Net architecture. We use data sets with very high resolution images labeled pixel by pixel to train neural networks.

The performance of a model is strongly influenced by how it is optimized using a small set of fundus images [25].

The deep learning-based DR segmentation and classification system objectives are described below.

- To design a deep learning-based DR segmentation and classification system that effectively detects the DR disease at an early stage that is helpful for DR patients in the hospitals to prevent blindness effectively.
- To develop an effective Modified U-Net-based abnormality segmentation to effectively segment the DR disease abnormalities from the fundus images.
- To implement a MABGFA optimization to optimize the parameters from HSADMGR Net to maximize the classification performance of the suggested model in terms of accuracy and precision.
- To develop an effective HSADMGR Net-based classification with optimized parameters like resblock, hidden neuron count, activation function and to maximize precision and accuracy.
- To validate the performance of the developed deep learning-based DR segmentation and classification model with different performance metrics among various optimization strategies and baseline models.

The suggested deep learning-based DR segmentation and classification system is explained in the remaining sections. The explanation of conventional DR segmentation and classification algorithms and methods and their advantages and disadvantages are given in Section II. The research problem is stated in Section II. The developed DR segmentation and classification model developed algorithm, and dataset details are given in Section IV. The segmentation of the DR disease and the preprocessing explanations are given in Section V. Section VI explains the DR classification and the developed model's objective functions. The experimental setup and results are given in Section VII and Section VIII. The summarization of the conclusion is given in Section IX.

II. Literature Survey

In 2021 Saeed *et al.* [1] have proposed a new DR screening system using effective deep learning with fundus images. The low-to-high-level characteristics was taught through a deep learning model. Initially, the first layer of the deep learning model was re-initialized utilizing regions of lesions derived from fundus images. To extract discriminatory characteristics from the input of fundus images in an unsupervised way, they replaced the full layers that encoded features that were global in domain-specific. The model was adjusted, so the lower layers were easy to find the normal regions and local, regional features of the lesion. It reduces the over-fitting problem. A classification layer based on gradient boosting was added last. The 10-fold cross-analysis used to evaluate the suggested model on two difficult datasets showed higher performance than conventional techniques. It helped graders make quick decisions regarding to a therapist for therapy during the early screening of DR patients.

In 2020 Shankar *et al.* [2] have investigated a new DR classification and automatic detection system using a deep learning framework. The contrast level of the fundus image was enhanced during the pre processing stage by utilizing the new method. The required features from the segmented image were then extracted using the developed model and the classification was done by Multilayer Perceptron (MLP) model. To ensure the accuracy of the investigated technique, a number of experiments were conducted using the MESSIDOR DR Dataset. The results of these experiments showed the superiority of the suggested system to the conventional techniques.

In 2020 Qiao *et al.* [3] have designed a new hybrid CNN-based DR segmentation system. Using deep learning techniques that were accelerated by Graphics Processing Units (GPUs), it was possible to identify the DR disease in fundus images and segments them with high performance and little latency. It offered a computerized technique to help ophthalmologists classify the input of fundus images like severe, early, or moderate NPDR. A deep learning technique suggested for early diagnosis of DR is also capable of effectively segmenting the DR disease. Also, it showed high prediction accuracy.

In 2023 Wang *et al.* [4] have implemented a method for segmenting four different types of fundus lesions associated with DR simultaneously using Deep Convolution Neural Networks (DCNN) that was totally autonomous. They suggested a collaborative design consisting of a local branch and a contextual branch to take advantage of multi-scale image information. To efficiently and completely merge informative features from the two branches, new mechanism was suggested to fuse the maps. To further increase the accuracy of the lesion segmentation model, it needed to reduce the over fitting issue in the developed model. Three publicly available fundus datasets were used in extensive experiments, and their methodology yielded a mean AUC value for each dataset separately. The outcomes of the suggested model showed greater performance when compared to other exceeding competing strategies and other cutting-edge models.

In 2023 Zhang *et al.* [5] have suggested a new DR identification and classification using a deep learning approach by examining the correlations between unlabeled data. It was possible to see that the ability of the network to extract crucial features could be improved by a mechanism that fitted the information. Unlabeled data was just potentially valuable labeled data, and the samples could increase the system's effectiveness. In order to effectively use the vast amounts of unlabeled information for precise DR classification, they offered a new framework. They specifically suggested an attention approach to remove the incorrect regions and retrieve factors from various views of input images in the developed model. Additionally, depending on the chosen annotated samples, they developed model compared to execute unknown samples to enhance the possibility of unlabeled information. For two open datasets, the experimental results showed that their new framework achieved high effectiveness.

In 2019 Jebaseeli *et al.* [6] have investigated a new framework for retinal disease segmentation using a neural network. That framework recommended the neural network model for automatically creating vectors, deep learning was proposed for classification, and pre processing method was used to remove the unwanted details. The Firefly method was used to adjust the DLBSVM parameters. The suggested methods were evaluated using the DRIONS, DRIVE, STARE, REVIEW, and HRF fundus imaging datasets. The outcomes showed that the proposed methods given enhanced segmentation due to sensitivity, specificity and accuracy.

In 2022 Kushnure *et al.* [7] have implemented a new architecture for classifying the DR disease using the U-Net approach. It changed the skip pathways utilizing feature fusion mechanisms and local feature reconstruction that represent the specific information in features. To represent the rich spatial information in the low-level features, they used the new module in the bottleneck layer to represent multi-scale features with different receptive fields. These modifications improved segmentation efficiency while decreasing the complex problem of model compared to existing techniques. Furthermore, the independent results confirmed the investigated system's robustness.

In 2018 Mansour *et al.* [8] have investigated an effective DR disease prediction using neural network-based hybrid optimization. That was done in consideration of deep learning to solve very sophisticated segmentation issues. Pre-processing, Principle Component Analysis (PCA) connected component analysis-based localization, adaptive learning-based region segmentation, a high feature extraction, and Linear Discriminate and feature selection were included in the DR system. The proposed DR model performed better with LDA feature selection than with FC7 features or PCA, according to simulation results using common KAGGLE fundus datasets. It also performed more accurately among FC7 features than with LDA features. A comparison with the SIFT technique-based DR feature extraction demonstrated that deep learning-based DR works better than SIFT-based DR.

III. Problem Statement

Most of the conventional diabetic retinopathy classification and segmentation models do not detect the disease in its early stage. It takes more a lot of time while predict the DR disease and it provides more economic cost. Hence, it is unaffordable to many researchers. To overcome these problems, most existing approaches have been designed for the DR segmentation system and advantages and disadvantages are shown in Table I. CNN [1] provides high classification accuracy. Also, it reduces computational complexity. But, it is hard to segment the retinopathy disease because of the complicated structure, fewer voxel intensity variation and highly variable shape and also, it increases the economic cost. SVM [2] gives robustness. Also, it improves the accuracy of lesion segmentation. Yet, it takes a lot of time to classify the disease of DR and also it suffers from Class imbalance during the training. DCNN [3] avoids over fitting problems. Also, it maximizes specificity, accuracy and sensitivity. Yet, it is hard to predict the various lesions regardless of their complex texture, form and small scale and also, it gives very less quality input images and poor performance in the final outcome. DCNN [4] gives strong efficacy. Also, it avoids vision deterioration. But, it is hard to detect the dark spots in retinal images at the early stage and also, and it gives less classification accuracy. MASL [5] reduces the model complexity. Also, it gives more reliable results. Yet, it does not find the local structures of the lesion and also, and it needs a huge amount of data for effective segmentation. TPCNN [6] effectively reduces the cataract progression in the early stage. Also, it reduces the risk of vision loss. Yet, it does not generate the vectors to predict the DR disease and also, and it takes more time. UNet-CNN [7] effectively predicts the dark lesions also and it effectively classifies the DR disease. But, it is difficult to predict the fine vascular structures and causes of skin infections. CNN-DNN [8] effectively solves the intricate segmentation issue. Also, it effectively predicts the DR disease in its early stage. But, it is highly time-consuming while predicting the DR disease and also, and it suffers from over fitting issues. Hence, this disadvantage arises to design an efficient diabetic retinopathy classification and segmentation system with advanced deep-learning techniques.

TABLE I. FEATURES AND CHALLENGES OF EXISTING DR CLASSIFICATION MODELS UTILIZING DEEP LEARNING

Author [citation]	Methodology	Advantages	Disadvantages
Saeed et al. [1]	CNN	It provides high classification accuracy. It decreases computational complex issue.	It is hard to segment the retinopathy disease because of the complicated structure, fewer voxel intensity variations and highly variable shape. It increases the economic cost.
Shankar et al. [2]	SVM	It gives robustness. It improves the accuracy of lesion segmentation.	It takes a lot of time to classify the disease of DR. It suffers from Class imbalance during the training.
Qiao et al. [3]	DCNN	It avoids over fitting issues. It maximizes specificity, sensitivity and accuracy.	It is hard to predict the various lesions regardless of their complex texture, form and small scale. It gives very less contrast images and poor performance in the final outcome.
Wang et al. [4]	DCNN	It gives strong efficacy. It avoids vision deterioration.	It is hard to detect the dark spots in retinal images at the early stage. It gives less classification accuracy.
Zhang et al. [5]	MASL	It reduces the model's complexity. It gives more reliable results.	It does not find the local structures of the lesion. It needs a huge amount of data for effective segmentation.
Jebaseeli et al. [6]	TPCNN	It effectively reduces the cataract progression in the early stage. It reduces the risk of vision loss.	It does not generate the feature vectors at low level to detect the DR disease. It takes more time
Kushnure et al. [7]	UNet-CNN	It effectively predicts dark lesions also. It effectively classifies the DR disease.	It is difficult to predict fine vascular structures. It causes skin infections.
Mansour et al. [8]	CNN-DNN	It effectively solves the intricate segmentation issue. It effectively predicts the DR disease in its early stage.	It is highly time-consuming while predict DR disease. It suffers from the over fitting issue.

IV. Implementation of Diabetic Retinopathy Classification Framework using Hybrid Deep Learning Network

A. Proposed Diabetic Retinopathy Classification Framework

The main challenges with the DR detection models are it does not detect small-scale lesions, low-contrast images, and ambiguous boundaries. Successful segmentation also requires color information because lesions and healthy tissues can seem to be the same when they are of the same color. In the conventional models, they are suffered from the above segmentation issue. Low sensitivity is another problem with the previous segmentation models. The early detection of DR is still difficult, though, for a number of reasons. The current DR screening procedure takes more time. No baseline model used today produces findings that are satisfactory outcomes, because they are frequently very small micro aneurysms and they are susceptible to misdiagnosis or being mistakenly categorized as hemorrhages. For a significant portion of the retinal image collection, some of them were difficult to detect. Because of their resemblance in the color of blood arteries and their proximity to them in some research studies, micro aneurysms are challenging to segment. The abnormality intersection characteristic has been employed in some approaches to derive the optic disc. However, the interference from the blood vessels, they use the entire abnormality structure, which may produce inaccurate or inconclusive results. As a result, the majorities of studies using various image processing methods have not reached a high level of accuracy and fail to categorize the DR disease. The investigated deep learning-based disease of DR segmentation and classification model is shown in Fig. 1.

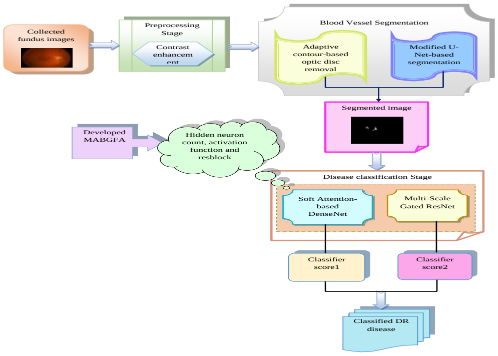

Collected fundus images
Preprocessing Stage
Contrast enhancem ent
Blood Vessel Segmentation
Adaptive contour-based optic disc removal
Modified U-Net-based segmentation
Segmented image
Developed MABGFA
Hidden neuron count, activation function and resblock
Disease classification Stage
Soft Attention-based DenseNet
Multi-Scale Gated ResNet
Classifier score1
Classifier score2
Classified DR disease

Fig. 1. Proposed deep learning-based DR segmentation and classification model.

The newly designed DR segmentation and classification model using new technique to predict the disease in its early stage. It helps in hospitals to prevent vision loss or blindness for patients. It effectively predicts the retinal thin outer structure. The retinal fundus images are collected using the internet resources. The collected fundus images are given to the next stage. Here, the pre processing can be done utilizing a contrast enhancement approach [36]. The pre-processed retinal images are given to the segmentation section. Here, the two types of segmentation process such as abnormality segmentation and regional segmentation can be segmenting the pre processed images. The abnormality segmentation can be performed using Adaptive contour-based optic disc removal process [24]. The regional segmentation can be performed using Modified U-Net method technique. After, the segmentation the images are undergoes the next stage. The classification can be done using HSADMGR Net technique. Here, the investigated MABGFA optimization used for optimizing the parameters that is activation function, hidden neuron count and resblock to maximize the precision and accuracy. The designed HSADMGR Net-based DR

Segmentation and classification system performance is validated to other traditional methods and heuristic algorithms in terms of a number of performance measures.

B. Dataset Details

The collected fundus images are gathered through the benchmark dataset using the below link "https://www.it.lut.fi/project/imageret/diaretdb1/. This dataset contains normal and diseased retinal images. It also contains 89 types of color fundus images. Here, the fundus images are gathered using 50 degree view of workable camera. This dataset give a better accurate solution during DR detection. The model's primary goal was to clearly describe a database and a testing procedure that can be used to compare various DR detection techniques. Hence, the collected retinal fundus images are denoted by A_q^{Np}, where $q = 1,2,\cdots,Q$. The total numbers of sample images are indicated by N. The collected fundus images adopted for the DR segmentation and classification system are shown in Fig. 2.

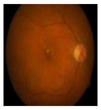
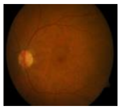
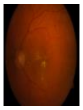
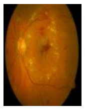
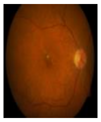
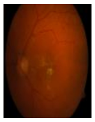
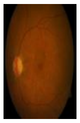
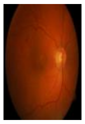
Dataset specificatio n	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Normal retinal images					
Diseased retinal images					

Fig. 2. Gathered fundus images from the dataset for the segmentation and classification of DR diseases.

C. Proposed MABGFA

The developed MABGFA optimization is adopted to improve the effectiveness of the deep learning-based DR classification and segmentation model with the help of optimized the parameters such as hidden neuron count, activation function, and resblock for maximizing accuracy. The MABOS algorithm is simple to understand and simple to use. But, it continues to experience problems such as unequal exploitation, lack of diversity, and local optimum. It is simple to fall into an optimal local situation in high-dimensional space. The GFA algorithm increases computing effectiveness and control parameter resilience. But, the convergence rate is poor. That takes a huge amount of time to finish. The GFA algorithm suffers from processing vast amounts of data in order to produce superior results. The implementation is also very challenging. So, they developed MABGFA optimization is adapted to segment and classify the disease from fundus images. The MABGFA optimization is designed related to mean, worst and best fitness functions. Here, the random number to be distributed is indicated by rnd. In the traditional optimization, the term rnd is chosen in the interval of [0,1]. But, in the developed algorithm, it is estimated using Eq. (1).

$$rnd = \frac{\sqrt{worstfit} + \sqrt{meanfit}}{besfit} \tag{1}$$

The best fitness is represented as $besfit$ and the worst fitness value is indicated by $worstfit$. The term $meanfit$ indicates the mean fitness value. By using this newly estimated value, this algorithm converges very fast and gives superior optimization results in the solution space.

MABOS [26]: The balance between exploration and exploitation is crucial in the classic reinforcement learning problem known as Multi-Armed Bandits (MAB). Here, various utility qualities are used that are named armed bandits. Initially, there is no prior knowledge of bandits. The resource is fixed and limited, which can be seen in terms of computing time. When they provide the bandit's resources, the objective is to decrease the regret or increase the gain. The estimated parameter is denoted by R. The true-action parameter is denoted by r. The term b is the selected action. The expected reward is calculated using Eq. (2).

$$r(u,b) = F[S(u)B(u)] = b \tag{2}$$

Here, the iteration count is denoted by u. The term $B(u) = b$ indicates the selected action. The current reward is indicated by $S(u)$. The term $B^*(u)\arg\max_b R(u,b)$ indicates the greedy action. It is measured using $R(u,b)\alpha\, r(u,b)$. The optimal value is calculated using Eq. (3).

$$w^*(u) = r(b^*,u) = \max_{b\in B} r(b,u) \tag{3}$$

The opportunity loss is calculated by the total number of regret and it is given in Eq. (4).

$$M_U = F\left[\sum_{u=1}^{U}\left(w^*(u) - r(b(u),u)\right)\right] \tag{4}$$

Hence, regret is unavoidable because they can't always choose the best action. If the system changes to smooth, they anticipate the estimation to get more accurate results. Initially, the action is chosen for every time and, a reward is collected, the evaluation of the action value is updated. The term b is denoted by the action operator and it is chosen by using the number of times parameter $O_u(b)$. It is given in $g\left(S_1, S_2, \ldots, S_{O_u(b)}\right)$. The achieved rewards are measured by the term R and it is given in $R(u,b) = \sum_{j=1}^{O_u(b)} S_j / O_u(b)$. Using the non-stationary and stationary measures the updated increment is calculated using Eq. (5).

$$R(u+1,b) = (\beta)R(u,b) + (1-\beta)S(u) \tag{5}$$

Here, the term hyper parameter is denoted by β and it is selected using $(\beta \in [0,1])$. Smaller values of β are suitable for dynamic systems because they focus more on recent rewards. On the other hand, choosing a small value β relies on the present reward. Hence, choosing this parameter requires better balance. The experiments section includes the value of β and this value is utilized for this algorithm.

The main issues with this problem are known to be the most crucial part of MAB to maintain the balance between the exploration for improved estimates and unreliable estimates in exploitation. It is used to increase the regret with the low cost. In this algorithm, the action uncertainty is considered to determine the expected value. The term VDC is measured using Eq. (6).

$$VDC(u,b) = R(u,b) + dV(u,b) = R(u,b) + d\sqrt{\frac{L(u)}{M_u(b)}} \tag{6}$$

Here, the hyper parameter is denoted by d and it increases the confidence level of the upper bound in final validation. The non-negative real value is indicated by d and it decreases the uncertainties in the final validation. The term $soft\max$ is used for selecting the VDC action easily, and it is measured using in Eq. (7).

$$Q(B=b) = soft\max(VDC(b)) = \frac{\exp[VDC(b)]}{\Sigma_{b\in B}\exp[VDC(b)]} \tag{7}$$

The fitness value is increased using the term L. If the condition is $d = 0$ they do not change the final process. Balance is based on the selection of values. The term Q gives the best solution using the upper confidence actions and better-estimated value.

GFA [27]: In the suggested optimization technique and traveling multiple paths in search of a solution is the crucial issue. The dimension is indicated by E. The GFA gives the best solution. The GFA initialization is given in Eq. (8).

$$y_{jk} = mc_k(g) + rnd_k(1) \times (vc_k(g) - mc_k(g)) \tag{8}$$

The candidate solution is denoted by y_{jk}. The term rnd indicates the random number. Every candidate solution is determined by using the velocity. Every GFA velocity is assigned to 0 and it is given in Eq. (9).

$$w_j\left(\vec{0}\right) = 0 \tag{9}$$

Here, the initial velocity is indicated by $w_j\left(\vec{0}\right)$. The terms $E\vec{B}_j(u)$ and $M\vec{B}_j(u)$ are calculated using Eq. (10) and Eq. (11), respectively.

$$\begin{gathered} E\vec{B}_j(u) = rnd_j\left(E\vec{B}_1(u), E\vec{B}_1(u), \ldots, E\vec{B}_h(u)\right), \\ h = flr\left(\frac{Q\times o}{100}\right) \end{gathered} \tag{10}$$

$$M\vec{B}_j(u)=mean\left(y_k{}^{\rightarrow}(u)\right)\forall k \tag{11}$$

Here, the selected value of EB is indicated by $E\vec{B}_j(u)$. The term $M\vec{B}_j(u)$ denotes the local average value of GFA. The ratio calculations of Δi_{je} and Δi_{jm} are measured using Eq. (12) and Eq. (13), respectively.

$$\Delta i_{je}{}^{\rightarrow}(u)=E\vec{B}_j(u)-y_k{}^{\rightarrow}(u) \tag{12}$$

$$\Delta i_{jm}{}^{\rightarrow}(u)=M\vec{B}_j(u)-y_k{}^{\rightarrow}(u) \tag{13}$$

Here, the terms e and m represents the subscripts of EB and MB, respectively. The terms $M_{je}(u)$ and $M_{jm}(u)$ are measured using Eq. (14) and Eq. (15), respectively.

$$M_{je}(u)=\frac{\max_{k\neq j}\left|g\left(y_k{}^{\rightarrow}(u)\right)-g\left(y_k{}^{\rightarrow}(u)\right)\right|}{\left|g\left(EBy_k{}^{\rightarrow}(u)\right)-g\left(y_k{}^{\rightarrow}(u)\right)\right|} \tag{14}$$

$$M_{jm}(u)=\frac{\max_{k\neq j}\left|g\left(y_k{}^{\rightarrow}(u)\right)-g\left(y_k{}^{\rightarrow}(u)\right)\right|}{\left|g\left(MBy_k{}^{\rightarrow}(u)\right)-g\left(y_k{}^{\rightarrow}(u)\right)\right|} \tag{15}$$

Darcy's law calculates the velocity and it is directly proportional to the term IH and it is calculated by Eq. (16) and Eq. (17), respectively.

$$IH_{je}{}^{\rightarrow}(u)=\frac{\Delta i_{je}{}^{\rightarrow}(u)}{M_{je}{}^{\rightarrow}(u)} \tag{16}$$

$$IH_{jm}{}^{\rightarrow}(u)=\frac{\Delta i_{jm}{}^{\rightarrow}(u)}{M_{jm}{}^{\rightarrow}(u)} \tag{17}$$

The term we is a discharge velocity. The coefficient factor is denoted by l. The discharge velocity is measured by Eq. (18) and Eq. (19), respectively.

$$we_{je}{}^{\rightarrow}(u)=l\times IH_{je}{}^{\rightarrow}(u) \tag{18}$$

$$we_{jm}{}^{\rightarrow}(u)=l\times IH_{jm}{}^{\rightarrow}(u) \tag{19}$$

Both discharge velocities of EB and MB are combined using Eq. (20).

$$we_j{}^{\rightarrow}(u)=\beta\times we_{je}{}^{\rightarrow}(u)+(1-\beta)\times we_{jm}{}^{\rightarrow}(u) \tag{20}$$

Here, the term β controls the water levels using the selected weight range in $[0,1]$. Water's discharge velocity is its speed when there is no barrier in its path. In general, groundwater travels via porous media called aquifers. Hence, the aquifers' pores are the only places where water can pass. The division of discharge velocity and the intermediate medium is known as velocity of seepage. The velocity of seepage is measured using Eq. (21).

$$wt_j{}^{\rightarrow}(u)=\frac{we_j{}^{\rightarrow}(u)}{\gamma_j} \tag{21}$$

The intermediate medium is denoted by γ. The dimension is updated using DG and it is indicated by e. The term DG is a controlling factor calculated by Eq. (22).

$$DG(u)=\left(1-\frac{u}{U}\right)\times E \tag{22}$$

Here, the term U is the maximum number of iterations. The randomly selected integer is indicated by e. The dimensions are updated by Eq. (23).

$$\gamma_j=\frac{e_j}{E} \tag{23}$$

The velocity of $w_j{}^{\rightarrow}(u)$ is measured using Eq. (24).

$$w_j{}^{\rightarrow}(u)=\alpha\times w_j\,(\overrightarrow{u-1})+\kappa\times wt_j{}^{\rightarrow}(u) \tag{24}$$

Finally, the velocity of any one GFA can be determined by adding its seepage velocity to its previous velocity. When all GFAs share the same fitness measures, the best solution can be found. The pseudo-code of the investigated MABGFA is given in Algorithm 1.

Algorithm 1: Designed MABGFA

```
Initialize the iterations $U$ and population $y_j$
Load the initial positions
Calculate the initial fitness value
  **Update the parameter $rnd$ with the adaptive concept**
  While $(u \le U)$
      For
    $(j = 1 to Maxiter)$
    For $(k = 1 to pop)$
        If $(rnd < 0.5)$
                        Update the parameter using MABOS in Eq. (9).
                        Evaluate the seepage velocity
        Else:
                        Update the parameter using GFA in Eq. (23).
        End if
        Calculate the discharge velocity
     End For
    Select the value using softmax sampled by Eq. (7).
    Validate the best fitness solution
  End while
  Return the parameters
End
```

V. Abnormality Segmentation and Regional Segmentation for Diabetic Retinopathy Classification using Fundus Images

A. Preprocessing of Fundus Images

The collected fundus images are given to the pre processing stage indicated by A_q^{Np}. Image enhancement is the technique of enhancing the original data images and details. Typical techniques add the density slicing, contrast enhancement and spatial filtering,. By applying a linear transformation, the original range of grey levels is expanded, resulting in contrast enhancement. The natural linear factors such as shear zones, lineaments and fault are improved by contrast enhancement. In order to indicate various aspects, density slicing breaks up the continuous intervals, each designated by a different color. The preprocessed image is denoted by A_m^{pre}.

B. Active Contour-based Optic Disc Removal

The preprocessed image A_m^{pre} is given in the segmentation section. The optical disk is eliminated from the inverted equalization image before it is given to element. It is used to open the edge-improved image based on curvelets in order to remove it. For the purpose of removing the optic disc, the morphological procedure is applied, which is used for the complete removal of the optic disk. The term $Dj - Tf$ defines the erosion process and it is calculated using Eq. (25).

$$Dj - Tf = \{W \in F / Ty \subseteq Dj\} \tag{25}$$

Here, the initial input is indicated by Dj. As a result, the term Ty is measured using Eq. (26).

$$Ty = \{g + y / g \in Tf\}, \forall_y \in F \tag{26}$$

The active contour is a recent method that divides the further analysis and processing using energy forces and challenges. The approach of collecting deteriorated structures from an image with respect to issues and energy forces for the segmentation of blood arteries is known as active contouring. The bounds of the image elements are decided by the contour model to produce a contour. Using a variety of approaches that include both internal and external forces, the structure's curvature is determined. The energy function is determined in order to acquire the required structure. To define contour deformations, a group of points that define a contour are used. As a result, the segmented image produced by active contouring is represented by KR_r^{out}.

C. Modified U-Net-based Abnormality Segmentation

The input given to the regional segmentation section is indicated by KR_r^{out}. The modified U-Net-based segmentation procedure will receive as its input. It is used to encode the fundus image using the Modified U-net encoder. The decoder contains number of U-Net blocks to given the output of the model. The output of the previous block is to be up sampled and activation features are sent from an intermediary layer of the encoder to each U-Net block.

Encoder and Decoder: The decoder and encoder are the two components that make up the modified U-Net. The encoder utilizes ResNet architecture that has already been trained. The residual network is primarily used because, when compared to other pre-trained models, it has a higher accuracy in image classification issues. It contains four blocks ReLu, decoder, encoder and batch normalization. There are four U-Net blocks in the decoder. The up sampling mechanism in the improved images convolution initialized to convolution block. The model's deconvolution layer has been swapped out for a sub-pixel convolution layer. Therefore, the segmented image produced by modified U-Net is represented by OT_s^{Bld}. The basic diagram of the Modified U-Net-based segmentation system is shown in Fig. 3.

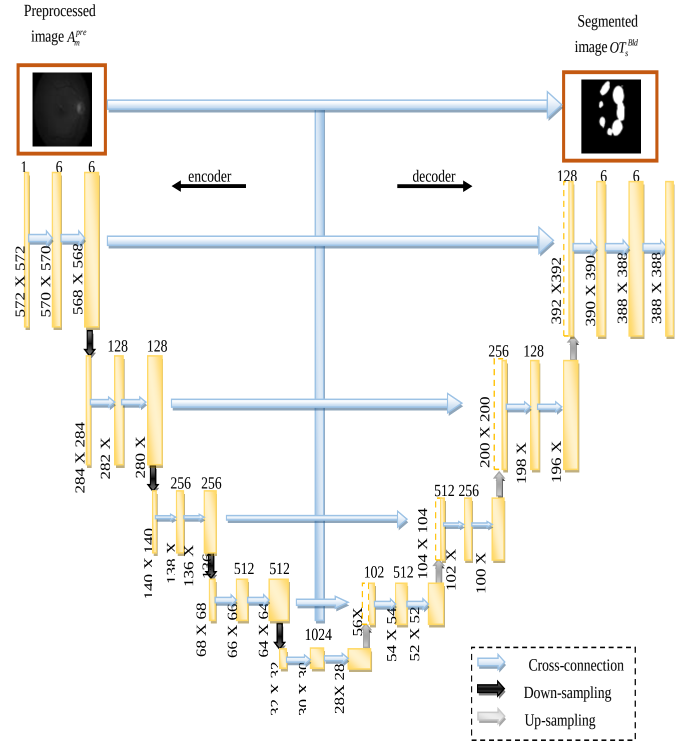


Fig. 3. Modified U-Net-based segmentation system.

VI. Classification of Diabetic Retinopathy using Hybridization of Soft Attention-based Densenet and Multi-Scale Gated Resnet

A. Soft Attention-based DenseNet

The segmented image given to the classification section is denoted by OT_s^{Bld}. The classification process is a one of the supervised learning technique. It adopted a data into different classes. It trained the models and predicts the groups.

DenseNet [29]: Transition and dense blocks are used to build the DenseNet model. The pooling layer is part of the Convolution unit in the transition block. The transition block is employed to eliminate the complexity of computation. The bottleneck layer used in the convolution reduces the input feature map sizes. Therefore, it minimizes the feature mappings at transition layers. The transition stage receives the output.

Soft attention block: Soft attention increases the robust ssystem. It favors the most pertinent input while allowing a portion of the other data to influence the system's judgment. It is based on the precise location determined by the feature map that was extracted. The term T is a soft attention score and it is indicated by Eq. (27).

$$T = \sum_{L=1}^{L} \frac{\exp\left(g_{3e_{jk}}\right)}{\sum_{j=1}^{x^y}\sum_{k=1}^{i^y}\exp\left(g_{3e_{jk}}\right)},\ g_{3e} = I\left(g^{y}\right) \tag{27}$$

$$\beta_z = g^{y} + zg_t^{y} \tag{28}$$

This model is assisted in choosing the precise positions of the feature map's relevant properties by the finished soft attention layer.

B. Multi-Scale Gated ResNet

Multi-scale Gated model [32]: They develop an attention-gating module for the top-down adaptive fusion as opposed to the typical method of feature concatenation from several layers or multi-scale inputs. It directs the second level's learning through the gating operation, the fusion process begins. The hierarchical fusion operation is calculated using Eq. (29).

$$B_j = \beta\left(X^{7\times7}\left(\left[AvgPl\left(G_j\right); MaxPl\left(G_j\right)\right]\right)\right) \tag{29}$$

$$G^{`}_{j+1} = up\left(B_j\right)\circ C\left(G_{j+1}\right)\left(j = 1,2,3\right) \tag{30}$$

Finally, the gated feature at the top level includes both high-level semantic knowledge and high-resolution spatial data. The saliency map is normalized using a sigmoid activation function.

ResNet: The DR is primarily classified with the resnet technique. The block of the resnet is used to determine the direct residual output. The term $J(z)$ is a target output. The degradation happens much more quickly when there are multiple convolutional layers. The resnet has a lot of small connections that allow us to identify the mapping process indirectly by skipping one or two tiers. The residual mapping function is given in Eq. (31).

$$H = B_2\gamma\left(B_1 z\right) \tag{31}$$

Here, the term γ indicates the Relu activation function.

$$a = H\left(z,\{B_k\}\right) + z \tag{32}$$

Here, the term a is a representation of the second Relu's shortcut connection. Finally, they effectively predict the DR disease and also increase the performance of classification using res net.

C. Proposed HSADMGRNet-based Diabetic Retinopathy Classification

The HSADMGR Net-based system enhances the classification effectiveness over the segmentation and classification of DR with high accuracy. The designed MABGFA optimization is utilized for optimizing the parameters like res block, hidden neuron count and activation function. The HSADMGR Net model with MABGFA optimization maximizes the accuracy and precision of the designed DR segmentation and classification model. The Soft Attention-based DenseNet method improves classification accuracy. It takes less time to segment the DR disease. DenseNet replicates the data numerous times and splices the feature maps of each layer with the layer before. Sometimes it makes misdiagnosis results. A Multi-Scale Gated ResNet approach considerably reduces the error rate below a certain threshold, further enhancing the network. But, it takes weeks of segmentation, which makes it practically impossible to use in real-time. The computational density of the ResNet method is one of its main disadvantages. To resolve these problems, the suggested MABGFA-HSADMGR Net-based DR classification model is developed to enhance the classification results. The developed MABGFA-HSADMGR Net-based DR classification model gives better fitness in terms of maximization of accuracy and precision and it is given in Eq. (33).

$$Ob_j = \underset{\left\{\begin{array}{l} SA_{DesNet}^{Linear}, SA_{DesNet}^{sigmoid}, SA_{DesNet}^{\tanh}, SA_{DesNet}^{\mathrm{Re}Lu} \\ KJ_{\mathrm{Re}sNet}^{Linear}, KJ_{\mathrm{Re}sNet}^{sigmoid}, KJ_{\mathrm{Re}sNet}^{\tanh}, KJ_{\mathrm{Re}sNet}^{\mathrm{Re}Lu}, \\ KJ_{\mathrm{Re}sNet}^{Hidden}, KJ_{\mathrm{Re}sNet}^{resbk} \end{array}\right\}}{\arg\min}\left(\frac{1}{Ac} + \frac{1}{\mathrm{Pr}}\right) \tag{33}$$

Here, the optimized ReLu, linear, tanh and sigmoid activation functions from the DenseNet are denoted by $SA_{DesNet}^{\mathrm{Re}Lu}$, SA_{DesNet}^{Linear}, $SA_{DesNet}^{\tanh}$ and $SA_{DesNet}^{sigmoid}$ in the interval of $[0,3]$, respectively. The optimized ReLu, linear, tanh and sigmoid activation functions from the ResNet are indicated by $KJ_{\mathrm{Re}sNet}^{\mathrm{Re}Lu}$, $KJ_{\mathrm{Re}sNet}^{Linear}$, $KJ_{\mathrm{Re}sNet}^{\tanh}$ and $KJ_{\mathrm{Re}sNet}^{sigmoid}$ in the interval of $[0,3]$, respectively. The hidden neuron count from ResNet is indicated by $KJ_{\mathrm{Re}sNet}^{Hidden}$ and it is selected in the interval of $[5,255]$. The optimized res block in the ResNet is indicated by $KJ_{\mathrm{Re}sNet}^{resbk}$ and it is selected in the interval of $[3,15]$. The precision parameter is calculated using Eq. (34).

$$PR = \frac{UO_f}{UO_T + UF_m} \tag{34}$$

The accuracy is measured using Eq. (35).

$$A = \frac{\left(UO_T + UF_p\right)}{\left(UO_T + UF_p + UO_f + UF_m\right)} \tag{35}$$

The true positive and negative values are denoted by UO_T and UO_f, respectively. The false positives and false negatives are indicated by UF_p and RU_s, respectively. A diagrammatic representation of the developed MABGFA-HSADMGR Net-based DR classification is shown in Fig. 4.

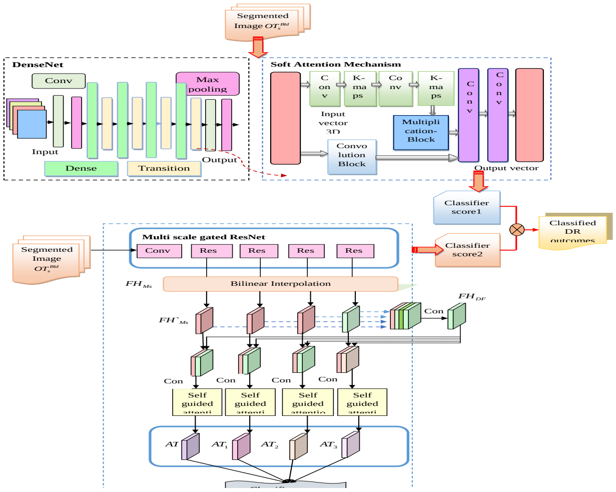


Fig. 4. MABGFA-HSADMGR Net-based DR classification system.

VII. Results and Discussions

A. Experimental Setup

The developed MABGFA-HSADMGR Net-based DR segmentation and classification model was developed by using Python environment. The performance of the DR segmentation and classification model was validated over different classification methods and hybrid algorithms via many evaluation measures. The analysis was conducted using a population rate of 10, an iteration gap of 10 and a chromosomal length of 5. For validation, classifiers including Mobile Net [28], Dense net [29], VGG16 [30], XGBOOST [31], MG_Res Net [32], and RN-Bi-LSTM [33] were considered and several optimization algorithms like Harris hawks optimization (HHO) [34], Salp Swarm Algorithm (SSA) [35], MABOS [26], and GFA [27] were utilized.

B. Pre-Processing and DR Segmentation Output

The outputs of the Modified U-Net-based DR disease segmentation system are given in Fig. 5. The segmentation performed utilizing the Modified U-Net approach.

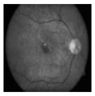
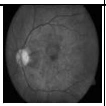
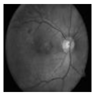
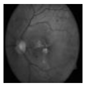
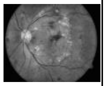
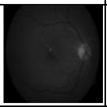
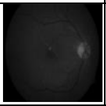
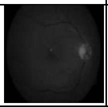
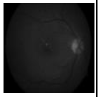
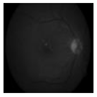
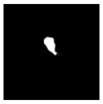
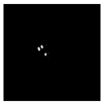
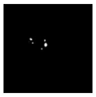
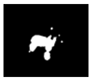
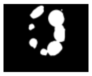
Description	Sample1	Sample 2	Sample 3	Sample 4	Sample 5
Original retinal images					
Pre-processed retinal images					
Modified U-Net-based Segmented retinal images					

Fig. 5. Outcomes of the designed Modified U-Net-based DR segmentation.

C. Evaluation Measures

The effectiveness measures used for the implemented DR segmentation and classification model are described as below.

a) FNR: The FNR is measured by Eq. (36).

$$FNR = \frac{UO_T}{UF_p + UF_m} \quad (36)$$

b) NPV: It is measured using Eq. (37).

$$NPV = \frac{UF_p}{UF_p + UO_m} \quad (37)$$

c) Sensitivity: It is validated using Eq. (38).

$$SEN = \frac{UF_m}{UF_m + UO_f} \quad (38)$$

d) FPR: The FPR is calculated using Eq. (39).

$$FPR = \frac{UF_p}{UF_p + UO_f} \quad (39)$$

e) F1-score: The F1-score value is measured using Eq. (40).

$$F1 = \frac{2 \times UF_m}{2UF_m + UO_f + UO_T} \quad (40)$$

f) MCC: The MCC value is calculated by Eq. (41).

$$MCC = \frac{UO_T \times UO_f - UF_m \times UF_p}{\sqrt{(UO_T + UF_m)(UO_T + UF_p)(UO_f + UF_m)(UO_f + UF_p)}} \quad (41)$$

g) FDR: it is measured by Eq. (42).

$$FDR = \frac{UF_p}{UF_p + UO_T} \quad (42)$$

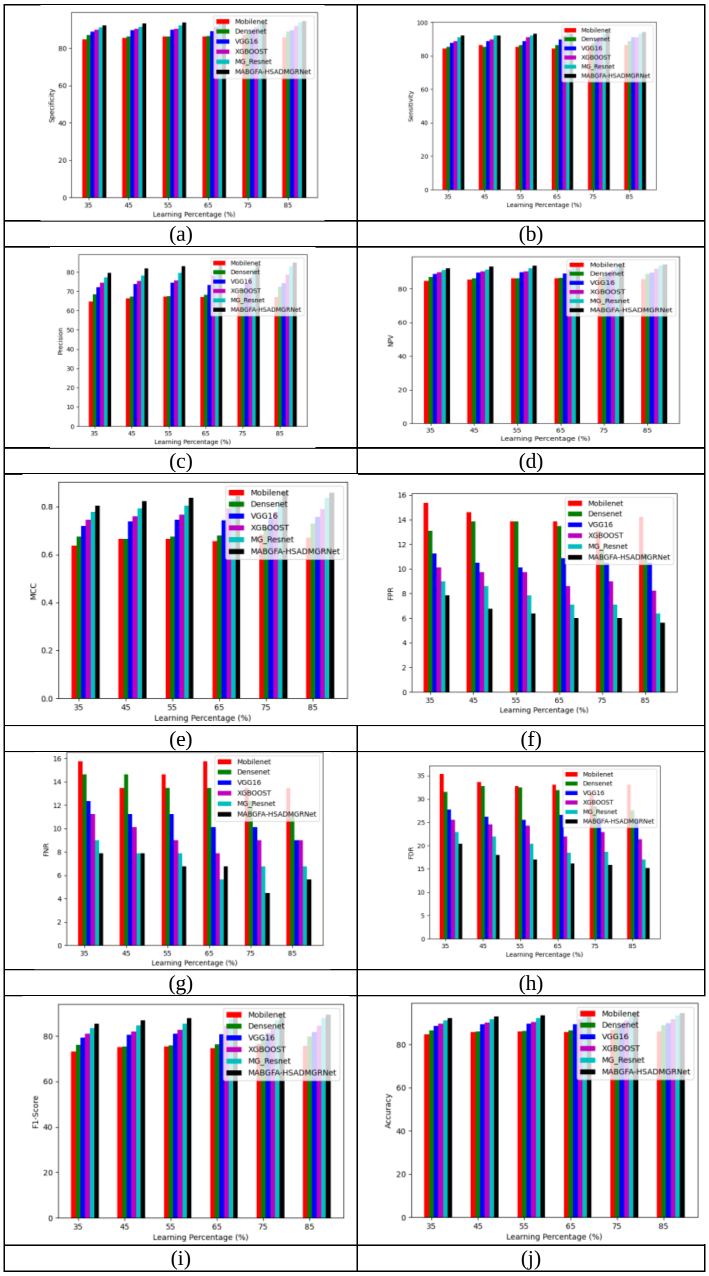
h) Specificity: The specificity is measured by Eq. (43).

$$SP = \frac{UO_f}{UO_f + UF_m} \quad (43)$$

VIII. Result Analysis

A. Performance Analysis of the Developed DR Segmentation and Classification Model with respect to Learning Percentage

Performance comparison of the developed MABGFA-HSADMGR Net-based DR segmentation and classification system to conventional techniques and hybrid algorithms is shown in Fig. 6 and Fig. 7, respectively. Then, the developed MABGFA-HSADMGR Net-based DR segmentation and classification system improved with a high sensitivity of 18% more than Mobile Net, 16.6% more than Dense Net, 15.2% more than VGG16, 12.6% than XGBOOST, 8.8% than MG_Res Net and 7.6% than RN-Bi-LSTM with the value of learning percentage is 35. As a result, the designed MABGFA-HSADMGR Net-based DR segmentation and classification model has achieved better classification rates when compared to conventional techniques and heuristic strategies.

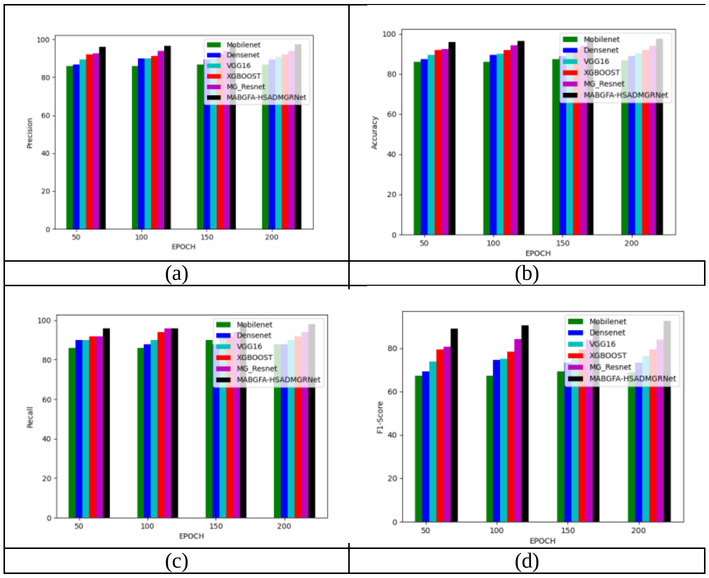


(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 6. Performance validation of various heuristic algorithms in terms of “(a) Specificity, (b) Sensitivity, (c) Precision, (d) NPV, (e) MCC, (f) FPR, (g) FNR, (h) FDR, (i) F1-Score, and (j) Accuracy”.

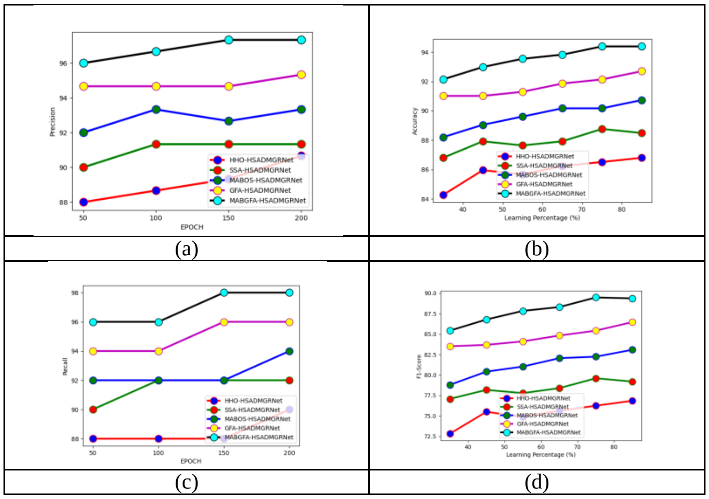
B. *Performance Analysis of the Developed Model with respect to Epochs*

A performance comparison of the developed MABGFA-HSADMGR Net-based DR segmentation and classification model to existing techniques and algorithms to existing techniques and algorithms in terms of epoch analysis is shown in Fig. 8, and Fig. 9, respectively. In the Epoch value of 100, the developed MABGFA-HSADMGR Net-based DR segmentation and classification showed a high precision of 18.6% more than Mobile Net, 18.1% more than Dense Net, 18.1% than VGG16, 15% than XGBOOST, 13% than MG_Res Net and 12% than RN-Bi-LSTM. Therefore, the designed MABGFA-HSADMGR Net-based DR segmentation and classifications system has achieved high efficacy due to accuracy.



(a) (b) (c) (d)

Fig. 7. Performance analysis of over various techniques in terms of “(a) Precision, (b) Accuracy, (c) Recall, (d) F1-Score”.



(a) (b) (c) (d)

Fig. 8. Performance analysis of over different heuristic algorithms in terms of “(a) Precision, (b) Accuracy, (c) Recall, (d) F1-Score”.

C. *Cost Function Analysis of the Developed DR Classification and Segmentation System*

A performances comparison of the developed MABGFA-HSADMGR Net-based DR segmentation and classification system to previous classification techniques and hybrid algorithms to existing techniques and algorithms in terms of cost function analysis is shown in Fig. 6. In the value of 5, The developed MABGFA-HSADMGR Net-based DR segmentation and classification showed with cost function of 4.4% than HHO-HSADMGR Net, 4.2% than SSA-HSADMGR Net, 3.8% than MABOS-HSADMGR Net, 4.1% than GFA-HSADMGR Net. Therefore, the designed MABGFA-HSADMGR Net-based DR segmentation and classification model has showed high efficacy.

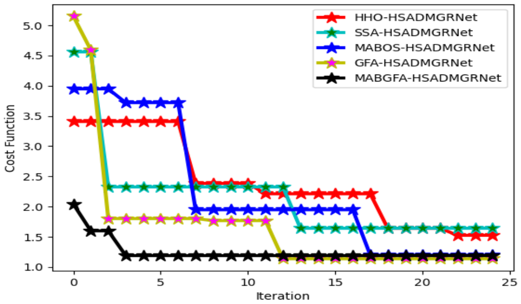


Fig. 9. Cost function analysis of suggested DR classification and segmentation model among different heuristic algorithm.

D. Overall Validation of the Implemented DR Segmentation and Classification System

The newly designed MABGFA-HSADMGR Net-based DR segmentation and classification model performance is analyzed using various heuristic algorithms and existing methods, shown in Table II and Table III, respectively. The developed DR classification and segmentation model has improved the f1-score of 3.4% than HHO-HSADMGR Net, 7.2% than SSA-HSADMGR Net, 9.8% than MABOS-HSADMGR Net, 14.1% than GFA-HSADMGR Net. The developed model performed more accurately and efficiently over previous hybrid algorithms and existing methods.

TABLE II. PERFORMANCE ANALYSIS OF DR SEGMENTATION AND CLASSIFICATION SYSTEM WITH HEURISTIC ALGORITHM

Terms	HHO-HSADMGR Net [28]	SSA-HSADMGR Net[29]	MABOS-HSADMGR Net[30]	GFA-HSADMGR Net[31]	MABGFA HSADMGR Net
Accuracy	86.51685	88.76404	90.16854	92.13483	94.38202
Sensitivity	86.51685	87.64045	91.01124	92.13483	95.50562
Specificity	86.51685	89.13858	89.88764	92.13483	94.00749
Precision	68.14159	72.8972	75	79.61165	84.15842
FPR	13.48315	10.86142	10.11236	7.865169	5.992509
FNR	13.48315	12.35955	8.988764	7.865169	4.494382
NPV	86.51685	89.13858	89.88764	92.13483	94.00749
FDR	31.85841	27.1028	25	20.38835	15.84158
F1-Score	76.23762	79.59184	82.2335	85.41667	89.47368
MCC	67.94095	72.51063	76.20008	80.47167	85.98179

TABLE III. PERFORMACE ANALYSIS OF DEVELOPED DR SEGMENTATION AND CLASSIFICATION MODEL WITH EXISTING METHODS

Terms	HHO-HSADMGRNet [28]	SSA-HSADMGRNet[29]	MABOS-HSADMGRNet[30]	GFA-HSADMGRNet[31]	MABGFA HSADMGRNet
Worst	3.408678	4.560688	3.948538	5.15174	2.04325
Best	1.524702	1.641787	1.197277	1.136816	1.189257
Mean	2.400973	2.176032	2.231579	1.696258	1.256214
Median	2.212312	2.325272	1.948192	1.136816	1.189257
Std	0.691964	0.775711	1.040727	0.989689	0.195281

IX. CONCLUSION

A newly developed DR segmentation and classification system is used to detect the DR in the early stage with high accuracy. The retinal fundus images were gathered through resources in internet. The gathered fundus images were given to first stage of pre processing step. Here, the fundus images would be performed by contrast enhancement technique. Then, the pre processed images undergoes to the segmentation section. Here, the pre processed images were segmented by two types of segmentation such as abnormality segmentation and regional segmentation. The abnormality segmentation could be performed using an Adaptive contour-based optic disc removal technique. The regional segmentation could be performed using the Modified U-Net method technique. Then, the segmented images were given to the classification stage. The classification could be done using the HSADMGR Net model. Here, the implemented MABGFA optimization was adapted to optimize the values such as activation function, hidden neuron count and resblock to maximize the precision and accuracy. The suggested MABGFA-HSADMGR Net-based DR classification and segmentation model achieved high performance in terms of F1-score of 13.6% than Mobile Net, 16.1% than Dense net, 14.1% than VGG16, 16% than XGBOOST, 17% than MG_Res net and 15% than RN-Bi-LSTM. The designed HSADMGR Net-based DR segmentation and classification system efficacy was validated among different traditional methods and hybrid algorithms in terms of accuracy and it showed greater performance.

REFERENCES

[1] F. Saeed, M. Hussain and H. A. Aboalsamh, "Automatic Diabetic Retinopathy Diagnosis Using Adaptive Fine-Tuned Convolutional Neural Network," IEEE Access, vol. 9, pp. 41344-41359, 2021.

[2] K. Shankar, Y. Zhang, Y. Liu, L. Wu and C. -H. Chen, "Hyperparameter Tuning Deep Learning for Diabetic Retinopathy Fundus Image Classification," IEEE Access, vol. 8, pp. 118164-118173, 2020.

[3] L. Qiao, Y. Zhu and H. Zhou, "Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms," IEEE Access, vol. 8, pp. 104292-104302, 2020.

[4] Xiyue Wang, Yuqi Fang, Sen Yang, Delong Zhu, Minghui Wang, Jing Zhang, Jun Zhang, Jun Cheng,"CLC-Net: Contextual and local collaborative network for lesion segmentation in diabetic retinopathy images", Neurocomputing, vol. 527, pp. 100-109, 28 March 2023.

[5] Chenrui Zhang, Ping Chen, Tao Lei, "Multi-point attention-based semi-supervised learning for diabetic retinopathy classification", Biomedical Signal Processing and Control, vol. 80, pp.1044, 12 February 2023.

[6] T. Jemima Jebaseeli, C. Anand Deva Durai, J. Dinesh Peter,"Retinal blood vessel segmentation from diabetic retinopathy images using tandem PCNN model and deep learning based SVM", Optik, vol. 199, pp.163328, December 2019.

[7] Devidas T. Kushnure, Sanjay N. Talbar,"HFRU-Net: High-Level Feature Fusion and Recalibration UNet for Automatic Liver and Tumor Segmentation in CT Images", Computer Methods and Programs in Biomedicine, vol. 213, pp. 106501, January 2022.

[8] Romany F. Mansour, "Deep-learning-based automatic computer-aided diagnosis system for diabetic retinopathy", Biomedical Engineering Letters, vol. 8, p.41–57, 2018.

[9] Sri Laxmi Kuna and Dr.A. V. Krishna Prasad, “An Efficient Meta-Heuristic-Feature Fusion Model using Deep Neuro-Fuzzy Classifier” International Journal of Advanced Computer Science and Applications(ijacsa), 13(11), 2022.

[10] H. Narasimha-Iyer et al., "Robust detection and classification of longitudinal changes in color retinal fundus images for monitoring

diabetic retinopathy," IEEE Transactions on Biomedical Engineering, vol. 53, no. 6, pp. 1084-1098, June 2006.

[11] I. P. Okuwobi, Z. Ji, W. Fan, S. Yuan, L. Bekalo and Q. Chen, "Automated Quantification of Hyperreflective Foci in SD-OCT With Diabetic Retinopathy," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 4, pp. 1125-1136, April 2020.

[12] K. A. Goatman, A. D. Fleming, S. Philip, G. J. Williams, J. A. Olson and P. F. Sharp, "Detection of New Vessels on the Optic Disc Using Retinal Photographs," IEEE Transactions on Medical Imaging, vol. 30, no. 4, pp. 972-979, April 2011.

[13] L. Seoud, T. Hurtut, J. Chelbi, F. Cheriet and J. M. P. Langlois, "Red Lesion Detection Using Dynamic Shape Features for Diabetic Retinopathy Screening," IEEE Transactions on Medical Imaging, vol. 35, no. 4, pp. 1116-1126, April 2016.

[14] A. Bilal, G. Sun, Y. Li, S. Mazhar and A. Q. Khan, "Diabetic Retinopathy Detection and Classification Using Mixed Models for a Disease Grading Database," IEEE Access, vol. 9, pp. 23544-23553, 2021.

[15] K. Shankar, Y. Zhang, Y. Liu, L. Wu and C. -H. Chen, "Hyperparameter Tuning Deep Learning for Diabetic Retinopathy Fundus Image Classification," IEEE Access, vol. 8, pp. 118164-118173, 2020.

[16] B. Zhang, B. V. K. Vijaya Kumar and D. Zhang, "Detecting Diabetes Mellitus and Nonproliferative Diabetic Retinopathy Using Tongue Color, Texture, and Geometry Features," IEEE Transactions on Biomedical Engineering, vol. 61, no. 2, pp. 491-501, Feb. 2014.

[17] X. Wang et al., "Joint Learning of Multi-Level Tasks for Diabetic Retinopathy Grading on Low-Resolution Fundus Images," IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 5, pp. 2216-2227, May 2022.

[18] K. M. Adal, P. G. van Etten, J. P. Martinez, K. W. Rouwen, K. A. Vermeer and L. J. van Vliet, "An Automated System for the Detection and Classification of Retinal Changes Due to Red Lesions in Longitudinal Fundus Images," IEEE Transactions on Biomedical Engineering, vol. 65, no. 6, pp. 1382-1390, June 2018.

[19] Y. Yang et al., "Robust Collaborative Learning of Patch-Level and Image-Level Annotations for Diabetic Retinopathy Grading From Fundus Image," IEEE Transactions on Cybernetics, vol. 52, no. 11, pp. 11407-11417, Nov. 2022.

[20] S. Qummar et al., "A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection," IEEE Access, vol. 7, pp. 150530-150539, 2019.

[21] A. Osareh, B. Shadgar and R. Markham, "A Computational-Intelligence-Based Approach for Detection of Exudates in Diabetic Retinopathy Images," in IEEE Transactions on Information Technology in Biomedicine, vol. 13, no. 4, pp. 535-545, July 2009.

[22] E. Abdelmaksoud, S. El-Sappagh, S. Barakat, T. Abuhmed and M. Elmogy, "Automatic Diabetic Retinopathy Grading System Based on Detecting Multiple Retinal Lesions," IEEE Access, vol. 9, pp. 15939-15960, 2021.

[23] Sri Laxmi Kuna and A. V. Krishna Prasad, “Deep Learning Empowered Diabetic Retinopathy Detection and Classification using Retinal Fundus Images” International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11Issue:1, 2022 DOI: https://doi.org/10.17762/ijritcc.v11i1.6058.

[24] L. Dai et al., "Clinical Report Guided Retinal Microaneurysm Detection With Multi-Sieving Deep Learning," IEEE Transactions on Medical Imaging, vol. 37, no. 5, pp. 1149-1161, May 2018.

[25] E. O. Rodrigues, A. Conci and P. Liatsis, "ELEMENT: Multi-Modal Retinal Vessel Segmentation Based on a Coupled Region Growing and Machine Learning Approach," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 12, pp. 3507-3519, Dec. 2020.

[26] Kazem Meidani, Seyedali Mirjalili, and Amir Barati Farimani, "MAB-OS: Multi-Armed Bandits Metaheuristic Optimizer Selection," Applied Soft Computing, vol.128, pp.109452, 2022.

[27] Ritam Guhaa, Soulib Ghosha, Kushal Kanti Ghosha, Ram Sarkara, "Groundwater Flow Algorithm: A Novel Hydro-geologybased Optimization Algorithm," Engineering Optimization, 2020.

[28] H. Pan, Z. Pang, Y. Wang, Y. Wang and L. Chen, "A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects," IEEE Access, vol. 8, pp. 119951-119960, 2020.

[29] K. Zhang, Y. Guo, X. Wang, J. Yuan and Q. Ding, "Multiple Feature Reweight DenseNet for Image Classification," IEEE Access, vol. 7, pp. 9872-9880, 2019.

[30] J. Duan and X. Liu, "Online Monitoring of Green Pellet Size Distribution in Haze-Degraded Images Based on VGG16-LU-Net and Haze Judgment," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-16, 2021.

[31] D. Zhang and Y. Gong, "The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure," IEEE Access, vol. 8, pp. 220990-221003, 2020.

[32] Z. Zhu et al., "Juggler-ResNet: A Flexible and High-Speed ResNet Optimization Method for Intrusion Detection System in Software-Defined Industrial Networks," IEEE Transactions on Industrial Informatics, vol. 18, no. 6, pp. 4224-4233, June 2022.

[33] R. Zhong, R. Wang, Y. Zou, Z. Hong and M. Hu, "Graph Attention Networks Adjusted Bi-LSTM for Video Summarization," IEEE Signal Processing Letters, vol. 28, pp. 663-667, 2021.

[34] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti and S. Mirjalili, "An Improved Harris Hawks Optimization Algorithm With Simulated Annealing for Feature Selection in the Medical Field," IEEE Access, vol. 8, pp. 186638-186652, 2020.

[35] J. Zhang and J. S. Wang, "Improved Salp Swarm Algorithm Based on Levy Flight and Sine Cosine Operator," IEEE Access, vol. 8, pp. 99740-99771, 2020.

[36] Kuna, Sri Laxmi and Prasad, Dr. A. V. Krishna, Deep Learning Models for Classification of Diabetic Retinopathy Color Fundus Images (September 14, 2022). Available at SSRN: http://dx.doi.org/10.2139/ssrn.4218649.

Legal Entity Extraction: An Experimental Study of NER Approach for Legal Documents

Varsha Naik[1], Purvang Patel[2], Rajeswari Kannan[3]
Pimpri Chinchwad College of Engineering, Pune, India[1, 3]
Dr. Vishwanath Karad MIT World Peace University, Pune, India[1, 2]

***Abstract*—In legal domain Name Entity Recognition serves as the basis for subsequent stages of legal artificial intelligence. In this paper, the authors have developed a dataset for training Name Entity Recognition (NER) in the Indian legal domain. As a first step of the research methodology study is done to identify and establish more legal entities than commonly used named entities such as person, organization, location, and so on. The annotators can make use of these entities to annotate different types of legal documents. Variety of text annotation tools are in existence finding the best one is a difficult task, so authors have experimented with various tools before settling on the best one for this research work. The resulting annotations from unstructured text can be stored into a JavaScript Object Notation (JSON) format which improves data readability and manipulation simple. After annotation, the resulting dataset contains approximately 30 documents and approximately 5000 sentences. This data further used to train a spacy pre-trained pipeline to predict accurate legal name entities. The accuracy of legal names can be increased further if the pre-trained models are fine-tuned using legal texts.**

Keywords—Named Entity Recognition; NER; legal domain; text annotation; annotation tools

I. Introduction

Artificial Intelligence (AI) has the potential to improve both the efficiency and accessibility of numerous legal processes [1]. In the current digital era, online document collections are growing rapidly. Technology and automation can help to extract information from these collections. As the amount of data continuously increasing, it is more and more necessary to access and process these data. The use of natural language processing is significant. NER, one of Natural Language Processing's (NLP) fundamental building blocks, can be used to develop AI applications in the legal domain [2]. Name entity recognition is a process of locating and classifying named entities in an unstructured text into predefined categories.

Name entity recognition is used to find a link to rigid notations in text that are related to well-known semantic classes like person, place, organization, etc. NER is used not only as a standalone tool for information extraction (IE) [3], but also in a variety of natural language processing (NLP) applications such as text understanding, information retrieval, automatic text summarization, question answering, machine translation, and knowledge base construction, and many others. Information retrieval, question-and-answer systems, machine translation, and many more applications use NER as a crucial pre-processing step [4].

To achieve high performance in NER, large amounts of knowledge in the form of feature engineering and lexicons have traditionally been required [5]. Also, there is great advancement in machine learning algorithms and deep learning algorithms in natural language processing and more specifically name entity recognition and information extraction [6]. Depending on the problem, such methods typically require a large set of manually annotated data,[5] whereas some machine learning algorithms rely on unsupervised techniques that do not require a large set of annotated data. There is an active learning-based clustering technique that is a subset of the semi-supervised technique and is used to reduce manual annotation time [7].

Annotation is a practice of adding linguistic and interpretive information to an electronic corpus of spoken or written linguistic data. Basically, annotation means adding a note to the input data. Annotation of words and characters are quite common for exactly distinctive medical specialty entities, resembling genes, proteins, and diseases [8]. In previous work, Jackson M.Steinkamp, Abhinav Sharma has annotated the unstructured clinical notes to identify the symptoms within the electronic health records. In another work related to the medical name entity recognition have prepared their dataset by annotating notes of pneumonia patients [2]. And, annotation between two words or phrases are also done for syntactic dependencies or identifying relation between two words in a sentence. For new annotation project or for doing annotation from scratch, typically includes a variety of activities including defining annotation schemas [9], developing guideline for annotations and defining entity type assembling appropriate collections of documents, and properly pre-processing those documents and create the final corpus [10].

One of the important tasks while annotation is selecting appropriate annotation tool given the large number of tools available and the lack of an up-to-date list of annotation tools and their respective pros and cons [11]. Therefore, extensive review of available tools must be done to avoid poor decision of selecting tools. Weak decision can lead to the unnecessary wastage of time of installing and converting document to the specific format for tools.

In this task, an extensive review of annotation tools for manual annotation of documents has been presented. The basic requirements for selecting the annotation tools have been defined. To gain a better understanding of name entity recognition, an ER-system for the legal domain has been created. The first step in creating a corpus of annotated judgment papers is to define relevant entities, which can mainly be categorized into two: domain-specific named entities like legal terms, Act, legal institutes, etc., and general named entities like person, location, date, etc. An ER-system built with a spacy pre-trained model is then presented.

Following contributions were made in this paper:

- An extensive review was conducted on manual annotation tools for creating NER training corpus.
- A corpus for legal name entity recognition was created, consisting of 5000 judgment papers with 8 legal named entities.
- An Entity-Relation model was developed based on spacy pretrained model.

II. Literature Review

NER is regarded as a crucial activity in the information extraction process. While numerous studies on name entity recognition have been conducted. Several datasets have been offered over a long period of time. CoNLL' 03 [12], which was taken from a German newspaper and is regarded as a language independent NER dataset, is one of the more well- known datasets. Many datasets from many fields, including medical, law, archaeology, and many more, are afterwards proposed [13].

Early NER systems relied on rules that were created by humans. A rule-based NER system is thought to take a long time to design. Researchers have created a NER System based on a machine learning algorithm to solve this issue. They used a few learning techniques, including supervised learning, semi-supervised learning, and unsupervised training. Alex Brandsen et al. [9] have used machine learning approach for predicting name entities from Dutch Excavation reports. Not only machine learning algorithms, but also satisfactory research on NER systems using deep learning algorithms and neural networks, are being conducted. A study conducted by Franck Dernoncourt et al. [14] successfully performs NER using ANN and obtain satisfactory result out of it. Thomas AF Green et al. [15] have included a benchmark CRF-based Entity recognition model of a manually created corpus of job description and achieve accuracy of approx. 60-65 %.

While many type of research on NER is carried out using deep learning and machine learning approach. Very Few studies have been done using pre-trained model like BERT [16]. Mugisha et al. [2] have published a detailed comparison of the neuro-linguistic modelling pipeline for predicting outcomes from medical text notes using patients with pneumonia. Li, Jianfu, et al. [17] in their study, they have fine-tuned pre- trained contextual language models to support the NER task on clinical trial eligibility criteria. They have systematically explored four pre-trained contextual embedding models for biomedical domain (i.e., BioBERT, BlueBERT, PubMedBERT, and SciBERT).

Table I summarizes the literature survey conducted during this study.

TABLE I. Summary of Literature Review

Year	Paper No.	Domain of study	Comments
2019	[6]	Symptom extraction with unstructured clinical notes	To create clinically useful information extraction tools, a task definition, dataset and simple supervised NLP model were used.
2022	[2]	Outcome Prediction from medical notes	A deep comparison of natural language modelling pipelines from outcome prediction from unstructured medical text notes.
2020	[9]	NER in Archaeological domain	Developed a training dataset for name entity recognition in archaeology domain, for which Doccano tool was used for annotation.
2022	[15]	Entity recognition in job descriptions	Created a benchmark suite for entity recognition in job description which includes annotation schema, baseline model, and training of corpus.
2020	[18]	Dataset for legal name entity recognition	A dataset created for NER in German federal court decisions
2022	[19]	NER on Indian judgment paper	Created a training corpus of Indian court judgment and developed a transformer-based legal NER baseline model.

III. Research Methodology

A systematic review of literature on various annotation tools used for NER was conducted. In this paper, the standard SLR procedures as described by the authors Chitu Okoli and Kira Schabram [20] were taken under consideration. This methodology demonstrates a detailed examination of the NER system across multiple domains, as well as a manual annotation tool and various annotation methods.

A. Research Question

One of the crucial steps in a systematic review is the research question. In order to maintain focus at the start of the study, we write research questions (RQ) that will adhere to the review procedure. Table II show the lists of research questions.

B. Search Method

1) Choose keyword: Table III shows a list of keywords used in the search for the paper from the online library.

2) Inclusion and exclusion criteria: Only conference and journal papers published in English language within the last five years were considered, and any papers currently under review were excluded. Table IV show the inclusion and exclusion criteria.

IV. Review of Text Annotation Tools

The formal description of the text annotation problem and annotation tools was presented, followed by a detailed discussion of the selection criteria for annotation tools. Next, an introduction was provided on the commonly used annotation tools.

A. Annotation Tools

Data annotation tools are software used to create high-quality annotated machine learning training data such as text, images, and videos. There are wide variety of annotation tools from open-source tools that developers can modify accordingly to freeware applications that are free to use. Let us first discuss what is text annotation, what is the need of it and discuss some type of text annotation.

B. Text Annotation: Needs and Type

1) Labelling procedures: Adding labels entails putting a word in a sentence that explains its type. It can be explained using emotions, technical terms, etc. For instance, the phrase “I am satisfied with this product, it is amazing” could be given a label like “happy”.

2) Adding matadata: Similar to this, relevant information can be added to the statement “Mahadevapura police have fled charge sheet against the accused alleging that he has committed an offence punishable under Section 354C of I.P.C R/w sec.66(E) of Information Technology Act” to help the learning algorithm priorities and concentrate on particular terms. One might write something like, “Mahadevapura (Location) police have fled charge sheet against the accused alleging that he has committed an offence punishable under Section 354C (Legal) of I.P.C. (Act) R/w sec.66(E) of Information Technology Act (Act)”.

3) Now let us discuss in brief some of the types of data annotation: Sentiment Annotation: Sentiment annotation is nothing more than the assignment of labels to feelings like sadness, happiness, anger, positivity, negativity, and neutrality. Any activity involving sentiment analysis can benefit from sentiment annotation. (For example, in retail, facial expressions can be used to assess customer satisfaction.)

a) Intent annotation: The intent annotation also identifies the sentences but emphasises the purpose or motivation behind the statement. A message such as “I need to talk to Sam” in a customer service situation, for example, may direct the call to Sam by himself, or a message such as ”I have a problem with the credit card” could direct the call to the team handling credit card issues.

b) Named entity recognition: The goal of named entity recognition (NER) is to find and categorise special expressions or predefined named entities in a sentence [21]. It is used to look up words based on what they mean, such as names of people or places. Information can be extracted using NER, together with information classification and categorization.

c) Semantic annotation: It can be also known as meaningful annotation, semantic annotation is the addition of metadata, supplementary data, or tags to text that contains concepts and entities, such as persons, places, or themes.

C. Selection of Tools

Tools that are known and have been mentioned in previous studies were listed. Google, Google Scholar, Scopus, and other online databases were searched for tools mentioned in annotation tool-related publications. There are a wide range of annotation tools available, but for this survey, the tools selected are the most widely used in any domain and meet the criteria.

There are a few requirements that have been studied and presented for an annotation tool. In this research, total 22 criteria are considered to evaluate annotating tools which are further divided into different groups such as input-output, publication, system criteria, and function. For these categories important features of tools are considered such as accessibility, usability, and cost. All these mention categories are listed in Table V along with the associated criteria.

The input-output or data criteria address the input-output format of document, schema for annotation, and input format for multi-media file. Publication criteria include the year of last publications, number of citations, and number of publications in last five years. System criteria indicate installations architecture and simplicity of installations, quality and quantity of documents, license of tools and OS support. And the last set of criteria is functional criteria which contains multimedia annotation support, support of multiple language other than English, automatic text annotation, pre-annotation support and data security.

TABLE II. LIST OF RESEARCH QUESTION

Sr. No.	Research Question
1.	Domains where Name entity recognition is used?
2.	Dataset related to Name entity recognition?
3.	Which annotation tool is used for creating corpus?
4.	What are challenges and issue faced wile manual annotating dataset using tool?
5.	What are various techniques used for annotating dataset for NER?

TABLE III. LIST OF KEYWORDS

Sr. No.	Keyword	No of Articles
1	Name Entity Recognition Corpus and Annotation tools	3
2	Name Entity Recognition Dataset and Manual Annotation	2
3	Manual Text Annotation and annotation tools	4
4	Name entity recognition and (Deep learning or Machine learning)	32
5	legal name entity recognition or medical name entity recognition	14
6	Text annotation tool or lighttag or Doccano or brat or label Studio	45
	Total	100

TABLE IV. LIST OF INCLUSION AND EXCLUSION CRITERIA

Inclusion criteria	Last five-year publication: 2017-2021
	All Open Access
	Only Conference Paper and Journal Paper
	Only considered the ER system for text dataset
Exclusion Criteria	Unpublished paper
	Literature other than English language

The features with which the tools must comply are listed below:

- It should be freely available.
- It should be a web application that can be downloaded or used online.
- It should be able to installed easily.
- It should be approachable.
- It should support multiple file format and export annotation in multiple formats.

To satisfy the availability criteria a tool must be instantly accessible, either for direct online usage (via a web user interface) or to download at the time of writing, without requiring consumers to get in touch with the developers. The availability also depends on whether the tool is free or licensed.

The tool must be a web application, which means that it must either be easily accessible online or may be downloaded and installed as a web application. The requirement that annotations be web-based ensures that annotators can focus completely on their annotation tasks without having to fight with tool installation. Manual annotation is a labor-intensive and difficult task in and of itself, and additional work may annoy the annotators and jeopardise the annotation process.

The survey requires the tool to function properly, and it is a requirement for practical experiments. A minimal set of features, as described by the criteria (as defined in Table V), should be accessible regardless of whether the tool is locally installable or accessible online for use. Therefore, there is no need to contact the developers for help because the tool should be simple to use or the documentation should be thorough enough.

Few more additional features are considered in this research other than the functionalities listed above which makes annotation process much easier such as the smallest unit of annotation (character or token), built-in domain-specific named entity extraction, and quick annotations such as keyboard shortcuts, pre-annotations, or ontology.

Some additional feature which are not compulsory for the annotation tools are listed below but they might be useful for most of the NLP based task:

- It can support multimedia.
- It can support multiple language.
- Integration with AI model for automatic annotation tools.
- Good and simple User Interface.

D. Selected Tools

In this section total eight tools are studied and selected for the research work are listed in Table VI, detailed discussion is done for the selected tools with respect to their features.

1) BRAT (Browser based rapid annotation tool): One of the most well-liked tools for manually annotating documents, it has been employed in the creation of numerous corpora. BRAT is a browser-based free online annotation tool for collaborative text annotation [22]. BRAT is not accessible online and must be installed locally. Documents are imported in the same format as the plain text file that contains the schema configuration. It was designed for rich structured annotation for a range of NLP activities. BRAT was created to enhance manual curation efforts and boost annotator productivity using NLP approaches. It is possible to highlight entities and relations as well as normalize data to pre-established terminology. It has a rich range of features such as integration with external resources such as Wikipedia, support for automatic text annotation tools, and built-in annotation comparison.

BRAT is more suitable for annotating expressions and the relationship between them, because annotating longer texts like paragraphs is really inconvenient. It only accepts text files as input documents, and text files are not presented in their original format in the user interface. Despite the fact that the last version was issued in 2012, the product is still readily accessible and well-liked in the industry. Recent upgrades include, among other things, integrating with external TM tools and embedding visualizations in HTML pages.

TABLE V. CRITERIA FOR SELECTION OF ANNOTATION TOOLS

Criteria Categories	Criteria
Input & Output (data)	Input format for document
	Input for multi-format file
	Format for annotation
	Output format for annotation
Publication	Number of citations
System Criteria	Installation Design (Web, standalone, plugin)
	Simplicity of installation
	Quality and Quantity of documentation
	Licence of Tool
	Operating System Support
	Availability (Free/ Paid)
Function	Multimedia or Multimodal Support
	Multilingual Support
	Interactive UI
	Support of Fast Annotation
	Full Text Support
	Inter-annotator agreement
	Pre-annotation Support
	Integration with external sources
	Automatic text annotation
	Annotation Relationship
	Data Security

TABLE VI. LIST OF SELECTED TOOLS

Tools	Installation	Input Format	Output Format	License
BRAT	Web	TXT	brat standoff format	CC BY3.0
Djangology	Web	DB	DB	-
Doccano	Web	TXT	JSON, CSV	MIT
GATE teamware	Web/ SA	TXT	XML, DB	GPL
Label Studio	Web/SA	TXT, JSON, CSV	CSV, JSON, CONLL	-
Lighttag	Web	TXT	JSON	-
Prodigy	SA	TXT	JSON, txt	-
UBIAI	SA	TXT, JOSN, PDF, HTML	JSON, Amazon Comprehend, Stanford CoreNLP	-

a) Doccano: Doccano is an open-source web-based annotation tool for text files only [23]. It is an open-source tool that supports a variety of job types, such as tasks involving the annotation of text sequences or text classification, which may be applied to a variety of problems, such as the annotation of text for sentiment analysis, text summarization, NER, etc. [9] It has a more modern and attractive user interface, and all configuration is done in the web user inter- face. It also generates a basic overview of tagging statistics. All of these make Doccano more beginner-friendly and user- friendly in general. It supports multiple users, but there are no additional features for collaborative annotation.

b) GATE: Gate team-ware is a web-based open-source collaborative annotation and curation tool [24] and is freely available. Gate teamware is an extension of an annotation tool GATE, which is an annotation management tool. GATE teamware offers user automatic annotation which reduces the manual annotation tool. It offers the interface which can be used to create corpus, to define annotation schema, to load pre-annotated data. As it is collaborative tool, it allows the users to monitor the annotation process i.e., number of annotated document and remaining document to be annotated. It is also use to monitor statistics like time spent on a document, inter annotator agreement.

c) Light tag: Another browser-based text labelling tool is LightTag [25], however it’s not completely free. No local installation is required for annotation using lighttag. It offers a free edition with 5,000 annotations each month for its essential features. It supports working with different languages (like Arabic, Hebrew and CJK among others), document level, multi-word, nesting, relationship annotations, etc. Additionally, it uses machine learning to learn from active annotators and suggest possible annotations for hidden text. It assigns tasks to annotators and ensures that there is enough overlap and duplication to maintain a high degree of accuracy and consistency.

d) Prodigy: It is a paid tool, and the only free version is a demo. Prodigy is an active learning-based annotation tool that is also connected with the Spacy library. This annotation tool’s active learning feature allows you to only annotate cases for which the model does not yet have an answer, greatly accelerating the annotation process. By using transfer learning technology and a more flexible approach to data gathering, you can train models of production quality with a minimal number of samples. Prodigy allows you to annotate images, videos, and audio in addition to text. When exporting your files, you can select among the JSONL, JSON, and txt formats.

e) UBIAI: UBIAI is a powerful labelling platform for training and deploying custom NLP models. UBIAI is a tool for data labelling as a service category in the technology stack [26] . It offers free and paid plans, OCR annotation tools, document classification, auto-tagging for team collaboration, and more. Widely used in the corporate world to convey important information, this is a must, especially for businesses and organizations that need to create high-quality annotations to PDFs, but difficult to edit there is. With UBIAI you can easily annotate native his PDF documents, scanned images, images, invoices or contracts in over 20 languages including Japanese, Spanish, Arabic, Russian and Hebrew can be attached. Per- form named entity recognition (NER), relationship extraction, and document classification in the same interface. Export annotations in multiple formats including Spacy, IOB, and Amazon Comprehend. Supports various input formats such as native PDF, TXT, CSV, PNG, JPG, HTML, DOCX, JSON. It also offers team management features that allow you to track progress. Measurement of text annotations, performance of assigned projects, and agreement among annotators.

f) Label studio: Label Studio is an open-source data labeller that allows you to label and explore a variety of data written in Python. You can make different entries with several data formats. You can also integrate Label Studio with machine learning models to provide label predictions (examples) or perform continuous active learning. Label Studio is also available in Enterprise and Cloud versions with additional features. Simplicity of label studio is that it has no complicated configurations, and ease of integration into Machine Learning pipelines. Label Studio can be used in different places, depending on different use-cases. It is quickly configurable for many data types. The tool gets ready in a few minutes. There is an easy way to switch between labelling texts, audios or images, or even annotating all three types at the same time. Many existing labelling frameworks accept only one data type, and it becomes tedious to learn a new app each time whereas Label Studio works with Texts, Images, Audios, HTML documents and any imaginable combination of annotation tasks like classification, regression, tagging, spanning, pairwise comparison, object detection, segmentation and so on. After configuring what the labelling interfaces should look like, you can import your data. The web import supports multiple formats: JSON, CSV, TSV, and archives consisting of those.

V. Experiment and Results

A. *Annotation of Dataset*

The following section describes the premise for dataset annotation, including the defining of annotation setups, various entity types, and annotation method (Annotation guidelines).

1) Selecting suitable input documents for annotation: In order to construct a robust dataset for legal named entity recognition, a comprehensive effort was undertaken to collect a diverse range of case documents from the Indian Supreme Court and several High Courts throughout India. The documents were sourced from a multitude of publicly available repositories on the web, including the official websites of these courts and prominent legal databases such as https://www.indianka noon.org, as well as numerous other legal repositories. The dataset was made sure to represent a wide range of court cases accurately and thoroughly from various jurisdictions through a long and complex process of data collection.

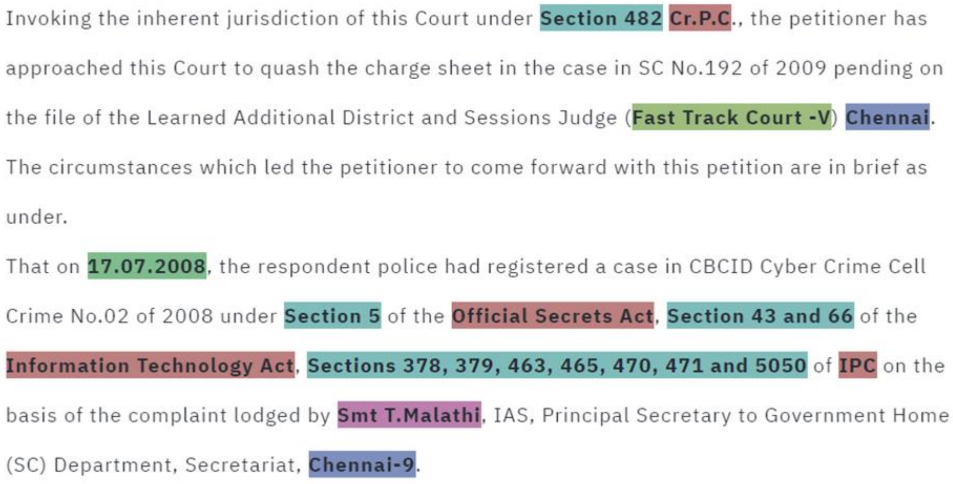
2) Annotation setup: Open-source data labeller Label Studio was used as an annotation tool. After comparing the system to other tools (as previously indicated), it was discovered that this was the most straightforward, user-friendly, and effective tool for our experimentation. There are several methods for installing label studio, including installing with pip, installing with docker, and installing from source, whether you are installing it locally or in the cloud. The only need for label- studio is that Python 3.6 or later must be installed on a machine running Linux, Windows, or MacOSX. Port 8080 is expected to be open by default in Label Studio. Label Studio installation needs SQLite 3.35 or later and PostgreSQL version11.5 or above. After installing Label Studio using pip, data was uploaded and entity types were defined in the tool after the system was downloaded and launched on a local machine.

3) Entity type: The targeted entities are listed in the Table VII, along with a brief description and an example for each category. After talking with legal experts on the pertinent information that may be gleaned from court rulings, the entity kinds were established.

Fig. 1 explain sample example of document to be annotated. The highlighted part of the text indicates the name entity to be annotated. The name entities that can be extracted from above text are given in Table VIII.

TABLE VII. List of Entities of the Legal Judgments

Name Entity	Descriptions	Example
PERSON	Name of the person	Praveen Kumar Wadi, Guruanna Vedi, B.L. Gupta
LOC	Locations which include name of states, cities, villages	Pune, Haryana, Gujarat, Mumbai
DATE	Any Date mentioned in judgment	10 April, 2001
ORG	Name of organization mentioned in text apart from the court.	General Insurance Co. Ltd.
Court	Name of the court which has delivered the judgment.	Supreme Court, Andhra Pradesh High Court, Bombay High Court
LEGAL	sections, Sub-sections, articles orders etc.	Section 110-A, Section 95(2)(d)
ACT	It includes Act name in constitution	Motor Vehicles Act, IT Act, Official Secret Act, IPC
CASE_NO	It indicates the particular case no. of court judgments	C.C. No. 3286 / 2019



Invoking the inherent jurisdiction of this Court under Section 482 Cr.P.C., the petitioner has approached this Court to quash the charge sheet in the case in SC No.192 of 2009 pending on the file of the Learned Additional District and Sessions Judge (Fast Track Court -V) Chennai. The circumstances which led the petitioner to come forward with this petition are in brief as under.

That on 17.07.2008, the respondent police had registered a case in CBCID Cyber Crime Cell Crime No.02 of 2008 under Section 5 of the Official Secrets Act, Section 43 and 66 of the Information Technology Act, Sections 378, 379, 463, 465, 470, 471 and 5050 of IPC on the basis of the complaint lodged by Smt T.Malathi, IAS, Principal Secretary to Government Home (SC) Department, Secretariat, Chennai-9.

Fig. 1. Example of annotated document.

TABLE VIII. NAMED ENTITY EXTRACTED FROM SAMPLE PARAGRAPH

Name Entity	Text
ACT	CrPC, Information Technology Act, Official Secret Act, IPC
LEGAL	Section 482, Section 5, Section 43 and 66, Sections 378, 379, 463, 465, 470, 471 and 5050.
PERSON	Smt. T. Malathi
LOC	Chennai
DATE	17.07.2008
COURT	Fast track court

4) Manual annotation process: The annotation for the judgment text was done at the sentence level, therefore each judgment sentence was given separately from the annota- tion without document-level context. In the event that extra background information is required for annotation, the whole judgment text is also available. The indiankannon URL was used to obtain the whole judgment text.

To Label and annotate data we have use the open-source data labelling tool, i.e., Label Studio. After importing your data, you can start labelling and annotating your data. Fig. 2 conceptualised name entity recognition using machine learning algorithm and manual annotation.

a) Open a project in Label Studio and optionally.

b) Click Label All Tasks to start labelling.

c) Use keyboard shortcuts or your mouse to label the data and submit your annotations.

d) Follow the project instructions for labelling and deciding whether to skip tasks.

e) Click the project name to return to the data manager.

5) Annotated corpus statistics: In this paper, a dataset of annotated judgment text with seven entities has been created. A dataset of almost 5000 Indian judicial judgment sentences with seven entities has been created. The Table IX lists the number of documents, sentences, and tokens in the annotated corpus as well as other general statistics.

TABLE IX. ANNOTATED CORPUS STATISTICS

No. of Documents	30
No. of Sentences	~5196
Average no. of sentences per document	173
No. of tokens (without stop words)	63155
Annotated tokens	~5286

B. NER Model

Several well-known NER model architectures were explored to identify legal named entities in judgment papers. Initially, spacy's pre-trained NER model was used to implement Legal NER. Two of spacy's pre-trained pipelines, namely en_core_web_trf and en_core_sci_sm, were integrated with unique rules created specifically for the legal domain to improve the accuracy of predictions.

During the training phase, the model's predictions were iteratively compared to the reference annotations to calculate the gradient of the loss as shown in Fig. 3. Backpropagation was then used to determine the gradient of the weights using the gradient of the loss. This approach enabled us to determine how to adjust the weight values so that the model's predictions gradually resembled the reference labels, hence enhancing the model's accuracy.

To make sure that our Legal NER model was optimized for the needs of legal named entity identification, we used a strict and systematic methodology. Our algorithm is capable of accurately identifying many different types of legal entities, such as court names and legal terms.

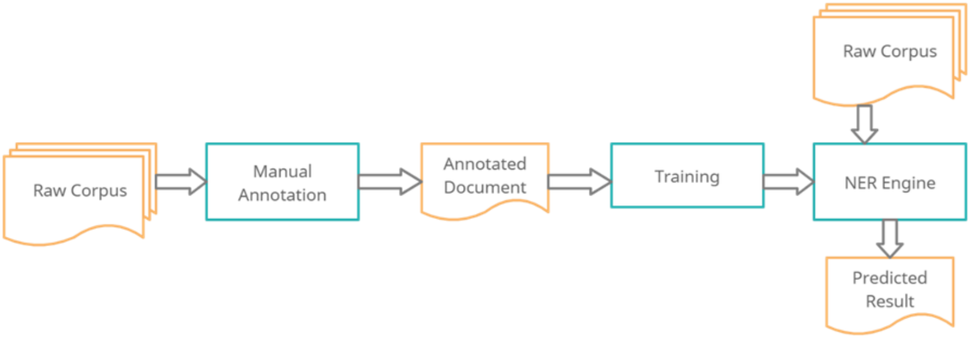


Fig. 2. Manual annotated data and NER system.

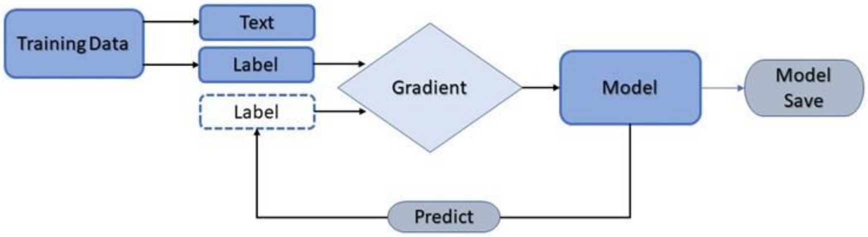


Fig. 3. Spacy pretrained pipeline.

C. Results

Various evaluation matrices were used, such as the F1 score, recall, and precision, to evaluate the model's efficiency. These metrics provide important information about how well the model can identify and categorize data points. The F1 score represents the harmonic mean of accuracy and recall, where recall represents the proportion of true positive values that the model correctly identified and precision represents the percentage of true positive values that the model correctly recognized. A variety of measurements can be utilized to better understand the model's advantages and disadvantages, which will help in deciding how to enhance its performance.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (3)$$

where,

TP = True Positive   FP = False Positive

TN = True Negative   FN = False Negative

NER model's aggregate F1 score of 0.62 indicates that the quality of our training data is higher than average. Precision, Recall, and F1scores on judgment sentences are used to assess the model. Table X displays the results of various tests and experiments.

TABLE X. RESULT OF SPACY TRAINED PIPELINE

Spacy Trained Pipeline	Precision	Recall	F1 Score
en_core_web_trf	0.6	0.41	0.48
en_core_sci_sm	0.51	0.4	0.45

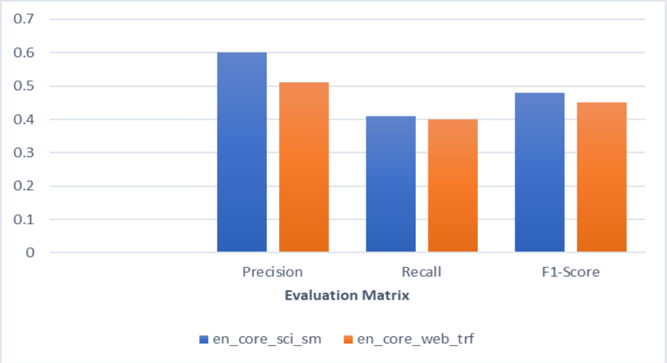


Fig. 4. Analysis of results of trained spacy models.

The Fig. 4 shows the comparison of the performance of the model on individual entities. Since the dataset is completely unbalanced, precision, recall, and F1 score have been calculated for comparison. Precision and recall are defined in terms of true positive, false positive and false negatives, whereas the F1 score is defined as the harmonic mean of precision and recall. The weighted average of precision, recall, and F1 score for spacy en_core_web_trf pipeline are 0.60, 0.41, and 0.48 respectively, and for spacy en_core_sci_sm are 0.51, 0.40, and 0.45 respectively. Good results have been obtained from the experiments and evaluations, and the Legal NER model can be a valuable tool for a variety of legal applications such as legal information retrieval, document summarization, and more.

VI. DISCUSSION

In the legal domain, NER is typically used for tasks such as document classification, contract analysis, and case law research [27]. The accuracy of NER is crucial in the legal domain, as incorrect recognition of entities can lead to incorrect legal decisions [28].

There are several challenges in NER for the legal domain compared to other domains [29]. Firstly, the language used in legal documents is often complex and technical, which might be difficult to identify with traditional NER models. Secondly, legal named entities can have multiple forms and variations, such as acronyms, abbreviations, and synonyms, requiring NER systems to have a comprehensive understanding of legal terminology. To solve this issue, NER models in the legal domain are frequently fine-tuned using massive annotated legal corpora, which can increase the accuracy of legal entity recognition [30].

Another challenge in NER in the legal domain is the presence of named entities with several mentions, such as the names of legal parties. These entities may be referred to by multiple names or titles in different places of the document, making proper identification difficult. To overcome this problem, NER models in the legal sector typically include named entity disambiguation approaches, which assist in the identification and resolution of ambiguity in named entities.

Despite these challenges, NER has proven to be a valuable tool in the legal domain. By automating the process of identifying named entities [31], NER can significantly reduce the time and effort required for legal research and analysis. This can result in increased efficiency and productivity for legal professionals, as well as improved accuracy and consistency in the analysis of legal data. Overall, NER in the legal domain is a critical tool for facilitating legal research, analysis, and decision-making. With advances in machine learning and NLP techniques [32], NER models in the legal domain are becoming more accurate and efficient, helping to make the legal process faster and more effective.

Name Entity Recognition has great potential to improve the process of legal research and analysis, but it faces significant challenges in the legal domain due to the complexity and technical nature of legal language [33]. Further development and refinement of NER systems for the legal domain will likely result in even greater benefits for legal professionals in the future. Once these entities have been extracted and tagged, they can be used for research and analysis of legal texts. Furthermore, policy-making can be informed by the knowledge gained by Legal NER. Overall, the use of LNER in legal research and text analysis can enhance legal research, inform policy decisions, and result in more efficient and fair legal systems.

VII. Conclusion and Future Work

In this paper, a corpus of Indian judgment papers is presented that is annotated with 7 distinct types of entities and can be used to identify legal named entities. In order to create the annotated dataset, a variety of annotation tools were reviewed. 30 court documents that are available publicly were manually annotated. With the dataset, a spacy model was also trained utilizing the trained NER pipelines en_core_sci_sm and en_core_web_trf. The model displays an F1-score of almost 60%, indicating that the dataset has better quality. It is believed that the dataset will be useful for additional NLP tasks on Indian judicial material, such as relationship extraction, knowledge graph modelling, extractive summarization, etc.

In terms of future work, the author will explore approaches for extending and further optimizing the dataset. They will also perform additional experiments with more recent state-of-the-art approaches. The researchers plan to produce a CSV version of the dataset, which will simplify the data format, enhance compatibility, facilitate data pre-processing, and enable data analysis.

References

[1] J. Marrero, S. Urbano, J. S. nchez Cuadrado, J. M. Morato, and G. mez Berb´ ıs, “Named entity recognition: fallacies, challenges and opportunities,” *Computer Standards & Interfaces*, vol. 35, no. 5, pp. 482–489, 2013.

[2] I. Mugisha and Paik, “Comparison of Neural Language Modeling Pipelines for Outcome Prediction from Unstructured Medical Text Notes,” *IEEE Access*, vol. 10, pp. 16–489, 2022.

[3] Han, Xu, Chee Keong Kwoh, and Jung-jae Kim. "Clustering based active learning for biomedical named entity recognition." In 2016 International joint conference on neural networks (IJCNN), pp. 1253-1260. IEEE, 2016.

[4] U. Neves and Leser, “A survey on annotation tools for the biomedical literature,” Briefings in bioinformatics, vol. 15, no. 2, pp. 327–340, 2014.

[5] Neudecker, “An open corpus for named entity recognition in historic newspapers,” Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), pp. 4348–4352, 2016.

[6] J. M. Steinkamp, W. Bala, A. Sharma, and J. J. Kantrowitz, “Task definition, annotated dataset, and supervised natural language processing models for symptom extraction from unstructured clinical notes,” Journal of biomedical informatics, vol. 102, pp. 103–354, 2020.

[7] J. Rodriguez, A. Diego, A. Caldwell, and Liu, “Transfer learning for entity recognition of novel classes,” Proceedings of the 27th International Conference on Computational Linguistics, pp. 1974–1985, 2018.

[8] K. Bontcheva, H. Cunningham, I. Roberts, and V. Tablan, “Web-based collaborative corpus annotation: Requirements and a framework implementation New Challenges for NLP Frameworks,” pp. 20–27, 2010.

[9] A. Brandsen, S. Verberne, K. Lambers, M. Wansleeben, N. Calzolari, F. B. chet, and P. Blache, “Creating a dataset for named entity recognition in the archaeology domain,” The European Language Resources Association, pp. 4573–4577, 2020.

[10] Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, Serena Villata. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker. ICAIL-2017 - 16th International Conference on Artificial Intelligence and Law, Jun 2017, Londres, United Kingdom. pp.22. ffhal-01541446.

[11] S. Tripathi, H. Prakash, and Rai, “SimNER-an accurate and faster algorithm for named entity recognition,” Second International Conference on Advances in Computing, Control and Communication Technology (IAC3T), pp. 115–119, 2018.

[12] E. F. Tjong, K. Sang, and F. D. Meulder, “Introduction to the CoNLL-2003 Shared Task: Language Independent Named Entity Recognition,” Proceedings of the Seventh Conference on Natural Language Learning, 2003.

[13] B. Glaser, F. Waltl, and Matthes, “Named entity recognition, extraction, and linking in German legal contracts,” IRIS: Internationals Rechtsinformatik Symposium, pp. 325–334, 2018.

[14] F. Dernoncourt, J. Y. Lee, and P. Szolovits, “NeuroNER: an easy-to-use program for named-entity recognition based on neural networks,” 2017.

[15] T. Green, D. Maynard, and C. Lin, “Development of a benchmark corpus to support entity recognition in job descriptions,” Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 1201–1208, 2022.

[16] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186, 2019.

[17] J. Li, Q. Wei, O. Ghiasvand, M. Chen, V. Lobanov, C. Weng, and H. Xu, “Study of Pre-trained Language Models for Named Entity Recognition in Clinical Trial.”

[18] E. Leitner, G. Rehm, and J. Moreno-Schneider, “A dataset of German legal documents for named entity recognition,” 2020.

[19] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Karn, and V. Raghavan, “Named Entity Recognition in Indian court judgments,” 2022.

[20] K. Okoli and Schabram, “A guide to conducting a systematic literature review of information systems research,” 2010.

[21] S. Yadav and Bethard, “A survey on recent advances in named entity recognition from deep learning models,” 2019.

[22] P. Stenetorp, S. Pyysalo, G. Topic´, T. Ohta, S. Ananiadou, and J. Tsujii, “BRAT: a web-based tool for NLP- assisted text annotation,” Proceedings of the Demonstra- tions at the 13th Conference of the European Chapter, pp. 102–107, 2012.

[23] V. Sarnovsky´, N. M.-K. kova´, and Hrabovska´, “Annotated dataset for the fake news classification in Slovak language,” 2020 18th International Conference on Emerging eLearning Technologies and Applications (IC- ETA), pp. 574–579, 2020.

[24] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, and G. Gorrell, “GATE Teamware: a web-based, collaborative text annotation framework,” Language Resources and Evaluation, vol. 47, no. 4, pp. 1007–1029, 2013.

[25] T. Perry, “Lighttag: Text annotation platform,” 2021.

[26] J. B. Gillette, S. Khushal, Z. Shah, S. Tariq, Algamdi, Krstev, M. Ivan, B. Mishkovski, S. Mirchev, and G. Golubova, “Extracting Entities and Relations in Analyst Stock Ratings News,” 2022 IEEE International Conference on Big Data (Big Data), pp. 3315–3323, 2022.

[27] A. Barriere and Fouret, ““May I Check Again? -A simple but efficient way to generate and use contextual dictionaries for Named Entity Recognition. Application to French Legal Texts,” 2019.

[28] S. Paul, P. Goyal, and S. Ghosh, “LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 11–139, 2022.

[29] I. Angelidis, M. Chalkidis, and Koubarakis, “Named Entity Recognition, Linking and Generation for Greek Legislation,” JURIX, pp. 1–10, 2018.

[30] S. Paul, A. Mandal, P. Goyal, and S. Ghosh, “Pre-training transformers on Indian legal text,” 2022.

[31] Chiu, Jason PC, and Eric Nichols., “Named entity recognition with bidirectional LSTM-CNNs,” Transactions of the association for computational linguistics, vol. 4, pp. 357–370, 2016.

[32] S. Yadav and Bethard, “A survey on recent advances in named entity recognition from deep learning models,” 2019.

[33] Lison, Pierre, Aliaksandr Hubin, Jeremy Barnes, and Samia Touileb. "Named entity recognition without labelled data: A weak supervision approach." arXiv preprint arXiv:2004.14723 (2020).

Research on the Model of Preventing Corporate Financial Fraud under the Combination of Deep Learning and SHAP

Yanzhao Wang*
Henan Institute of Economics and Trade, College of Finance,
Zhengzhou, 450000, China

***Abstract*—Preventing financial fraud in listed companies is conducive to improving the healthy development of China's accounting industry and the securities market, is conducive to promoting the improvement of the internal control system of China's enterprises, and is conducive to promoting stability. Based on the combination of SHAP (Shapley Additive explanation), a prediction and identification model should be built to determine the possibility of financial fraud and the risk of fraud for the company. The research model has effectively improved the identification accuracy of financial fraud in listed companies, and the research model has effectively dealt with the gray sample problem that is common in the forecasting model through the LOF algorithm and the IF algorithm. When conducting comparative experiments on the models, the overall accuracy rate of the research model is over 85%, the recall rate is 78.5%, the precision rate is 42%, the AUC reaches 0.896, the discrimination degree KS reaches 0.652, and the model stability PSI is 0.088, compared with traditional financial fraud Forecasting models FS model and CS model has a higher predictive effect. In the empirical analysis, selecting a company's fraud cases in 2020 can effectively analyze the characteristic contribution in the fraud process and the focus on fraud risks. The established model can effectively monitor the company's finance and prevent fraud.**

Keywords—Financial fraud; deep learning; ensemble algorithm; feature selection

I. Introduction

With the rapid development of China's social economy, listed companies also occupy the majority of the market economy, and the quality of financial information of listed companies has an important impact on the efficiency of the capital market. The frequent occurrence of financial fraud in listed companies has seriously hindered the healthy development of the capital market, leading to the weakening of the company's control structure, reducing the effectiveness of corporate governance, and deteriorating the quality of its audit function [1-3]. At present, traditional financial fraud prevention and prediction systems are gradually unable to meet the needs of the market economy, and financial fraud prediction models are not yet sound. Relevant institutions and enterprises are difficult to effectively evaluate financial fraud phenomena, resulting in difficulties in effectively improving the financial risks of listed companies. How to enable different institutions and relevant enterprises to avoid financial fraud is the key to improving social economy and improving the market system. XGBoost integrated learning algorithm can more effectively predict financial fraud, which is currently an excellent classification algorithm. Therefore, in order to more accurately predict the existence of fraud in enterprises, a model based on deep learning combined with SHAP to prevent financial fraud in enterprises was studied, enabling regulators to conduct regulatory early warning for listed companies, creating a good ecological environment for the healthy growth of the capital market [4-6]. The research proposed combining ratio scaling with XGBoost model. The financial fraud prediction model can be quantitatively evaluated, and the design of a scoring card can not only have high prediction accuracy, but also have a negative sample capture rate. Derivative screening of the financial fraud risk indicator system, dimensionality reduction processing features make the model easy to use, combining different corporate governance perspectives with financial data as indicator variables. Reduce the cost of manual investigation, reasonably assess audit risks, and determine a more efficient audit scope. Reasonably evaluate the potential value and future growth space of different enterprises. At the same time, the model can be used to evaluate the potential fraud risks of debt companies, providing a more favorable analysis of the security of borrowing funds.

II. Related Works

Financial fraud means that the managers of the enterprise cover up the real financial status and cash flow of the enterprise by changing or falsifying the accounting information, so as to bring huge economic benefits to the fraudsters. How to predict or detect financial fraud has been researched and analyzed by various scholars. HWA et al. constructed a thinking map based on the relationship between enterprises and audit firms, and used the method of feature extraction to build a thinking framework to analyze whether enterprises have financial fraud [7]. Houssou et al. use homogeneous and non-homogeneous Poisson processes to detect financial fraud in imbalanced datasets. Experimental results show that applying the model to financial datasets shows better predictive ability than baseline methods especially in the case of high data imbalance [8].

The XGBoost algorithm is an integrated learning method, which is widely used in all walks of life, and is also deeply researched and developed by different professionals. C Zhao et al. established a prediction model of speed estimation and distance between other vehicles during vehicle operation by mixing neural network and limit gradient, and found that the model can help drivers effectively predict the speed and distance of other vehicles when changing lanes in different scenarios, with good prediction effect [9]. W niu et al. integrated a gradient lifting algorithm into computer traffic monitoring to protect computer security and monitor hacker attacks. XGBoost is mainly used to distinguish whether there is malicious traffic. The performance of the established model can effectively distinguish between malware traffic and normal traffic from normal data sets in actual use, and the false alarm rate is less than 1% [10]. Bao et al. combined AdaBoost ensemble learning with under sampling data processing methods, and introduced a model evaluation index that is more suitable for fraud prediction tasks. Experimental results prove that the model is superior to the financial ratio-based regression model and single-core support vector machine model in predicting financial fraud [11].

To sum up, in the research on the prediction and detection of financial fraud, there is relatively more research on the detection of financial fraud. However, there is little research on the prediction of financial fraud and economic fraud. The SGBoost algorithm is widely used in engineering, Internet security, data set classification, and system modeling, but there are relatively few related studies on the application of the XGBoost algorithm to predict financial fraud. Therefore, research on deep learning combined with SHAP to prevent corporate financial fraud models, Using Benford’s law, LOF local anomaly detection, and isolated forest detection method to eliminate gray samples in the prediction model, establish a financial fraud prediction model for listed companies based on machine learning methods, and create a good and healthy ecological environment for the capital market.

III. Construction of Financial Fraud Prevention Model based on Deep Learning and SHAP

A. Fraud Prediction Model Index Construction

Financial fraud hides or changes the relevant financial status, operating conditions and capital flow of the enterprise through forgery, omission, fabrication, etc., which can bring huge economic benefits to the fraudsters. Therefore, choose the appropriate financial fraud early warning indicators to establish a suitable model foundation. In the process of selecting indicators, follow the principles of reference, systematization, operability, and loose before tightening to make the model have good versatility [12-14]. The premise of an efficient and accurate deep learning model is to select good financial fraud characteristic variables, and the index selection is shown in Fig. 1.

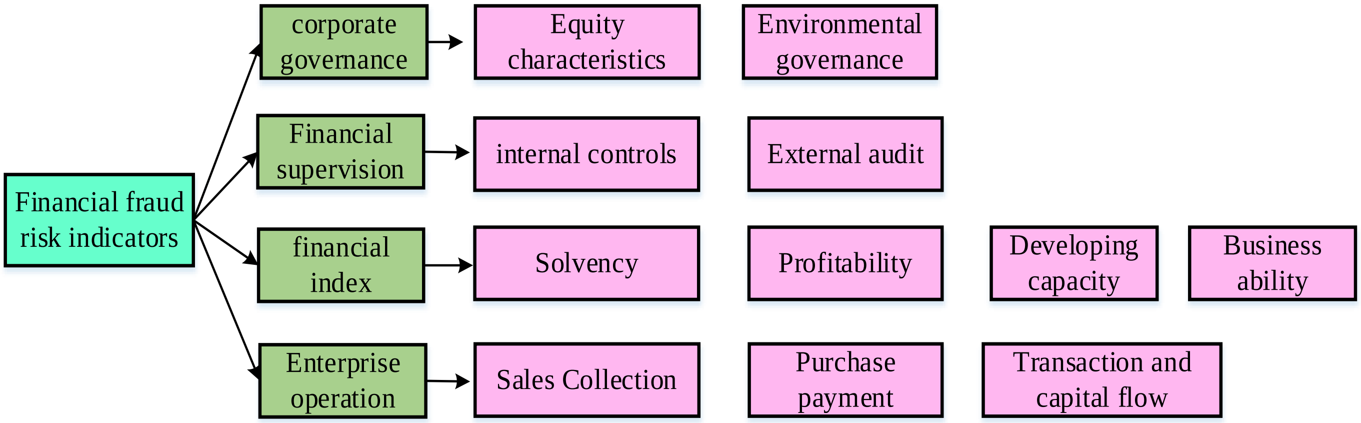


Fig. 1. Financial fraud risk indicator system.

Fig. 1 shows the financial fraud risk indicator system. Corporate governance indicators reflect the level of managers within the company and the degree of internal checks and balances. The secondary indicators are divided into equity characteristics and environmental governance. Financial and accounting supervision indicators divide supervision into internal control and external audit. Financial fraud is more likely to occur in companies that lack effective internal control systems. External audit indicators, such as firm size, auditors and other external factors, examine the factors affecting financial fraud. Financial indicators reflect the development of the company’s economy, solvency reflects an important indicator of the company’s long-term operation and development, and profitability is an indicator for evaluating the company’s created value and company value. Operating ability reflects the turnover of different assets of the company and the level of management. Development ability is the basis for measuring whether the company has good growth potential in the future. The cash flow cycle is a risk indicator that reflects whether the company has abnormal operations. The sales collection indicator reflects whether the company’s revenue is normal, the purchase payment indicator reflects whether the credit purchase and cash purchase at the purchasing end of the enterprise are normal, and the transaction and capital flow reflect whether the company has engaged in financial fraud through related transactions. After determining the financial fraud risk indicator system, research and select appropriate financial fraud samples. The financial fraud samples for this research are selected from the CSMAR China listed company financial annual report database and the violation information summary table of the violation event database, and select the handling documents issued by the regulatory agency. The data of illegal companies involving "fictitious profits" and "fictitious assets" between 2010 and 2021 are used as financial fraud samples in this paper. The financial fraud sample noun is divided into three different training sets before building the model: training samples, test samples, and out-of-time samples. The ratio of training samples to testing samples is 7:3. The out-of-time samples are the last samples of the time slice among all the samples. The training samples are used to train the model, while the test and out-of-time samples are utilized to evaluate the model's capability, stability, and generality [15]. In order to make the model more flexible and stable, and avoid the impact of extreme data values on the model, feature discretization is selected. The discretization formula is shown in formula [16-18] (1).

$$woe_i = \ln(\frac{p_{y_i}}{p_{n_i}}) \tag{1}$$

In formula (1), p_{y_i} is the ratio of fraudulent samples in the current group to all fraudulent samples in the sample, is the ratio p_{n_i} of all non-fraudulent samples in the current group to non-fraudulent samples in all samples, and woe is the difference between the two ratios, the difference Expressed logarithmically. Ensure the operational efficiency of the model and avoid overfitting, the overall features are screened. In the preliminary screening process, after determining the feature missing rate, the distribution difference of the sample labels with different values of the feature is measured. The calculation formula is shown in formula (2).

$$IV = \sum iv_i \tag{2}$$

In formula (2), iv_i the expression of is shown in formula (3).

$$iv_i = (p_{y_i} - p_{n_i}) woe_i \tag{3}$$

In formulas (2) and (3), it IV represents the amount of feature information, which can reflect the contribution of a single feature to label distinction. When the amount of feature information is too large, it means that the model simulation effect is excellent. The characteristic correlation calculation between two characteristic variables is measured by the Spearman correlation coefficient. The measurement formula is as follows.

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{4}$$

In formula (4), it n represents the quantity of data, which d_i is the difference between the order of two feature data. In feature correlation screening, feature indicators with a correlation greater than 0.7 are eliminated. The calculation formula for calculating the distribution difference of the same index on different data sets, measuring characteristics and model stability is shown in formula (5).

$$PSI = \sum_i (p^i_{t\arg et} - p^i_{base}) * \ln(\frac{p^i_{t\arg et}}{p^i_{base}}) \tag{5}$$

In formula (5), PSI is the characteristic stability, $p^i_{t\arg et}$ is the ratio of the samples in the first box of the target set to the i total samples, and is the total proportion of samples in the p^i_{base} first box of the basic distribution i. In the characteristic stable value, the characteristic items less than 0.02 are eliminated. After screening the information quality of the features, Use Chichi information criterion and Bayesian information criterion to select the characteristics of the model to ensure that the model has sufficient complexity and data set description. The calculation formula of AIC is as follows.

$$AIC = 2k - 2\ln L \tag{6}$$

The calculation formula of BIC is as follows.

$$BIC = 2\ln n - 2\ln L \tag{7}$$

In formula (6) and formula (7), k is the number of model parameters, n is the number of samples and L is the likelihood function. The fraudulent companies in the non-fraudulent samples are called "gray samples". Eliminating gray samples can ensure the quality of training data of the deep speech learning model and improve the reliability of the model; balance problem. The study uses Binford’s law, LOF local anomaly factor method and isolated forest algorithm (IF) to eliminate gray samples.

The calculation formula of probability distribution and correlation coefficient in Binford’s law is shown in formula (8).

$$r(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X) \cdot Var(Y)}} \tag{8}$$

In formula (8), $\mathrm{Cov(X,Y)}$ is the covariance, $Var(X)$ and $Var(Y)$ is the variance respectively. In Binford’s hypothesis test, the chi-square test is selected to test the degree of conformity of the numerical distribution of the observed samples. The calculation formula of statistics is shown in formula (9).

$$\chi_n^2 = \sum_{i=0}^{9} \frac{(F_n(i) - F_0(i))^2}{F_0(i)} \tag{9}$$

In formula (9), $F_n(i)$ is the actual observed value, and $F_0(i)$ is the theoretical distribution value of the law. The LOF anomaly detection method introduces local reachability density to measure the degree of sample anomaly, which can output anomaly scores very well and has strong interpretability. The calculation formula is shown in formula (10).

$$lrd_k(p) = \frac{1}{\frac{\sum_{o \in N_{k(p)}} reach\_dis_k(p,o)}{|N_k(p)|}} \quad (10)$$

In formula (10), p and o are two different sample points, and the reachable distance is $reach\_dis_k(p,o)$, which $N_k(p)$ means that the distance between the sample points is less than or equal to the k distance between the nearest point and the sample point. The local anomaly factor is calculated using the following formula.

$$LOF_k(p) = \frac{\frac{\sum_{o \in N_{k(p)}} lrd(o)}{|N_k(p)|}}{lrd(p)} \quad (11)$$

The IF algorithm is an integrated algorithm is based on the idea of random partitioning of space. The principle of a single isolated tree in the two-dimensional space of the IF algorithm is shown in Fig. 2.

The blue point in Fig. 2 means normal sample, and the outlier point is indicated by red. The straight lines that divide the space are randomly selected along the two coordinates respectively. The probability of the red dot being isolated before the blue dot is much greater than that of the blue dot. The probability of being isolated before the red point is uncertain, however, the presence of a single tree can increase the chance. Therefore, the combination of multiple isolated trees can enhance the stability of the model, leading to the final IF model. The final IF abnormal score is defined as formula (12).

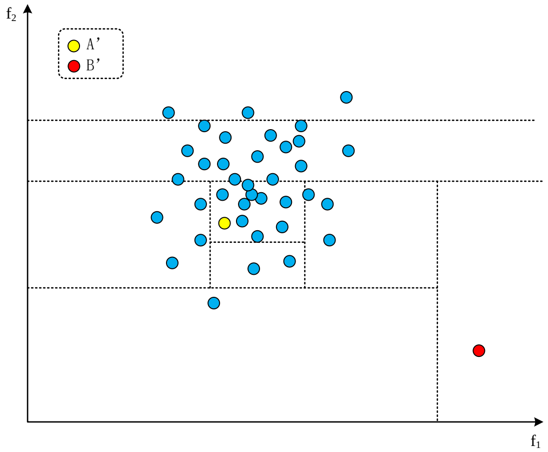


Fig. 2. Schematic diagram of IF algorithm.

$$Score(x_i) = 2^{\frac{E(h(x_i))}{C(n)}} \quad (12)$$

In the formula (12), it $E(h(x_i))$ represents the average path length of the blue point on the isolated tree, n is the number of training samples, and $C(n)$ is n the average path length of the binary tree trained by samples. After the gray sample is proposed, SMOTE is selected for oversampling. The basic principle of SMOTE is retracted as shown in the Fig. 3 below.

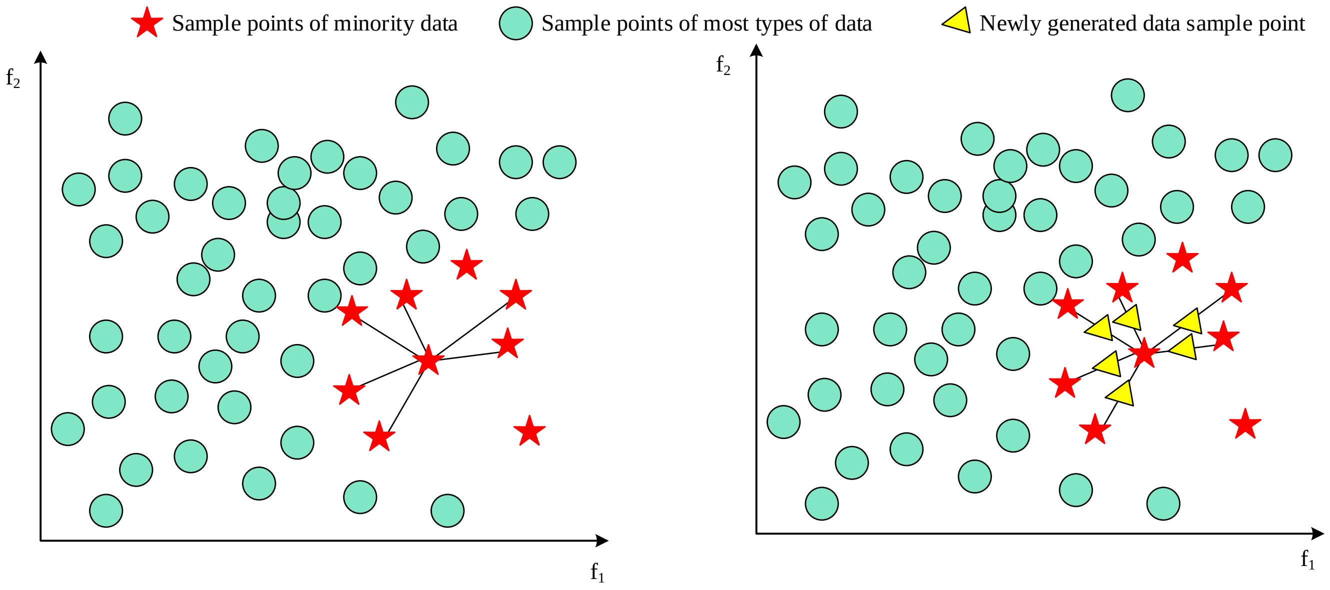

Sample points of minority data
Sample points of most types of data
Newly generated data sample point
f2
f1
f2
f1

Fig. 3. Schematic diagram of SMOTE algorithm.

SMOTE algorithm interpolates and synthesizes existing minority samples by analyzing minority samples, and then adds the samples to the dataset for training. The Light GBM algorithm was selected in this study to clean the samples. This was achieved by reducing the weight of samples with poor head and tail prediction results, so that they would not participate in the interpolation process of the SMOTE algorithm. The SMOTE algorithm has a high reliability for classification and oversamples the samples. Finally, the sampling results are combined.

B. Construction of Financial Anti-Fraud Model based on Deep Learning

After preprocessing the data, establish an anti-fraud model, and use the XGBoost model to improve the gradient boosting regression tree (GBRT) algorithm. The XGBoost model can support CART trees and linear classifiers at the same time. The study method is based on the forward distribution algorithm to achieve the integration of the additive model. The iterative process of the traditional GBRT algorithm is shown in formula (13).

$$\hat{y}^{(T)} = v\sum_{j=1}^{T} f_j(x;\Theta_j) = \hat{y}^{(T-1)} + vf_T(x;\Theta_T) \tag{13}$$

In formula (13), T is the number of basic regression trees, Θ_j is the corresponding regression tree structure, v is the scaling weight factor, $\hat{y}^{(T)}$ is the prediction result of the regression tree, and $f_j(x;\Theta_j)$ is the output result when the scaling weight is not considered.

The gradient boosting process of the XGBoost model is shown in Fig. 4.

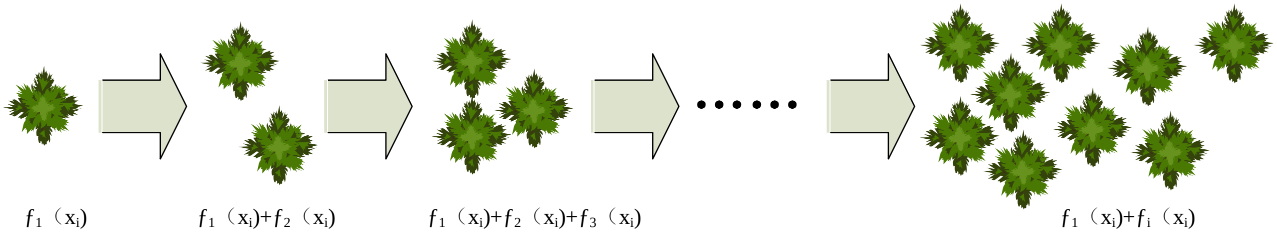


Fig. 4. XGBoost model principle.

The principle of a regression or classification model based on gradient lifting is shown in Fig. 4: First, establish a tree, and then gradually iterate. Each iteration process adds a tree, gradually forming a strong evaluator that integrates multiple tree models. The XGBoost model based on the tree model approximates the negative gradient of the model through the second-order Taylor expansion of the loss function, and learns it as the residual of the previous model. A higher learning weight is assigned to the samples with insufficient accuracy during the previous training process to improve the model accuracy, and serial iterations of multiple different models are implemented to gradually correct the deviation until the loss meets the convergence condition. The objective function after introducing the second-order Taylor expansion of the loss function is shown in Formula (14).

$$\hat{\Theta}_J \approx \arg\min_{\Theta_J}\left\{\sum_{i=1}^{N}\left[\hat{y}_j^{(j-1)} - y_i + vf_j(x_i;\Theta_j)\right]^2 + \Omega(\Theta_j)\right\} \tag{14}$$

In formula (14), N is the sample size, y is the minimum loss per additional node branch, and y_i is the given sample, and $\hat{y}^{(j-1)}$ has been determined. Therefore, $L(y_i,\hat{y}_j^{(j-1)})$ can be considered as a constant term to offset, and it is the regular term of the $\Omega(\Theta_j)$ th regression tree of j. After finding all the optimal single regression trees, the training is completed for enterprise fraud prevention prediction. The principle of $\Omega(\Theta_j)$ is shown in Formula (15).

$$\Omega(\Theta_j) = \gamma M_j + \frac{1}{2}\lambda\sum_{k=1}^{M_j}(w_k^{(j)})^2 \tag{15}$$

IV. Model Analysis and Empirical Application of Preventing Corporate Financial Fraud under Deep Learning

A. Model Prediction Effect Analysis

Effectively evaluate the actual put-to-use influence of the research model, the Fscore (FS) model and the Cscore (CS) model were selected to carry out a control experiment with the research model (RS), and the out-of-time samples were selected to test the model to test the stability of the model. In the control experiment, the application effects of different methods are compared by matrix. The matrix results are shown in the Table I.

TABLE I. Research Model Prediction Results

/	RS		CS		FS	
	Forecast		Forecast		Forecast	
Actual	Negative	Positive	Negative	Positive	Negative	Positive
FALSE	3146	513	2004	1763	1265	2574
TRUE	130	457	169	310	155	252

The research model correctly predicted 3146 cases of non-fraud samples and 457 cases of fraud samples, which is the model with the most correct prediction rate among the three models. The CS model correctly predicted only 2004 cases of non-fraud samples. Only 310 fraud samples were predicted, the FS model has the lowest prediction accuracy, only 1265 non-fraud samples were correctly predicted, and only 379 fraud samples were correctly predicted. The overall accuracy of the research model reaches 85%, the overall accuracy of the CS model reaches 54.5%, the prediction accuracy of FS model is relatively lower, reaching 36%. When analyzing the confusion matrix, we need to pay attention to the accuracy of the model, but also need to comprehensively evaluate the recall rate and accuracy rate of the model. The recall rate indicates the proportion of all companies that the model correctly predicts fraudulent companies, and the precision rate reflects the credibility of the model. The recall rate-precision rate results are shown in Fig. 5.

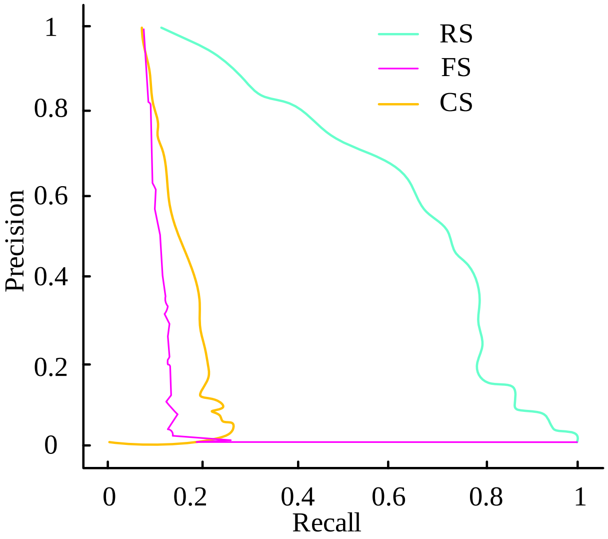


Fig. 5. PR curves of three models.

From the recall rate-precision rate result graph, it can be seen that the PR curve of the research model is the best, and the AP area is the largest, reaching 0.64. The precision rate and recall rate of the research model on samples out of time are higher than those of the CS model and the FS model. The indicators are superior, the AP area of the CS model is 0.16, and the AP area of the FS model is the smallest, which is 0.11, and the FS model is a commonly used forecasting model in the western capital market, and it has the worst forecasting effect on Chinese companies. The ROC curve can simply and intuitively observe the accuracy of different experimental models and make judgments through illustrations. This curve can accurately reflect the internal specificity relationship of the model and is a comprehensive representation to ensure the accuracy of the model. The results of the ROC experiment are shown in Fig. 6.

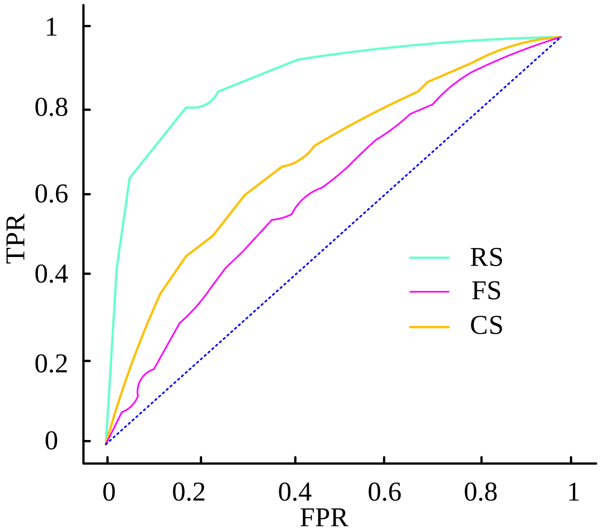
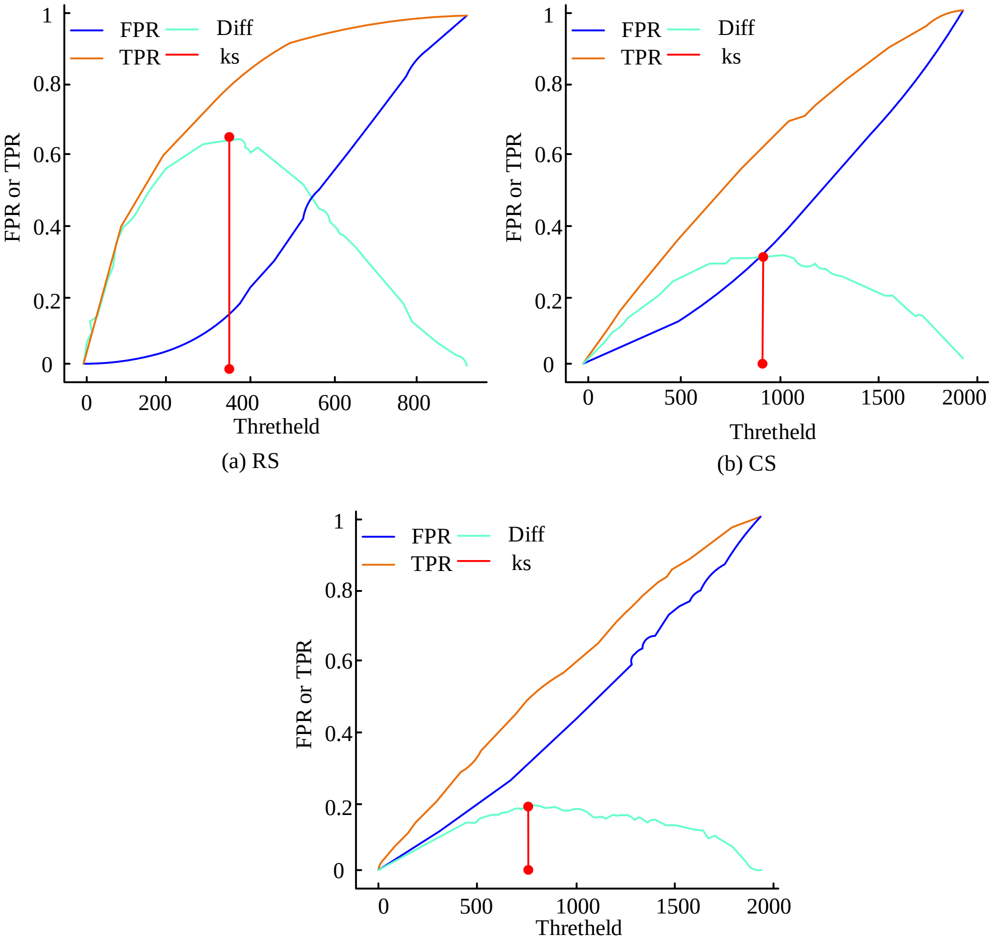


Fig. 6. ROC curves of different models.

When the area under the curve of the pure random classifier is 0.5, the larger the area under the ROC curve, the more realistic the fitting effect of the model. The research model's ROC curve performs best among the three models, with the largest area under the curve reaching 0.9. The ROC curve of the CS model is lower than that of the research model, and the area under the curve is only 0.7. The FS model had the lowest curve magnitude and the smallest area under the curve, reaching a value of 0.62. Combining Fig. 5 and Fig. 6, the research model can achieve a high accuracy rate when predicting fraud samples. The KS curve reflects the distinguishing ability of the model. After the model predicts the scores of all samples, it is divided into different parts according to TPR and FPR, and the distribution of the scores of different sample groups is tested by KS statistics. The KS curves of the three models are shown in Fig. 7.



(a) RS

(b) CS

(c) FS

Fig. 7. KS curves of three models.

Fig. 7 shows the comparison of KS curves for different models. The abscissa is arranged in descending order with thresholds of 1-0, and the ordinate is the difference between TPR and FPR under different thresholds. The higher the KS value is, the stronger the distinguishing ability of the model is. However, if it is too large, there is the problem of over-fitting. It can be seen from the above figure that the KS value of the research model is 0.65, which is within the range of the ideal model and has excellent positive and negative sample differentiation ability. Although the KS value of the CS model is in the ideal state, it is 0.35 lower than the KS value of the research model. The FS model has the worst ability to distinguish positive and negative samples among the three samples, only 0.18, and has a lower prediction effect on Chinese companies. The out-of-time sample test results of the three models combined with different evaluation indicators are summarized in Table II.

TABLE II. COMPREHENSIVE TEST RESULTS OF DIFFERENT MODELS

Model	**Accuracy**	**AUC**	**Recall**	**Precision**	**AP**	**KS**	**PSI**
RS	0.859	0.896	0.785	0.422	0.641	0.6 52	0.088
CS	0.545	0.697	0.564	0.143	0.159	0.30 5	0.115
FS	0.357	0.619	0.564	0.102	0.11 5	0.18 2	0.116

From the comprehensive test results in Table II, the accuracy of the research model reaches 0.859, the accuracy of the CS model is only 0.545, and the accuracy of the FS model is only 0.357. In AUC, the research model reaches 0.896, while the AUC of CS and FS Both are lower than the research model, reaching 0.697 and 0.619 respectively. In terms of recall rate, the recall rate of the research model reaches 0.785, the recall rate of the CS model is only 0.697, and the recall rate of the FS model is only 0.619. In terms of accuracy, the two models compared in the experiment are lower than the research model, the research model reaches 0.422, the CS model is only 0.142, and the FS model is only 0.102. In the PR curve comparison, the research model has the highest AP among the three models, which is 0.641, the CS model is only 0.159, and the FS model is only 0.115. Also in the KS value comparison, the research model reached 0.652, which has a good discrimination ability, the KS value of the CS model is only 0.305, and the FS model is only 0.182. PSI measures the distribution difference between test samples and model training samples. When PSI is less than 0.1, it can be considered that the model stability is very high. When PSI is 0.1-0.2, the stability of the model is average. When PSI is greater than 0.2, the model is stable. It can be seen that the PSI of the research model is the lowest among the three models, only 0.0883, and the stability is the highest, while the PSI of the CS model and the FS model are both higher than 0.1, and the stability is average, so the research model has a strong generalization ability.

B. Practical Application and Analysis of the Model

In the practical application and analysis of the research model, the SHAP method is used to explain the research model by using the SHAP library of Python, and a real fraud case in the training set is selected for analysis and prediction. The prediction results of the research model can be visualized with a single-sample SHAP graph. In the SHAP diagram, blue indicates that the contribution of the feature is negative, and orange indicates that the contribution of the feature is positive. The longer the orange color, the higher the probability of the predicted result being fraud. The specific results are shown in Fig. 8.

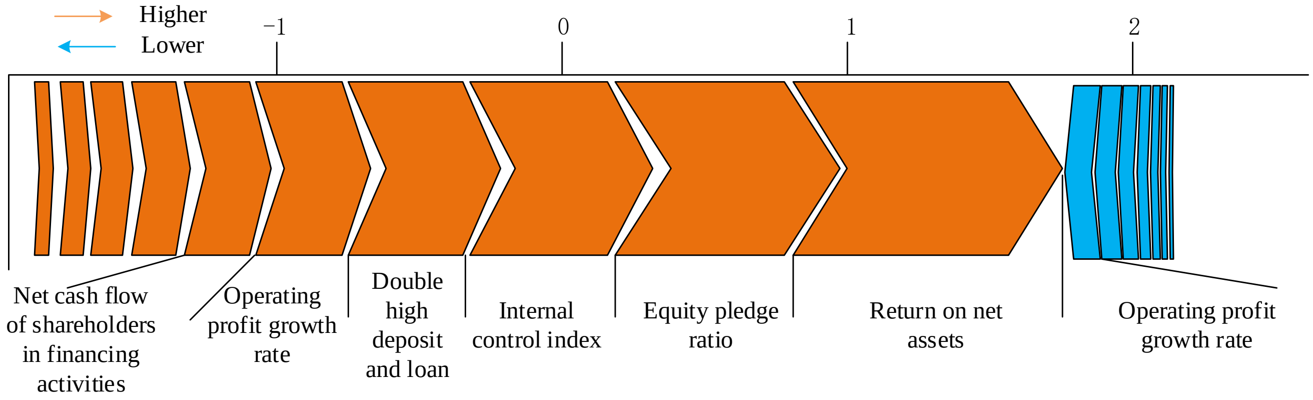


Fig. 8. SHAP chart of X enterprise’s research model forecast in 2020.

Fig. 8 shows the prediction results of the financial fraud scorecard. To improve the usability of the research and prediction model, a credit scoring mechanism is introduced into the model, and the model output is normalized through a unified scoring map to make the financial fraud risk readable. Among them, the high-risk score is ranked according to the feature contribution degree from high to low. To analyze the degree of dependence on related transactions, etc. In the prediction results of Company X in 2020, the important contributing variables are the proportion of equity pledges, internal control index, and high deposit and loan ratios, which reflect that Company X has risks in funding sources, risks in financial and accounting supervision, and abnormal business operations. The annual financial report disclosed by Company X contains major false information. First, there is a false increase in deposits. When there is a huge amount of monetary funds in the financial statements, there is a shortage of funds at the same time, and there is a phenomenon of high deposits and loans. Secondly, the forged business certificates lead to an inflated rate of return on net assets; in addition, part of the funds is transferred to the accounts of related parties for stock transactions, reflecting the illusion of a closed-loop capital flow. In addition, in terms of financial and accounting supervision, in the annual report, Company X reported that the company had major deficiencies in internal control, and its internal control audit agency maintained a negative audit opinion on the company’s internal control in 2020 [19-22].

V. CONCLUSION

With the development of the social economy, the quality of financial information in market information is more and more important to the efficiency of the market economy, but the problem of financial fraud is the most serious problem affecting the social economy. How to accurately predict financial fraud for listed companies is becoming more and more important. In this study, the integrated learning in deep learning is used to construct a risk warning model with more complete identification, more accurate and more robust, to score the financial fraud risk of listed companies, and to judge the prediction model of the company’s financial fraud possibility. The results show that the research model effectively solves the gray sample problem commonly faced when machine learning is applied to financial fraud identification research. The research model can improve the prediction effect of the model prediction. The overall accuracy rate of the research model is over 85%, and the recall rate is 78.5%. The accuracy rate reached 42%, AUC reached 0.896, the discrimination KS reached 0.652, and the model stability PSI was 0.088. Compared with the traditional financial fraud prediction models FS model and CS model, it has a higher prediction effect. In the empirical analysis, choose a certain company. The analysis of fraud cases in 2020 can effectively analyze the characteristic contribution in the fraud process and the focus of fraud risks. The research model is more suitable for the financial fraud prediction of listed companies in my country. However, in view of the availability of data, the research model does not sufficiently screen text analysis variables, there may be missing variables, and no specific analysis is carried out for the characteristics of different industries. The model also has a certain room for improvement, hoping to provide direction for future research.

REFERENCES

[1] H. Xia, H. Ma, P. Cheng. PE-EDD: An efficient peer-effect-based financial fraud detection approach in publicly traded China firms. CAAI Transactions on Intelligence Technology, 2022, 7(3):469-480.

[2] L. Liao, G. Chen, D. Zheng. Corporate Social Responsibility and Financial Fraud: Evidence from China. Accounting and Finance, 2019, 59(5):3133-3169

[3] Y. Jiang, Y. Zhao. Financial fraud contagion through board interlocks: the contingency of status. Management Decision, 2020,58(2):280-294.

[4] M. Wang, W. Zhao, W. Zhang, "Can Reform of Information Disclosure by an Exchange Restrain Corporate Fraud? Evidence from China." Asia-Pacific journal of financial studies, 2022,51(2):223-255.

[5] Z. M. Sanusi, A. Hudayati, T. K. Nisa. "Financial Pressure and Related Party Transactions on Financial Statements Fraud: Fraud Triangle Perspective." International Journal of Business and Emerging Markets, 2022, 14(2):213-230.

[6] R. Cao, G. Liu, Y. Xie, C. Jiang. "Two-Level Attention Model of Representation Learning for Fraud Detection." IEEE transactions on computational social systems, 2021,8(6):1291-1301.

[7] B. Hwa, C. Yca, D. Jl, "XZA Envelope. Financial fraud risk analysis based on audit information knowledge graph." Procedia Computer Science, 2022, 199:780-787.

[8] R. Houssou, J. Bovay, S. Robert. "Adaptive Financial Fraud Detection in Imbalanced Data with Time-Varying Poisson Processes." Journal of Financial Risk Management, 2019, 08(4):286-304.

[9] C. Zhao, X. Zhao, Z. Li, Q. Zhang. "XGBoost-DNN Mixed Model for Predicting Driver's Estimation on the Relative Motion States during Lane-Changing Decisions: A Real Driving Study on the Highway." Sustainability, 2022, 14(11):1-23.

[10] W. Niu, T. Li, X. Zhang, T. Hu, H. Wu. "Using XGBoost to Discover Infected Hosts Based on HTTP Traffic." Security and Communication Networks, 2019, 2019(1):1-11.

[11] Y. Bao, B. Ke, B. Li, Y. J. Yu, J. Zhang. "Detecting accounting fraud in publicly traded US firms using a machine learning approach." Journal of Accounting Research, 2020, 58(1): 199-235.

[12] R. H. Davidson. "Who did it matter: Executive equity compensation and financial reporting fraud." Journal of accounting & economics, 2022,73(2/3):101453.1-101453.24.

[13] A. Ys, B. Cg, A. Hl, B. JC, C. YG, XQ A. "Financial Feature Embedding with Knowledge Representation Learning for Financial Statement Fraud Detection." Procedia Computer Science, 2021, 187:420-425.

[14] A. Lotfi, M. Salehi, M. L. Dashtbayaz. "The effect of intellectual capital on fraud in financial statements." TQM Journal, 2022,34(4):651-674.

[15] A. Kumar, G. S. Mishra, P. Nand, M.S. Chahar, S.K. Mahto. "Financial Fraud Detection in Plastic Payment Cards using Isolation Forest Algorithm." International Journal of Innovative Technology and Exploring Engineering, 2021, 10(8):132-136.

[16] LaliSransrdjan.lalic.efb@gmail.comJoviieljanazeljana.jovicic@ef.unibl.orgBonjakoviTanjabosnjakovict@gmail.comUniverzitet u Istonom SarajevuEkonomski fakultet Banja LukaPoreska Uprava Republike PJ Bijeljina. The most common examples of financial fraud in Bosnia and Herzegovina: A practical insight. Journal of Forensic Accounting Profession, 2021, 1(2):80-88.

[17] Gepp A , Kumar K , Bhattacharya S . Lifting the numbers game: identifying key input variables and a best erforming model to detect financial statement fraud. Accounting and Finance, 2021, 61:4601-4638.

[18] Zhou H, Sun G, Fu S, Internet Financial Fraud Detection Based on a Distributed Big Data Approach With Node2vec. IEEE Access, 2021, PP(99):1-1.

[19] Swa B, Jl C, Xz A, Envelope MLA. Analysis of financial fraud based on manager knowledge graph. Procedia Computer Science, 2022, 199:773-779.

[20] Geng X., Yang D. Intelligent Prediction Mathematical Model of Industrial Financial Fraud Based on Data Mining. Hindawi Limited, 2021,2021(34):1-8.

[21] Verykios V S , Stavropoulos E C , Zorkadis V , et al. Sensitive data hiding in financial anti-fraud process. International journal of electronic governance, 2022,14(1/2):7-27.

[22] Ys A, Cg B, Hl A, JC B, YG C, XQ A. Financial Feature Embedding with Knowledge Representation Learning for Financial Statement Fraud Detection. Procedia Computer Science, 2021, 187:420-425.

Conjugate Symmetric Data Transmission Control Method based on Machine Learning

Yao Wang
Chongqing City Vocational College, College of Creative Design,
Chongqing, 402160, China

***Abstract*—In conjugate symmetric data transmission, due to insufficient judgment of congestion during transmission, the amount of data is large and the transmission rate is low. In order to improve the data transmission rate, a conjugate symmetric data transmission control method based on machine learning is designed. Firstly, the data to be transmitted is tracked and determined, and then conjugate symmetric data fusion is completed according to the calculation result of the best tracking signal. According to the fusion results, the framework of the conjugate symmetric data coding system is established, and the data coding is completed. The average congestion mark value is calculated by the machine learning method, and the congestion judgment of data transmission is completed. On the basis of congestion determination, the efficient transmission control of conjugate symmetric data is realized by specifying the conjugate symmetric data transmission protocol. Experimental results show that compared with traditional control methods, this control method has the advantages of a high delivery rate, low message transmission overhead and low data transmission delay. Compared to the traditional two-way path model, the scheduling method proposed in this study increases the transmission delivery rate by 5%, while reducing the transmission cost and delay by 0.7 cost index and 1.1 min delay, respectively. In comparison with the performance of accurate error tracking equalization, the transmission delivery rate of the research method increased by 21%, and in transmission cost index and delay analysis, it also decreased by 3.1 and 2.9 minutes. Based on the above performance comparison analysis, it can be concluded that the machine learning method has more superior transmission control performance.**

Keywords—Machine learning; conjugate symmetric data; transmission control; data coding; transmission delivery rate

I. INTRODUCTION

In the transmission process of conjugate symmetric data, the energy and storage resources of data nodes are limited, and the mobility of nodes and other practical application conditions are limited. At the same time, most of the nodes in the data transmission channel are in motion, and their physical positions often change. The data transmission link is not a very reliable wireless connection, which will also increase the difficulty of conjugate data transmission [1]. Due to the limited communication distance, low connection bandwidth, insufficient energy resources of nodes and other factors, the network connection is frequently interrupted, which cannot guarantee a constant end-to-end connection, which increases the difficulty of conjugate data transmission. At present, the research on data transmission control methods mainly includes two methods, one is based on the two-way path data transmission congestion control method, and the other is based on accurate error tracking equalization data transmission control method.

The congestion control method of data transmission based on a bidirectional path is mainly based on conjugate symmetric data transmission protocol. The congestion control mechanism can well avoid data loss caused by congestion. In this method, the connection is established through three handshakes, and then the congestion control algorithm is started. The application layer data is divided into data packets which are transmitted to the opposite end one by one. Each packet is composed of a byte sequence number and a message size, both of which are little in size. Each time the receiving terminal receives a data message, it will generate a corresponding response message to control the data transmission [2]. The data transmission control method is based on precise error tracking equalization, by constructing the historical record tracking process, adopting the way of threshold control to equalize the sensor data, and reducing the cost of fusion. Then the recursive flow is constructed to control the data transmission by packet tracking [3]. However, due to the large number of conjugate symmetric data that need to be transmitted at present, there are some problems such as transmission delay, low delivery rate and high transmission cost. Yi LA et al. used artificial bee colony algorithm to obtain the shortest path analysis of each cluster head node in the transmission of the Internet of Things. Experiments have shown that this algorithm can effectively reduce the amount of data that sensor nodes transmit to the sink through cluster head node fusion, improve data collection efficiency, energy consumption balance, and network reliability, and extend the network life cycle [4]. Mu et al. reconstructed routing problems such as the limitations of routing protocols under the condition of rapid data growth into a Markov decision process, combining deep reinforcement learning to solve the high blocking probability problem caused by increased data volume. Experiments show that this method can significantly reduce the probability of data congestion and improve network throughput [5]. In order to fully utilize multi-channel video transmission, Li H et al. proposed a joint optimization method for conversational high-definition video services, taking into account the connection between video coding and transmission. Experiments show that this method is superior to existing schemes in terms of data transmission and playback quality [6]. Pyeon D et al. proposed an efficient multipath pipeline transmission (EMP) to support low latency and high energy efficiency large data transmission under various network conditions. Experiments have shown that EMP outperforms existing protocols in terms of transmission time and energy efficiency, and can maintain improved EMP performance regardless of network environment (such as link quality, hop count, and network density) [7]. WANG Zhan Yu et al. designed a distributed communication platform based on P2P network technology to achieve concurrent path transmission on the internet. Experiments have shown that the effectiveness of this method has been verified by actual experiments on a simulation platform [8].

Therefore, a control method of conjugate symmetric data transmission based on machine learning is designed. Machine learning is an interdisciplinary subject, involving probability theory, statistics, approximation theory, convex analysis, algorithm complexity theory and other disciplines. The purpose of this paper is to study how a computer can simulate or realize human learning behavior, acquire new knowledge or skills, and reorganize the existing knowledge structure to improve its performance. Therefore, machine learning is applied to the transmission control of conjugate symmetric data. The conjugate symmetric data is used to provide some information to the learning part of the machine learning system. The machine learning part uses this information to modify the knowledge base, so as to improve the performance of the executive part of the system to complete the task. Firstly, the conjugate symmetric data are tracked, confirmed and fused, and the conjugate symmetric data coding architecture is constructed to code the data. Then, the data transmission congestion value is calculated by the machine learning method, which realizes the accurate judgment of data transmission congestion and provides the basis for the high-speed transmission of conjugate symmetric data. By formulating a conjugate symmetric data transmission protocol, the transmission control of conjugate symmetric data is realized, and the data loss caused by congestion is avoided.

This study conducted performance analysis experiments on delivery rate, transmission cost, and transmission delay for the constructed model. Firstly, the delivery rate compares the machine learning method used in this study with the traditional two-way path method and the accurate error tracking and balancing method. In the experiment, the model built in this study was in the delivery rate range of 0.76-0.83 after 6400 simulations, while the delivery rate of two-way paths was in the range of 0.77-0.71. Traditional accurate error tracking and balancing methods have the worst effect, with the highest delivery rate of only 0.52. In the second step of the experiment, the transmission cost was compared, and machine learning methods were also compared with traditional two-way path methods, as well as accurate error tracking and balancing methods. The results show that the cost index of the machine learning method is stable below 1.4, lower than other models. Finally, in the performance analysis of transmission delay, the delay of the machine learning method is controlled below 1.3 minutes, which is lower than other models. The experimental results can prove that the machine learning conjugate symmetric data transmission model constructed in this study has good transmission delivery rates, as well as excellent features such as low latency and low cost. The main purpose of this time is to provide a high-performance transmission scheduling control method for conjugate symmetric links of big data transmission. The proposed conjugate symmetric data transmission scheduling algorithm based on machine learning calculates data transmission congestion values, achieving accurate judgment of data transmission congestion. Therefore, the main contribution of this method is to simultaneously reduce information loss during transmission and increase the performance of transmitted information.

II. Conjugate Symmetric Data Fusion

Before controlling the transmission of the conjugate symmetric data, the conjugate symmetric data to be transmitted is first tracked and confirmed, because the signal strength of the conjugate symmetric data and the strength distribution proportion between the corresponding nodes need to be considered comprehensively in the node tracking. The tracking signal strength of the whole network node has a strong correlation proportion with the communication distance of the node. Therefore, by comprehensively judging the energy threshold and recursively measuring the threshold based on the intensity value, the best tracking signal can be obtained from the signal strength of other nodes. Therefore, the initial tracking of the node can be realized in this way. Assuming that the conjugate symmetric data to be transmitted is q, the preliminary tracking formula is as follows:

$$q = \frac{r}{W} / i \qquad (1)$$

In the formula, W is the joint quadratic probability function, r is the error estimation parameter, i represents the coordinates of the conjugate symmetric data nodes to be tested.

Due to the strong information interaction between the tracking node and the surrounding node, if a node is fully tracked, its tracking is related to the tracking connectivity factor of the node and the surrounding node. By comprehensively considering the tracking connectivity factor of the surrounding node and the node, the chaos inference based on the signal RF strength can effectively achieve the precision of the conjugate symmetric data node Make sure to track. Firstly, the nodes with the best energy surplus are selected as the tracking reference nodes, and the nodes to be tracked are tracked accurately through these nodes. While the nodes to be tracked are connected to other nodes through the broadcast mechanism, they are sorted according to the RF intensity threshold of other nodes. The expression is as follows:

$$y = \frac{qu}{t} * \sqrt{o} \qquad (2)$$

In the formula, t represents the maximum number of data tracking cycles of the tracking node, u is the probability that the node to be tracked can work normally at the next time, o is the coordinate accuracy of the current data to be transmitted.

On the basis of the above transmission data tracking, half of the output data in the conjugate symmetric sequence can be approximately 0. All the data to be transmitted are fused, and butterfly operation is carried out based on frequency decimation 2FFT. The two data with conjugate symmetric values are transformed into imaginary and real numbers at multiple levels, which reduces the computational complexity. The conjugate symmetric transmission data is fused by butterfly operation, and the calculation formula is as follows:

$$P = y \cdot F\left(M\right) / \frac{1}{C} \qquad (3)$$

In the formula, P represents all data to be transmitted, F represents the fusion error of the coordinates to be tracked, M represents data fusion parameters, C is the number of data traces.

According to the above process, the conjugate symmetric transmission data fusion is completed.

III. Conjugate Symmetric Data Coding

On the basis of the above-mentioned data fusion of conjugate symmetric transmission, the fused conjugate symmetric data is encoded, and the network coding technology is adopted. In order to realize the application of network coding technology in conjugate symmetric data transmission, a new strategy should be used to realize the anti-entropy transmission of data after two nodes establish a connection. Most existing link prediction studies originate from the resource transmission process of network evolution. Unfortunately, they ignore the impact of the topological stability of the structure around the path on the effectiveness of resource transmission on the path during the resource transmission process. The path topology stability in link prediction refers to the ability of the destination node to receive more resources from the originating node during resource transmission through the path. The stability of the structure around the path can reduce resource losses on the path, making both nodes receive more resources, and making both nodes more similar. The possibility of connecting edges between nodes is analyzed from the perspective of effective path topology stability. Therefore, this study will start with machine learning methods, analyze the transmission stability of conjugate symmetric data links, and conduct control optimization. First of all, we should establish the index of an encoded message rather than a simple message ID; secondly, we should increase the judgment of the value of the encoded message to ensure that the receiving node does not receive too many invalid messages, resulting in waste of storage resources and extra transmission load. The process of coding message anti-entropy transmission is shown in Fig. 1.

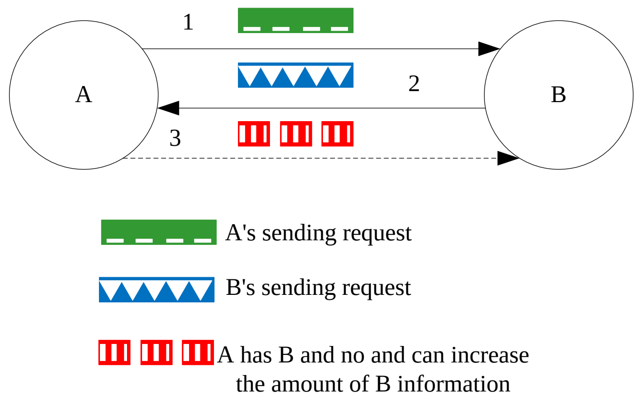


Fig. 1. Anti-entropy transmission process of encoded message.

The anti-entropy transmission process of encoded messages is shown in the figure above. When two nodes A and B establish a connection, in the first step, one node A sends the digest vector ACSV of the encoded message in its cache to another B; the second step node B performs an inverse operation on the digest vector of the encoded message in its own cache, and determines that the message is beneficial to increase the amount of information. Through the above process, the conjugate symmetric data is encoded. After the conjugated symmetric data is encoded, the amount of information contained in it differs from the source message. Multiple conjugate symmetric data encoded messages compare with multiple source messages. The amount of information is also different. The conjugate symmetric data encoding node evaluates the amount of information contained in the encoded message in its cache and determines whether it is necessary to encode a new message. This can greatly reduce the burden of node encoding calculation in the implementation of encoding. At the same time, it can avoid unnecessary "redundant" interference of encoded messages in the network, and can reduce the complexity of final decoding.

In order to evaluate the amount of information contained in the coded generation message and whether the new message contributes to the increase in the amount of information in the coded generation, it is proposed to use the concept of redundancy to characterize the number of coded sources messages and the number of generated coded messages in the coded generation. The relationship between them, and through this relationship to determine whether the newly received message can increase the amount of information in the current encoding generation, thereby determining whether it is necessary to encode the information, which can increase the effectiveness of the encoding result. From the perspective of whether the message in the encoding generation is sufficient to decode, consider the relationship between the dimension of the encoding generation (the number of messages in the generation) and the number of source messages (the number of messages originally generated in the encoded message contained in the generation before the encoding). The redundancy of an encoding generation is defined as the ratio between the dimension of the encoding generation and the number of source messages, and the calculation formula is:

$$w = P\frac{S}{D+Z} \quad (4)$$

In the formula, w represents the relationship between the number of initially generated and the number of messages contained in the encoded message, D represents the redundancy of an encoding generation, Z represents the number of messages, S represents the number of copies.

Redundancy can represent the relationship between the number of dimensions in a coding generation and the number of source messages. It can control the number of effective copies of messages in a coding generation. From a mathematical point of view, it is the relationship between the number of equations and the number of unknowns in a multivariate linear system. Mathematically, if the individual equations in the system of equations are linearly independent, only the unknown number of equations can be used to solve all the unknowns; and in the delay-tolerant network with network coding, the message is encoded through multiple paths. Sent to the target node, on one of the paths, the intermediate node does not need to obtain enough messages to reach the decodable condition. On the other hand, the message is stored with a new id after encoding. Under the infection routing strategy, another node will request this new message after seeing the message with this new id. Degree control can solve this problem. In order to realize the application of the network coding technology, this paper refers to the implementation method of the constrained protocol, and proposes an implementation method of inserting the "network coding layer" as the "overlay layer" between the application layer and the transmission layer. The framework of the conjugate symmetric data encoding system is shown in Fig. 2.

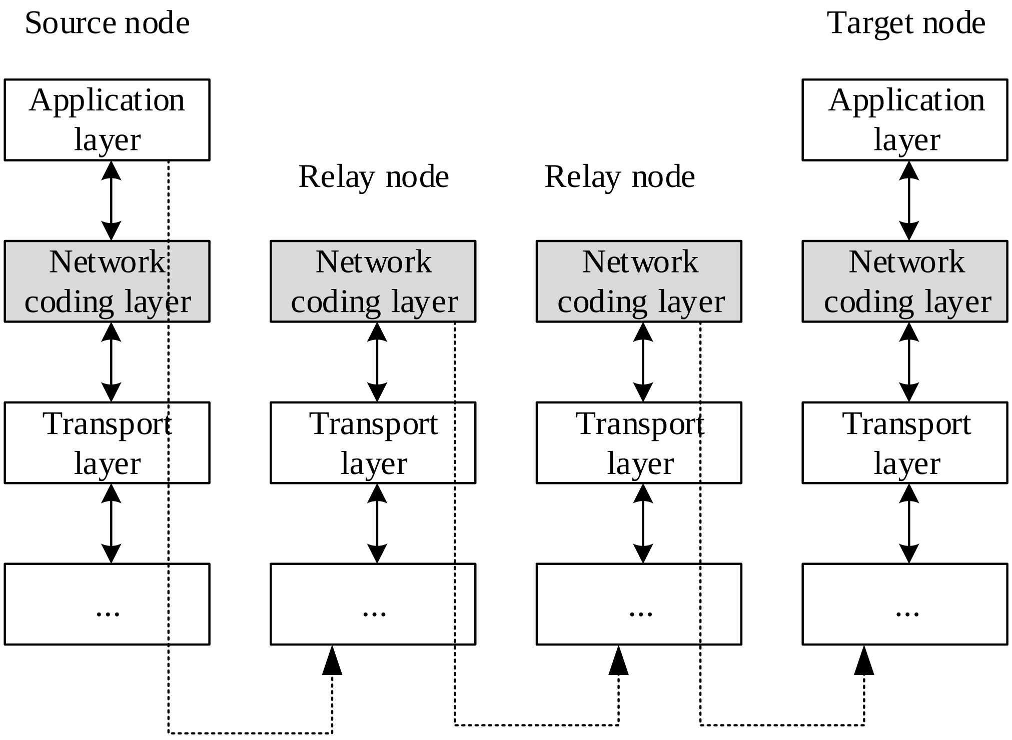


Fig. 2. Framework of conjugate symmetric data encoding system.

Regardless of the heterogeneous network environment, the relay node only needs to consider the transmission layer of a single mode, and does not need to consider compatibility with multiple network structures. For the source node, the conjugate symmetric data generated by the application layer is sent to the network coding layer. The network coding layer encodes the conjugate symmetric data and hands it to the bottom layer to complete the data transmission. The relay node uploads the message to the network coding layer after receiving the message. The network coding layer stores the message and encodes the conjugate symmetric data at the appropriate time. The network coding layer judges whether the remaining space of the current coding matrix is greater than zero when it receives the data packet sent to the receiving end by the transmission control protocol TCP layer. If so, it puts the data packet into the current coding matrix and performs linear coding to generate a linear-coded combined packet. The header of the combined packet includes the coefficient of this linear coding, the index number of the coding matrix and the serial number of the data packet. When the current node establishes a connection with other nodes, the encoded message is handed over to the bottom layer to complete the transmission of the conjugate symmetric data encoded data. The receiving node is the target node of the message. After receiving the encoded message, the network encoding layer of the target node stores the message in the local cache and uploads the successfully decoded message to the application layer to complete the encoding of conjugate symmetric data.

IV. Congestion Determination of Conjugate Symmetric Data Transmission Channel based on Machine Learning

Based on the above data fusion and data coding, machine learning methods are used to determine the congestion of the conjugate symmetric data transmission channel. In the research of multi-rate congestion control mechanisms, a key question is how to judge the current transmission path congestion [9].

Random early detection is performed first. Early detection means that the router detects congestion by monitoring the average queue length of the output queue. When congestion is found to be approaching, it randomly discards the data packets arriving at the queue to indicate that the terminal is congested and causes the source to overflow the queue the sending rate was reduced before to ease network congestion. The expression of random early detection is:

$$J = w\sqrt{e}\,/\,q + b \qquad (5)$$

In the formula, e represents the probability of dropping the packet, q represents the weight of the current queue length, b represents the actual queue length when sampling and measuring.

Through early detection, "filter" out the short-term queue length changes caused by data bursts, and try to reflect the long-term congestion. The relationship between drop probability and average queue length is shown in Fig. 3.

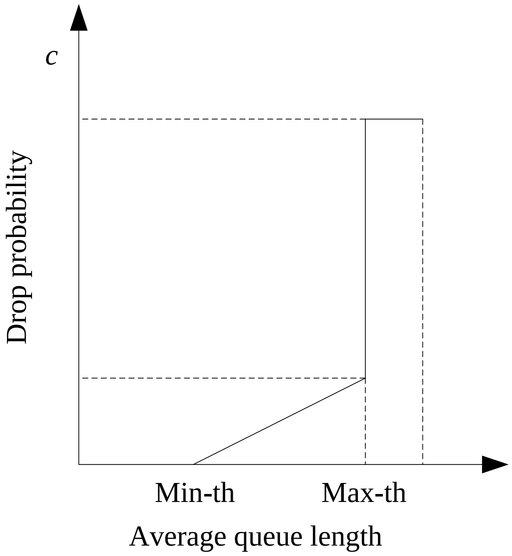


Fig. 3. Relationship between drop probability and average queue length.

In Fig. 3, c represents the probability of packet loss, and Min-th and Max-th respectively represent the thresholds related to queue length.

According to the relationship between the obtained drop probability and the average queue length, a deep learning algorithm is used to calculate the initial judgment weight:

$$\alpha = \frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)J \qquad (6)$$

In the formula, ε represents the normalized judgment factor.

The calculation formula for weight update is:

$$conf\,(i) = \sum_{i=1}^{I} \alpha\gamma(i) \qquad (7)$$

In the formula, $\gamma(i)$ represents the weight update training function.

On this basis, the available bandwidth is estimated. In multi-rate congestion control, each receiver needs to determine whether the current subscription level conforms to the available bandwidth of the transmission path, so estimating the available bandwidth is an important part of multi-rate congestion control. For the congestion mark statistical value marking, the receiving end only counts the congestion mark values collected in the synchronization interval after receiving the synchronization mark packet, considering that the mark value carried by the recent packet can better reflect the current network congestion status, So a machine learning algorithm is used to calculate the average congestion marker value, First, calculate the window value decline ratio. When packet loss occurs in congestion, three repeated ACK are received continuously, and the window threshold will drop by a certain ratio. Different congestion algorithms adopt different ratios, for example, RENO is 0.5, while CUBIC is 0.8. According to the ratio of congestion window values before and after the packet loss point, the window value reduction ratio can be determined. Then calculate the congestion avoidance window growth function. In the stage of congestion avoidance, for the functional characteristics of window growth, the difference between the window values at two-time points in this stage and the starting point window value in the stage of congestion avoidance is used, and the relative value is used to express the function curve characteristics in this stage. Then, the fast recovery window growth function is calculated. In the fast recovery stage, when the server receives three DUPACK in succession, different algorithms process differently, such as waiting for retransmission after the timeout, or constantly increasing the window value until the retransmitted packet is received. In this paper, this feature is expressed by the difference between the window value and the starting point value at a certain time point in this stage. As shown in the following formula:

$$k\left(a\right) = conf\,(i)\left(1-d\right) * R \qquad (8)$$

In the formula, $k\left(a\right)$ represents the congestion mark value carried by the a packet, d represents the statistical value of the congestion mark, R represents the weight of the congestion mark value carried by the packet. When the statistics are completed, the congestion mark value collected in the synchronization interval is cleared to avoid interference with the statistics of the next synchronization period [10].

Determine the transmission channel congestion according to the average congestion flag value obtained. When $k(a) > 0$, it means that the transmission channel is in a congested state. When $k(a) < 0$, it means that the transmission channel is in a non-congested state. When $k(a) = 0$, it means that the transmission channel at this time is just saturated.

V. Formulation of Conjugate Symmetric Data Transmission Protocol

Because the data collected by the transmission data itself has high redundancy, the direct communication between the single node and the cluster head in the cluster has little effect on the reliability of the data communication of the entire network, and consumes a lot of resources to maintain the data collected by the node is not worth the gains. Due to the limited communication range of the conjugate symmetric data cluster head, the cluster head can only forward the request message to the base station through the adjacent cluster head. This data packet can record the node information traversed during the forwarding process and the reception at each node for signal strength parameters, when the data packet reaches the base station, the base station analyzes and calculates the information recorded in the data packet to perform path generation. Convert all the data request messages that need to be transmitted, the expression is:

$$L = \beta / I + \frac{\lambda}{O} k(a) \quad (9)$$

In the formula, β represents the set of the head node and base station node of the transmission data cluster, I is the set of communication links between any two transmission data cluster heads, which can also be considered as the edge set corresponding to the node set, O is the signal strength received by the node, λ A is the transmitted signal strength of the parent node when forwarding the packet, the signal strength is related to the transmission power and the communication distance.

In order to maximize the multi-path routing from a node to the base station, the idea of breadth-first algorithm is applied in the path generation process. The algorithm process is as follows:

step1: Initialization, input all cluster head and base station node sets into the transmission data set;

step2: Calculate the number of all neighbor nodes of this node;

step3: If the number of neighbor nodes is empty, go to step1, otherwise go to the next step;

step4: Input all neighbor nodes into the set to make the following judgment: if there is a node and it is the same as the parent node of the input node, delete it; if there is a node and the energy has been excessively consumed, delete it; if there is a node Export the path.

The above process generates a transmission path, but due to the limitation of the transmission channel energy, the stored energy is also limited, and the output power of the sensor node increases exponentially with the increase of wireless coverage. At the same end-to-end distance, if each link uses limited transmission power, the power consumption of multi-link transmission is lower than the power consumption of directly transmitting information on a long link. In order to verify the survival time of the entire transmission channel, the multi-hop method is used when the long-distance transmission is required.

Based on the above analysis, the conjugate symmetric data transmission path is optimized. In the actual data transmission process, the energy consumption between the same distance and different links will be different due to the influence of objective factors, so energy-based the path optimization method adds transmission consumption weights to each link, so that the network can better integrate with the objective environment in which it is located, thereby better improving the reliability of data transmission [11-14]. The transmission consumption weight matrix is as follows:

$$H = \sqrt{L} * \frac{x}{K} \quad (10)$$

In the formula, K represents the set of transmission link, x represents the number of nodes transmitted in conjugate symmetric data.

It is also known that the relationship between energy consumption and communication distance during data transmission is:

$$U = \frac{i}{I} / \frac{l}{H} \quad (11)$$

In the formula, l represents the signal attenuation coefficient.

Combining the above two formulas, a link energy consumption matrix consisting of transmission energy on the corresponding link is obtained:

$$T = z(V) / n \quad (12)$$

In the formula, T represents the transmission energy consumption, z represents the signal strength parameter, V represents the transmission consumption weights, n represents the number of nodes.

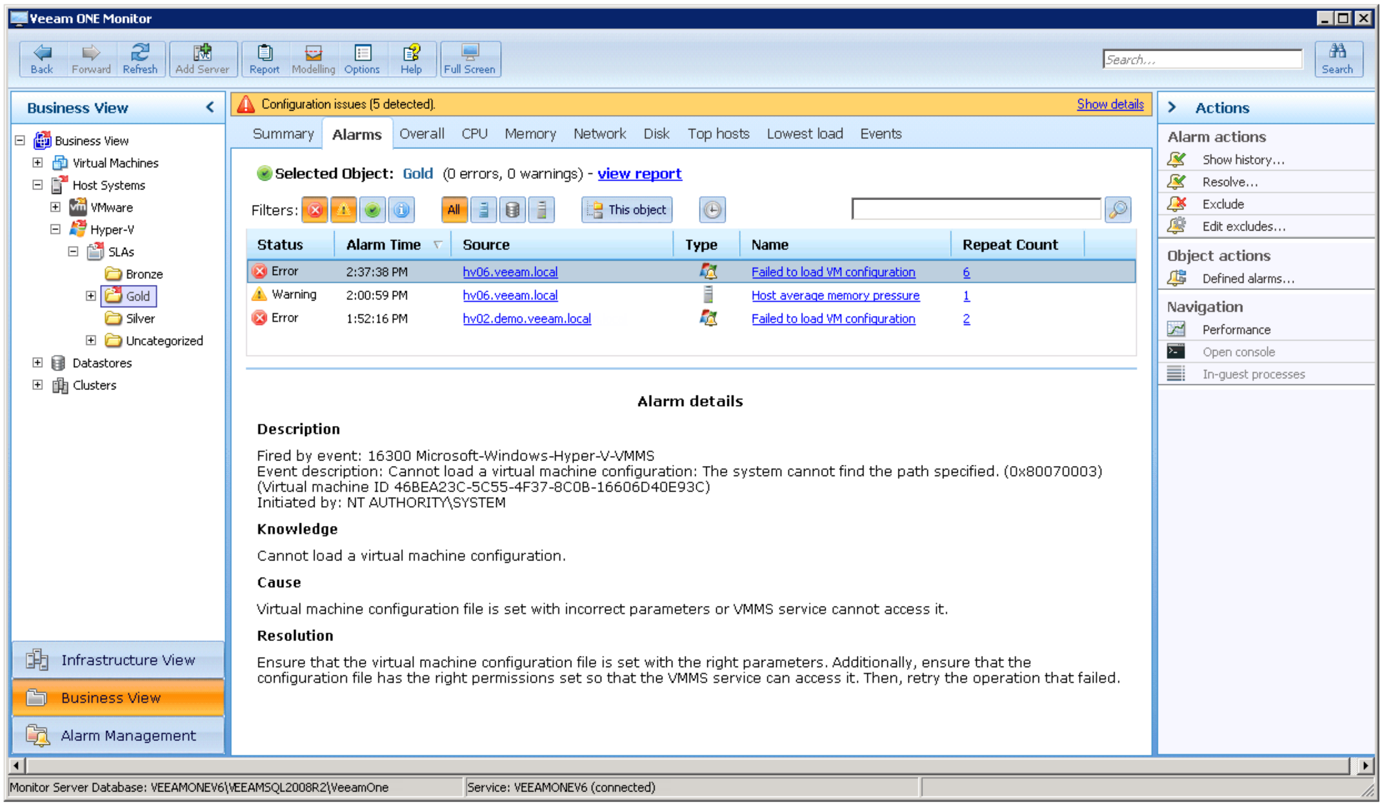
Calculate and sequence the energy consumption of all available paths, and the transmission path with the lowest energy consumption is taken as the optimal path, followed by the other paths. If there are paths with equal transmission energy consumption, they are sorted according to the sum of the consumption weights of the entire path. The smaller T is, the better the transmission environment is, that is, the higher the transmission reliability, so the higher the priority of this path.

Through the above process, the route optimization is completed. Since the data transmission workload undertaken by each cluster head is different, the data transmission workload of the cluster head close to the base station is much larger than the workload of the cluster head far from the base station, so the network the energy consumption rate of the cluster head is different. At the same time, because the network will consume a large amount of node energy consumption and generate a large number of redundant data packets during large-scale reselection, frequent reselection requires a lot of networks running time. The efficiency of the network in collecting and transmitting data is reduced, so multi-path updates and transaction processing are required in time. The remaining energy of the cluster head is low, and it cannot continue to assume the current role. The number of nodes in the area under the jurisdiction of the cluster head is exhausted. In order to facilitate routing updates and effectively handle network transactions, the following standards are set. Ordinary nodes indicate to the cluster head that their remaining energy is below the threshold and cannot continue to work. Calculate the total number of energy-depleted nodes at the same time as the base station, and then perform data transmission [15-20].

VI. Experimental Comparison

In order to verify the application performance of the proposed conjugate symmetric data transmission control method based on machine learning, a comparative experiment was carried out. This experiment was carried out on the ONE simulation platform. The ONE simulation platform is a very widely used simulator in tolerant network simulation. It can effectively simulate the routing and transmission of messages in the tolerant network and give detailed results reports. In addition to generating statistical message status reports and node connection status reports after the simulation is completed, it also supports separate reports on message generation, relaying, successful transmission, abnormal interruption, and connection time between nodes. The experimental platform is graphical. The real-time monitoring and event display interface are shown in Fig. 4.

The main parameter settings of the simulated scene are shown in Table I.



Configuration issues (5 detected). Show details

Selected Object: Gold (0 errors, 0 warnings) - view report

Status	Alarm Time	Source	Type	Name	Repeat Count
Error	2:37:38 PM	hv06.veeam.local		Failed to load VM configuration	6
Warning	2:00:59 PM	hv06.veeam.local		Host average memory pressure	1
Error	1:52:16 PM	hv02.demo.veeam.local		Failed to load VM configuration	2

Alarm details

Description

Fired by event: 16300 Microsoft-Windows-Hyper-V-VMMS
Event description: Cannot load a virtual machine configuration: The system cannot find the path specified. (0x80070003)
(Virtual machine ID 46BEA23C-5C55-4F37-8C0B-16606D40E93C)
Initiated by: NT AUTHORITY\SYSTEM

Knowledge

Cannot load a virtual machine configuration.

Cause

Virtual machine configuration file is set with incorrect parameters or VMMS service cannot access it.

Resolution

Ensure that the virtual machine configuration file is set with the right parameters. Additionally, ensure that the configuration file has the right permissions set so that the VMMS service can access it. Then, retry the operation that failed.

Monitor Server Database: VEEAMONEV6\VEEAMSQL2008R2\VeeamOne | Service: VEEAMONEV6 (connected)

Fig. 4. Graphical interface of the experiment platform.

TABLE I. MAIN PARAMETER SETTINGS OF EXPERIMENTAL SIMULATION SCENARIOS

Classification	Main content	Specification
Simulation scene	Simulation space size	5000 m×5000 m
	Number of nodes	20
	Data transfer model	Infection routing/NTC
	Nodal motion model	Random Way Point
Common node parameters	Movement speed	Random change between5m/s-10m/s
	Communication bandwidth	2Mbps
	Cache size	5M
	Communication distance	100 m
Message parameters	Size	500K–1M
	Number	200
	Production methods	Randomly generated within 1000 seconds
Datasets	IMDBDatasets	52.4MB

The experimental environment configuration for this study is as follows: The experiment was completed using the MATLAB platform, with a running memory of 16g, a computer operating system of WIN10, and a CPU of AMD Ryzen 7 4800U. The following points need to be explained about the parameter setting:

First: 20 nodes are evenly distributed and randomly move in a 5000m×5000m plane space. The communication distance of the nodes is 100m, so the network connection rate is low and belongs to sparse network.

Second: Only when the two nodes reach the communication distance of each other can the connection be established, and then the information exchange can be carried out.

Third: It is convenient for comparative analysis. The message event adopts a fixed configuration, and 200 unicast messages are randomly generated in 1000 seconds with a tool in advance. Each node has 10 messages sent to other nodes. The messages are random sizes of 500K to 1M. Since the node cache is only 5M, the network load is heavy.

A. Evaluation Index of Conjugate Symmetrical Data Transmission

The experimental comparison methods are the conjugate symmetric data transmission control method based on machine learning, the data transmission congestion control method based on the bidirectional path and the data transmission control method based on accurate error tracking equalization. From the perspective of the message delivery rate, three indicators are used to measure the data transmission effect of the tolerant network, including the message delivery rate, message transmission overhead, and average message transmission delay.

The delivery rate refers to the ratio of the number of successfully delivered messages to the total number of messages generated by the network. The expression is:

$$deliverRatio = \frac{N}{E} \quad (13)$$

In the formula, E represents the total number of messages generated in the experiment, N represents the number of successfully delivered messages.

In the case of multiple copies, the number of the same message is recorded as 1, regardless of the number of copies. The delivery rate indicates how many messages can be sent by the entire delay-tolerant network, and can measure the network data transmission capability. Increasing this value is the main goal of the delay-tolerant network data transmission problem.

Transmission overhead refers to the ratio of the total number of message transmissions in the network to the number of successfully transmitted messages. The expression is:

$$deliverOverhead = \frac{n_i}{\partial} \quad (14)$$

In the formula, $deliverOverhead$ represents the transmission overhead, ∂ represents the number of successfully transmitted messages, n_i represents how many times the message i has been transmitted during the transmission process. In the case of multiple copies, the transmission of each message copy must be included.

The average transmission delay represents the statistical average of the transmission delays of all successfully transmitted messages. The expression is:

$$AverageDelay = \frac{\varphi}{t_Q} \quad (15)$$

In the formula, t_Q represents the transmission delay of the message, φ represents a successfully transmitted message. The average transmission delay is a statistical average value that characterizes the efficiency of the network to transmit messages.

B. Delivery Rate Comparison

The results of the three methods of delivery rate comparison are as Fig. 5.

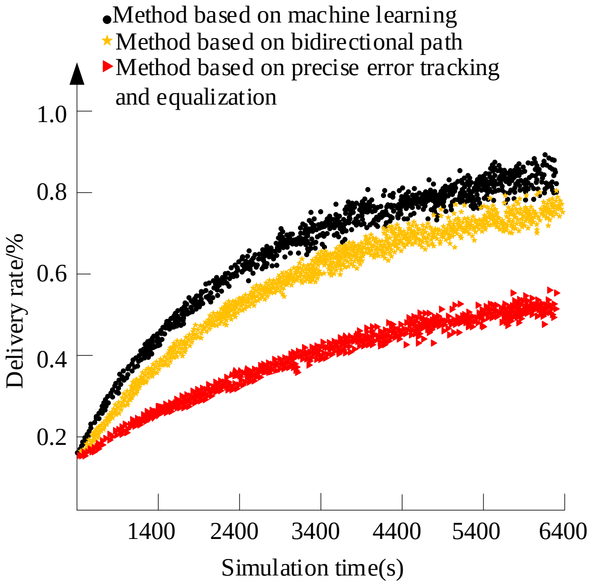


Fig. 5. Delivery rate comparison results.

It can be seen from the above figure that the application of conjugate symmetric data transmission control method based on machine learning to data transmission is beneficial to improve the delivery rate of the message, because the method designed this time applies the encoding technology to encode the transmission rate of the transmission data. It is very close to the infection route, and sometimes even lower than the infection route. This is mainly because the target node has not collected enough messages to complete the decoding when the time is short, which shows that the delivery rate is slightly lower; when the simulation reaches a certain time, the delivery rate of the method no longer changes significantly with time. At this time, the target node can decode the source message with a sufficient number of encoded messages. Therefore, as the time continues to increase, the message delivery rate in the strategy of applying network encoding continues to increase. The delivery rate of the traditional data transmission congestion control method based on the two-way path and the data transmission control method based on accurate error tracking equalization is lower than the method designed this time, mainly because the traditional method cannot store more information. Even if there is a connection opportunity between the nodes of the transmission network, more messages cannot be transmitted, thereby reducing the data delivery rate.

C. Transmission Cost Comparison

The transmission cost comparison results of the conjugate symmetric data transmission control method based on machine learning, the data transmission congestion control method based on the bidirectional path and the data transmission control method based on accurate error tracking equalization are as Fig. 6.

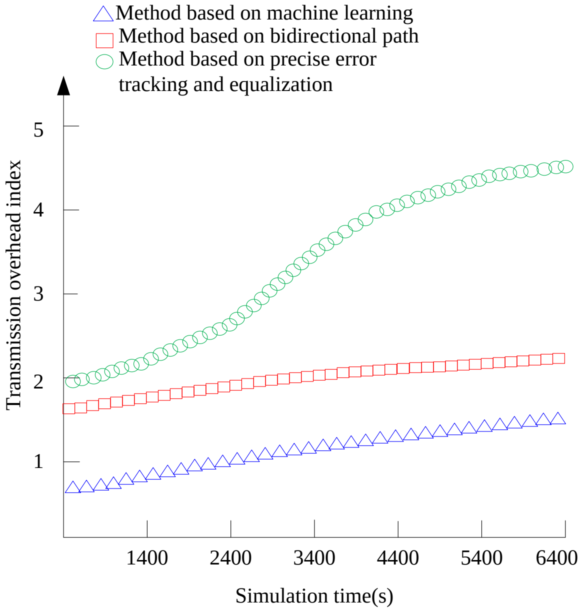


Fig. 6. Comparison of transmission overhead.

As can be seen from the above figure, from the comparison of the transmission costs of the three methods, in the initial data transmission, the transmission overhead of the three methods is lower, and as the time increases and the number of messages increases, the messages transmitted between the nodes also change. However, the message transmission overhead is closely related to the number of network transmissions. Therefore, in the later stage of simulation, the message transmission overhead becomes very large. However, it can be seen from the comparison that the transmission cost of the traditional two methods is much higher than the proposed method.

D. Comparison of Average Propagation Delay

The comparison results of the data transmission congestion control method based on the bidirectional path and the data transmission control method based on accurate error tracking equalization and the conjugate symmetric data transmission control method based on machine learning designed this time are shown in Fig. 7:

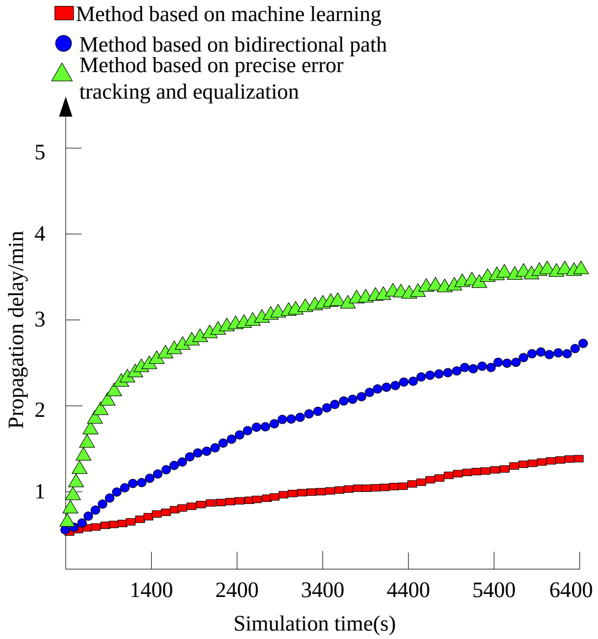


Fig. 7. Comparison of average transmission delay.

It can be seen from the above transmission delay curve that the transmission delay of the data transmission congestion control method based on the bidirectional path and the data transmission control method based on accurate error tracking equalization are higher than the transmission method designed this time. The reason for the low transmission delay of this design method is that the designed method can identify the congestion degree of the transmission channel and can encode the data, thereby improving the transmission efficiency of the message.

VII. Conclusion

In conjugate symmetric data transmission, the transmission path will be congested when the amount of transmitted data is large. This paper studies a conjugate symmetric data transmission control method based on machine learning, which tracks the target data, fuses conjugate symmetric data according to the calculation result of the best tracking signal, encodes the data, calculates the average congestion mark value by machine learning, and judges the congestion of data transmission. Transmission control of conjugate symmetric data is realized by transmission protocol. Experimental results show that the proposed method has low transmission cost, low transmission overhead and short transmission delay. The research method can identify the congestion degree of transmission channel and realize the efficient transmission of conjugate symmetric data.

The research on conjugate symmetric data transmission control is still in its preliminary stage, and there is still much work to be discussed and carried out. Further work mainly includes:

First: In the current congestion control algorithm based on asymmetric links, the fairness of data packets in bidirectional data transmission is still a problem that needs further research.

Second: Establish a reasonable data transmission model to theoretically analyze the characteristics of data transmission between nodes. The movement of nodes in the network is very random, and the connections between nodes are also opportunistic. The modelling of the entire network is very complicated, but the data transmission model between nodes can be mathematically modeled using random processes and probability theory, the mathematical description of the problem, and then theoretically evaluate the data transmission.

Third: Study the security problem of data transmission based on network coding. The transmission mechanism of this study is only applicable to the process of node "storing-carrying-forwarding" messages. It fails to consider how to ensure the security of messages in data transmission using network coding technology Transmission problems. If malicious messages are involved in the encoding, all subsequent encoded messages based on this message will carry the malicious message information. The spread of such malicious messages poses challenges to the security requirements of system data integrity. The security mechanism should contain guarantee mechanisms against such attacks.

Fourth: In-depth study of the problem of code generation management using network coding in the Rongchi network. Separate airspace coding is the preferred organization strategy when the network is small in scale, short in running time, and has few message events. However, when the large-scale network runs for a long time and there are many message events, it will bring about the problem of expansion of the coding generation space, so you need to consider "Time domain" encoding. That is, when the coding generation is divided, the message is divided into different generations according to a certain period of time. Although the time-domain coding is simple and looks like the superposition of several spatial coding processes, there are many problems that are very different from the simple superposition analysis, especially the problems of cache management and message receipt caused by multiple time-domain coding generations. The time-domain coding of messages also needs further study.

References

[1] I. A. Badruddin, N.J.S. Ahmed, A.E. Anqi, S. Kamangar, “Conjugate heat and mass transfer in a vertical porous cylinder,”Journal of Thermophysics and Heat Transfer, 2019, 33(2), 548-558.

[2] K. Zhou, “Simulation of data integrity detection method for long-distance wireless hybrid transmission,”Computer Simulation, 2019, 36(12), 447-450.

[3] N. A. Kuznetsov, D.V. Myasnikov, Semenikhin K.V., Optimal control of data transmission over a fluctuating channel with Unknown State, Journal of Communications Technology and Electronics, 2018, 63(12), 1506-1517.

[4] Yi L A , Peng Y B . Internet of Things transmission and network reliability in complex environment. Computer Communications, 2020, 150:757-763.

[5] Mu S , Wang F , Xiong Z , et al. Optimal path strategy for the web computing under deep reinforcement learning. International Journal of Web Information Systems, 2020, 16(5):529-544.

[6] Li H , Lei W , Zhang W , et al. A joint optimization method of coding and transmission for conversational HD video service. Computer Communications, 2019, 145(SEP.):243-262.

[7] Pyeon D , Yoon H . An efficient multi-path pipeline transmission for a bulk data transfer in IEEE 802.15.4 multi-hop networks. Wireless Networks, 2019, 25(1):117-130.

[8] WANG Zhan-Yu,LIU Guo-Ping. Performance Improvement of NCSs Under Complex Network via Concurrent Paths.System science and complexity,2019,032(002):453-478.

[9] C. M. Buldun, J. X. Jean, M. R. Bedford, M. Howarth, "Snoopligase catalyzes peptide-peptide locking and enables solid- phase conjugate isolation,"Journal of the American Chemical Society, 2018, 140(8), 3008-3018.

[10] X. Li, X. Fan, H. Ren, et al., "Data-driven feature analysis in control design for series-compensated transmission systems," IEEE Transactions on Power Systems, 2019, 34(4), 3297-3299.

[11] H. Han, K. A. Shore, "Dynamical characteristics of nano-lasers subject to optical injection and phase conjugate feedback, Iet Optoelectronics, 2018, 12(1), 25-29.

[12] C. Paroma, P. Anamitra, J. S. Thorp, et al., "Error reduction of phasor measurement unit data considering practical constraints, IET Generation," Transmission & Distribution, 2018, 12(10), 2332-2339.

[13] T. Fu, B. Jiang, Z. Li, "Approximation algorithms for optimization of real-valued general conjugate complex forms,"Journal of Global Optimization, 2018, 70(1), 99-130.

[14] H. Yamazaki, M. Nagatani, H. Wakita, et al., "IMDD transmission at net data rate of 333 Gb/s Using Over-100-GHz-Bandwidth Analog Multiplexer and Mach–Zehnder Modulator," Journal of Lightwave Technology, 2019, 37(8), 1772-1778.

[15] W. El-Shafai, S. El-Rabaie, M. M. El-Halawany, et al., "Effective multi-stage error control algorithms for robust 3D video transmission over wireless networks, "Wireless Networks, 2019, 25(4), 1619-1640.

[16] M. Pourali, J. A. Esfahani, S. A. Fanaee, Two-dimensional analytical investigation of conjugate heat transfer in a finite-length planar micro-combustor for a hydrogen-air mixture, International Journal of Hydrogen Energy, 2019, 44(23), 12176-12187.

[17] L. Li, H. Long, L. Zhao, et al., "Retransmission scheme for contention-based data transmission systems," Iet Communications, 2018, 12(2), 144-151.

[18] G. Singh, P. Goswami, R. V. Anand, "Exploring bis-(amino)cyclopropenylidene as a non-covalent bronsted base catalyst in conjugate addition reactions,"Organic & Biomolecular Chemistry, 2018, 16(3), 384-388.

[19] R. Zhang, D. Zeng, J. H. Park, et al., "A new approach to stabilization of chaotic systems with nonfragile fuzzy proportional retarded sampled-data control," IEEE Transactions on Cybernetics, 2018, 49(9), 3218-3229.

[20] A. P. Perkki, "Conjugates of integral functionals on continuous functions,"Journal of Mathematical Analysis and Applications, 2018, 459(1), 124—134.

Towards Finding the Impact of Deep Learning in Educational Time Series Datasets – A Systematic Literature Review

Vanitha.S[1], Jayashree.R[2*]
Department of Computer Applications-College of Science and Humanities,
SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, India

***Abstract*—Besides teaching in the education system, instructors do a bunch of background processes such as preparing study material, question paper setting, managing attendance, log book entry, student assessment, and the result analysis of the class. Moreover, Learning Management System(LMS) is mandatory if the course is online. The Massive Open Online Course (MOOC) is an example of the worldwide online education system. Nowadays, educators are using Google to efficiently formulate study material, question papers, and especially for self-preparation. Also, student assessment and result analysis tools are available to get instant results by feeding student data. Artificial Intelligence (AI) is driving behind these applications to deliver the most precise outcome. To accomplish that, AI requires historical data to train the model, and this sequential (year-wise, month-wise, etc) information is called time series data. This Systematic Literature Review (SLR) is conducted to find the contribution of time series algorithms in Education. There are enormous changes in algorithm architecture analogized to the traditional neural network to endure all kinds of data. Though it significantly raises the performance, it expands the complexity, resources, and execution time as well. Due to this, comprehending the algorithm architecture and the method of the execution process is a challenging phase before creating the model. But it is essential to have enough knowledge to select the suitable technique for the right solution. The first part reviews the time series problems in educational datasets using Deep Learning(DL). The second part describes the architecture of the time series model, such as the Recurrent Neural Network (RNN) and its variants called Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), the differences between each other, and the classification of performance metrics. Finally, the factors affecting the time series model accuracy and the significance of this work are summarized to incite the people who desire to initiate the research in educational time series problems.**

Keywords*—Deep learning; education; gated recurrent unit; long-short term memory; recurrent neural network; time series

ABBREVIATIONS

SLR	Systematic Literature Review
PRISMA	Preferred Reporting Items for Systematic Review and Meta-Analyses
LMS	Learning Management System
MOOC	Massive Open Online Courses
CNN	Convolutional Neural Network
AE	Auto Encoder
DBN	Deep Belief Network
GAN	Generative Adversarial Network
DRL	Deep Reinforcement Learning
FFNN	Feed Forward Neural Network
MLP	Multi Layer Perceptron
ERNN	Elman Recurrent Neural Network
ESN	Echo State Network
TCN	Temporal Convolutional Network
LR	Linear Regression
NB	Naive Bayes
SVM	Support Vector Machine
DT	Decision Tree
RF	Random Forest
GBM	Gradient Boosting Machine
AUC	Area Under Curve
ADAM	Adaptive Moment Optimization algorithms
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Oversampling Technique
HMM	Hidden Markov Model

I. INTRODUCTION

Human education assists in automating massive work with less intervention of human resources. That education system itself automated with the help of AI. Machine Learning is a subset of AI. Likewise, Deep Learning (DL) is a division of broader machine learning based on the Neural Network (NN) designed to mimic the human brain. DL is becoming an imperative buzzword in data handling technology due to the potential of prediction using extensive data. However, the prediction system was enlightened only after the innovation of RNN. The RNN is chosen for this study due to its architecture to operate on sequential nature data. For example, Predicting learner dropout rate in MOOC using LMS interaction data (user click events, weekly assignments, etc). All educational institutions switched online to continue the classes during the corona lockdown period. Many online courses started and then boomed. The MOOC is one of the popular platforms for online education. But, the course completion rate is significantly lower than the number of registration due to being free of cost. RNN helps to predict the success and dropout rate of MOOC learners. RNN applications are unrestricted in all the fields, such as finance [1], [2], medicine [3], [4], [13], and nature-related forecasts, such as weather, rainfall, temperature, and wind speed [5]-[8]. Also, enough surveys are available to enrich the existing work on those domains. But in education, reviews still need to be conducted

*Corresponding Author, jayashrr@srmist.edu.in

to find the related work in sequential data collectively. Hernández et al.,[21] did the same job, but that did not focus on time series. This work fills this research gap with the time series model architecture, the difference from the conventional neural network, and the parameters influencing the model performance. The following are the research questions identified for this work:

RQ1: Finding the impact of Deep Learning in educational time series problem.

RQ2: Identify the architecture of time series model and how it differs from the traditional approach.

RQ3:Discover the significant factors affecting the time series model accuracy.

The remaining paper encloses five sections. Section II defines the methodology of this work, and Section III describes the review results including previous work using the deep learning model,working methodology of RNN, LSTM, and GRU, and metrics used for the model. Section IV outlines the contribution of this paper through discussion. Finally, Section V explains the conclusion and future work of this article.

II. Research Methodology

This section elucidates the research methodology followed in carrying out this review process and the filtration of the downloaded papers. The following research repositories are accessed: Google Scholar and IEEE Xplore. The keyword used for this work is the following: "Deep Learning", "RNN", “Time Series", "Student”, and “Education”. The google search result showed many research papers, and all are evaluated manually to select the suited one for this work. The selection process considers the journal articles using time series data in Education and valid conference papers. The inclusion and exclusion criteria of this study is mentioned in Table I.

The article selection process followed the PRISMA method to carry out this study. PRISMA is an abbreviation of the "Preferred Reporting Items for Systematic Review and Meta-Analyses". Fig. 1 explains the step-by-step article inclusion and elimination details through PRISMA 2020 flowchart.

1) Identification: The initial search retrieved two hundred and ninety-one (n= 291) documents from Google scholar and the IEEE database. The filter is applied for the last five years (2018-2022) to restrict the search before 2018 and after 2022. Then removed, four duplicate records from various databases.

2) Screening: There are two screening steps to check the paper's eligibility. i) Preliminary check ii) Full-text analysis. Step 1 investigates the title and abstract to verify the document's relevance. It removed one hundred-six (n=106) reports and included eighty-one (n=81) articles for full-text retrieval. Then sixteen (n=16) documents are eliminated due to paid version. Step 2 inquiry prevents invalid articles, conferences, and other documents irrelevant to this context.

3) Included: The previous stage gives twenty-two reports (n=22), and the selected articles are used as a source for this Systematic Literature Review(SLR) or meta-analysis.

TABLE I. Selection and Rejection Criteria of the Study

Criteria	Inclusion	Exclusion
Year	Documents published between 2018 to 2022	Documents published before 2018 and after 2022
Language	Articles in English	Other language articles
Domain & Data	Educational time-series documents	Other domains and the data which are not using time-series
Article type	Journal and conference	- Articles less than 6 pages - Articles having less than 20 citation
Algorithm	Deep Learning	Machine Learning

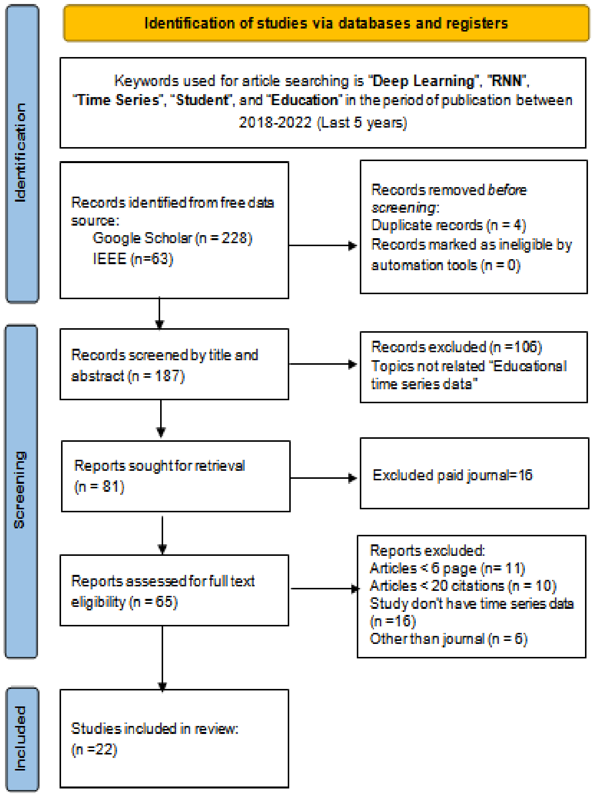

Identification
Screening
Included
Identification of studies via databases and registers
Keywords used for article searching is "Deep Learning", "RNN", "Time Series", "Student", and "Education" in the period of publication between 2018-2022 (Last 5 years)
Records identified from free data source:
Google Scholar (n = 228)
IEEE (n=63)
Records removed before screening:
Duplicate records (n = 4)
Records marked as ineligible by automation tools (n = 0)
Records screened by title and abstract (n = 187)
Records excluded (n =106)
Topics not related "Educational time series data"
Reports sought for retrieval (n = 81)
Excluded paid journal=16
Reports assessed for full text eligibility (n = 65)
Reports excluded:
Articles < 6 page (n= 11)
Articles < 20 citations (n = 10)
Study don't have time series data (n =16)
Other than journal (n = 6)
Studies included in review: (n =22)

Fig. 1. PRISMA 2020 flow diagram of the study.

III. Results

This section describe the review results obtained through previous section. Fig. 2. depicts the publisher's contribution to this topic. It shows that most well-known publishers are involved, but springer published more articles than others.

Fig. 3. explains the number of publications year-wise. It depicted the growing trend from 2018 to 2021 and decreased in 2022. Due to corona, online education peaked in 2021, and most of the research was conducted on various dimensions using vast online data such as MOOC and other LMS platform.

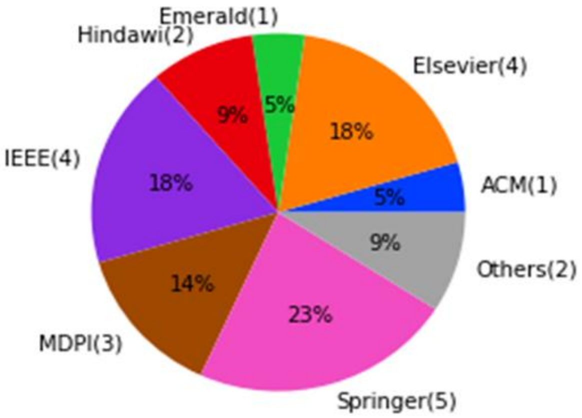


Fig. 2. Publisher contribution.

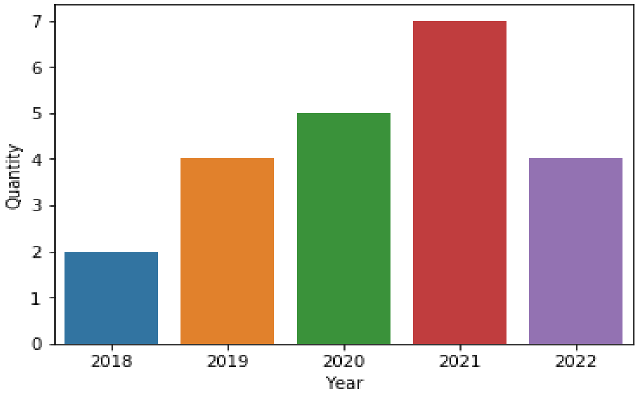


Fig. 3. Year wise publication.

A. Contribution of Deep Learning in Educational Time Series Data

Wang et al.,[24] proposed two novel methods to predict student learning status. The first one is to retrieve compelling features and performance using Conv-GRU. The second one, xNN (Explainable Neural Network) explains the relevance of student positive/negative results to improve the weak area. This approach helps to identify the hidden pattern of student behavior and early notification to improve the particular section.

Waheed et al., used the same dataset (OULAD) in both of their papers [25, 29], but followed different methodologies to predict the student category. The first work gives the highest accuracy (93%) than the second using DNN(84%), with a notable difference. The deep neural network(DNN) proves its power by providing the highest accuracy. Mubarak et al., [26] predict learner's weekly performance using video click stream for timely intervention. This model is created in such a way that it can adjust a variable window length routinely, which helps it to fit the RNN layer dimensions with different sizes of input data.

In studies [27, 28, 39, 42], all the authors used the same dataset (KDD cup 2015) to predict student dropout, and the result shows above 85% performance in all. It contains 39 courses and seven kinds of student behavioral information such as Access, video, wiki, discussion, navigate, page_close and problem. These multiple parameters allow applying a multi-variate time series approach. Though the dataset is the same, imbalanced data is handled only in [28, 39].

Zhang et al., [30] introduced a predictive model to pick the micro-level pattern from student learning behavior. To avoid data sparsity, the author divides the data into five clusters based on the nature of the student's learning behavior. Because, the author believes that every student's online learning behavior will change depending on their free time. An auto-encoder is used to encode time-series data. The significant difference between recall and accuracy values shows that classification errors need to fix in this model.

He and Gao [31] proposed a student performance predictive model by collecting student learning behavior information through terminal data acquisition tools to find the student concentration level in the classroom and explore the influencing factors of learning concentration. Aljaloud [32] suggested a model to predict student learning outcomes by selecting the number of essential features and evaluating the result by reducing the number of features. There are seven features(f1,f2,f3,f4,f5,f6,f7) and seven courses used in this LMS, and the final result shows the best accuracy in the more number of attribute combination.

Chen et al., [33] created an intelligent framework to handle imbalanced datasets and spatiotemporal information. This LMS contains eight learning features (F1-F8): assignment, file, forum, homepage, label, page, quiz, and URL. Course length is 16 weeks, but this model helps to warn the at-risk students much earlier than other models with higher accuracy.

Karim et al., [34] conducted ablation tests on time series data using LSTM. In this experiment the LSTM block is substituted by other techniques such as GRU, RNN, and Dense block. But LSTM-FCN performance was higher than others.

Chen et al., [35] provided a comparative study between deep learning and conventional machine learning using the data retrieved from the Learning Management System (LMS). This data tells the temporal behavior of the student activity in the form of time series. The author used classification and clustering techniques to predict early identification of at-risk students, and then compared the results using AUC.

Li et al., [36] also did a comparative study using higher education data such as student grades and levels to predict the performance. Prabowo et al., [37] tried dual input, the combination of categorical and numerical time series data. The proposed dual-input hybrid model combines MLP and LSTM networks and then compares the perrmance with the individual model.

Asish et al., [38] offered a comparative study using CNN, LSTM, and CNN-LSTM to classify the student distraction level using eye gaze data. The author collected the data through a Virtual Reality Environment. Wuet al., [39] proposed a hybrid model called CNN-Net to predict student dropout in MOOCs. Moreover, the author handled the class imbalance due to the massive dropout ratio of students.

Shin et al., [40] created a model to predict student performance using time series data by clustering the students using the k-shape technique. Each cluster helps to identify the student category to give a warning from the instructors. Bousnguar et al., [41] proposed a model for enrolment prediction using LSTM and statistical machine learning. The statistical model gives the highest accuracy than deep learning due to the insufficient data for training.

Qiu et al., [42] developed a model for dropout prediction using CNN. The author compares the results with baseline models, including LR, Naïve Bayes, Decision Tree, Random Forest, Gradient Tree Boosting, and SVM. Among those, CNN with windows size 10 showed better results than the others. Chen et al., [43] created a model for predicting course performance using an imbalanced dataset. The SMOTE sampling technique is applied to balance the minority data.The author used the KNN algorithm to fill in the arbitrary missing values.

Aljohani et al., [44] proposed a model to find the at-risk student in the early stage based on weekly performance sequence data. But it achieved the highest accuracy only after 38 weeks. He et al., [45] suggested a model for student performance prediction. The author used two fully connected neural networks for demographic information; RNN for handling student assessment and click stream time series data. The proposed method provided better performance than the baseline models.

Tables II provides the vital points of this review and Fig. 4 represent the classification of time series use case and workflow found in this review.

TABLE II. REVIEWS OF PREVIOUS WORK IN EDUCATIONAL TIME SERIES DATA

Source	Model	Dataset	Samples/Train-Test	Purpose/Findings	Individuality
[24]	ML, BPNN, RNN, GRU,LSTM, Conv-GRU-MaxP, Conv-GRU-AvgP	WorldUC, Liru Online Course Dataset	Datasets 1 - 7543 students, Dataset 2 - 347 students	Predictive model for student performance ML-75%, BPNN-76%, RNN-78%,LSTM-80%, GRU-81.3%, Conv-GRU-MaxP-81.8%, Conv-GRU-AvgP-82.2%	Weighted average pooling is used instead of max pooling in Conv-GRU to achive better performance.
[25]	LR,SVM,Deep ANN	Open University Learning Analytics Dataset (OULAD)	2014-2015 32,593 student log records. 70% - train 30% - test	The inclusion of legacy data and assessment-related data impact the model significantly. LR-85%, SVM - 89%, Deep ANN - 93%.	Instead of week- wise, at-risk students are identified in each quarter Q1,Q2,Q3, and Q4 along with distinction, pass, fail and withdrawal.
[26]	LR, SVM, Deep ANN, LSTM	MOOC - Stanford University Dataset	Student records for each dataset. course 1 - 5346 course 2 - 2135 course 3 - 3022 course 4- 2497 60%- training 30%- testing 10% - validation	Predicting learner performance and early dropout using video file click stream events and quiz score. LR=84%, SVM=85%, Deep ANN=85%, LSTM=93%	More than 90% accuracy in real time dataset.
[27]	CSLA(Hybrid model using CNN+Bi-LSTM+Attention Mechanism)	MOOC - KDD Cup 2015 dataset	2013-2014 79,186 students records 80% - training, 20%- testing	Predicting the student dropout rate based on learners' behavior data with accuracy - 87.6% and f1 score-86.9%.	Combined three different strategies to increase the performance over 2.8% . CNN - Feature selection LSTM - Time series Attention Mechanism- Assigning weight
[28]	LR, SVM, DNN, CONV-LSTM	MOOC- Dataset 1- Stanford University Dataset 2- KDD cup 2015.	Dataset 1- 78,623 records Dataset 2- 120,542 records 65% - training 20% - testing 15%-validation	Predicting Student Dropout using MOOC data with f1 score 89% and 90% for two datasets respectively.	Custom loss function applied to rectify classification error instead of SMOTE and other techniques used for imbalanced dataset.
[29]	LR, SVM, DT, GBT, KNN, ANN, LSTM	OLUAD	2014-2015 32,593 student log records. 75% - training 25% - testing	Predicting learners behavior using student online log data. LR-73%, SVM – 73%, DT-79%, GBT-78%, KNN-78%, ANN-83%, LSTM-84%.	Time series data(clickstream logs) converted into aggregated format for the purpose of applying classical machine learning.
[30]	LSTM - Encoder	Blackboard LMS	2014-2016 4706 students 3,625,619 log records	Predicting at-risk students using micro level behavioral pattern and time series clustering with accuracy-92%	Auto encoder used to extract best featuresStudent clustering based on the learning behavior
[31]	HMM, Machine Learning (ML), LSTM, CNN-LSTM	UCI - HAR dataset	10,299 data samples 70% - training 30 % - testing	Classroom attention behavior recognition using sensor data. HMM-72%, ML-78%, LSTM-89%, CNN-LSTM - 92%.	Identifying student concentration level through wearable device and mobile interaction data.
[32]	CNN, LSTM, CNN-LSTM	Blackboard LMS	35,000 students 1, 715, 000 records	Predicting student learning outcomes in LMS with fl	This study aims to find thedominate features called KPI

			70% - training 30%- testing	scoreLSTM - 90%, CNN - 92% CNN-LSTM - 93%	(Key Performance Indicator) to improve the model performance.
[33]	LR,SVM, LSTM, CNN-LSTM, Conv-LSTM	MOODLE-Gadjah Mada University	977 students 202,000 log records	Early prediction of at-risk students. LR-64%,SVM-80%,LSTM-85%, CNN-LSTM-88%, Conv-LSTM-91%	Hybrid SMOTE technique used for imbalance dataset More than 90% accuracy in predicting the first few weeks instead of final week.
[34]	LSTM-FCN, Attention LSTM-FCN, GRU, RNN, Dense block	University of California-Riverside (UCR) time series repository	Not specified due to the large number of experiment.	Z-normalization is recommended, if the training data having good representation of global population, Appling dimension shuffles before the LSTM block increases the performance.	Series of experiments(3627) using educational data, the State-of-Art performance for classifying the time series signal
[35]	NN,LR,NB, SVM,DT,RF, GBM,LSTM	Canadian university. (LMS log data- https://moodle.org/)	290 (semester 1)-train, validation, test data 311(semester 2)-Test data	Predicting the students' performance using LMS activity. LSTM (AUC: >60%) performs better than classical machine learning (AUC <60%).	SMOTE sampling technique is used to overcome the imbalanced dataset.
[36]	Linear Regression, LSTM	Multi-disciplinary university	2007-2016-training 2017-2019-testing	In Predicting student performance, deep learning offers the highest accuracy than the Linear model. MAE: 0.593 and RMSE: 0.785	Extracting informative data as a feature with corresponding weights. Multiple updated hidden layers were used for designing neural networks automatically
[37]	LSTM, MLP-LSTM	Bina Nusantara University	2011-2017 46,670 -univariate time-series and tabular data	Predicting student GPA using tabular and historical data. Hybrid model with dual input gives highest accuracy MSE:0.41, MAE:0.34, R-square:0.48.	Dual input to the hybrid model using tabular data and time series data
[38]	CNN, LSTM, CNN-LSTM	Student classroom data using Virtual Reality (VR) Environment	3.4 M data points 70%-Training 30%-Testing	Finding the student distraction level using Virtual Reality data by creating a deep learning model. The hybrid model achieves the highest accuracy at 89.8%	The large amount of data points were used for this experiment.
[39]	CNN-Net, CNN-LSTM,CNN-RNN, Classical Machine Learning	MOOC - KDD Cup 2015 dataset	79,186 students records 1,20,542 data points 80% - training, 20%- testing	Early prediction of student dropout in MOOC and Hybrid Model gives highest accuracy AUC: 91.5% than classical model.	The effort was given for pre-processing the data to handle categorical and imbalanced dataset.
[40]	RNN, LSTM, Deep LSTM	Dataset collected from Star Math-Formative Assessment tool	2017-2018, 10,107- records, 80% -training,10% test, 10% validation	Predicting student performance using previous test assessment. The short history (3 data points) prediction gives highest accuracy.	Student performance is predicted then categorized using clustering technique based on their performance.
[41]	ARIMA, LSTM, Exponential Smoothing, and Fuzzy Time Series algorithms	IBN ZOHR University	18 years data	Developed four different forecasting models using Time Series algorithms to predict the new student enrollment. Highest RMSE score Fuzzy Time Series: 211, ARIMA: 452,ES: 461, LSTM: 1152.	Comparison between Statistical and Deep Learning model.
[42]	CNN	MOOC - KDD Cup 2015 dataset	79,186 students records	Predicting student dropout in online courses.Precision: 86%, Recall: 87%, F1 score: 86%, AUC: 86%	CNN model is newly applied for student dropout prediction using click stream data.
[43]	NN, LR, NB, GBM SVM, DT, KNN, RF, and LSTM with SMOTE	Moodle LMS data –Canadian university	527 students 72% - training 28% - testing	Analyze student online temporal behavior using their LMS data for the early prediction of course performance. LSTM - AUC Score 80.1.	Separate models are created for 28, 48, 56, and 70 days data to evaluate the course performance for each semester.
[44]	LR, ANN, SVM, LSTM	Open University Learning Analytics Dataset (OULAD)	2014-2015 32,593 students with 20 different activity data.	Finding the at-risk students in the early stage using virtual learning environment video stream click event and demographic data. Recall score is LR=80, SVM=78, ANN=85, LSTM=95.	Student week-wise activities are stacked and given to the model for early prediction. The last week's data provides better performance.
[45]	RNN,GRU and LSTM	Open University Learning Analytics Dataset (OULAD)	2014-2015 32,593 student records.	Static and sequential informationswere combined for performance prediction. GRU gives better performance than LSTM due to minimal length data. The accuracy of the proposed model is above 80% in the last week.	The joint neural network is proposed to fit both static and sequential data, where the data completion mechanism is also adapted to fill the missing stream data.

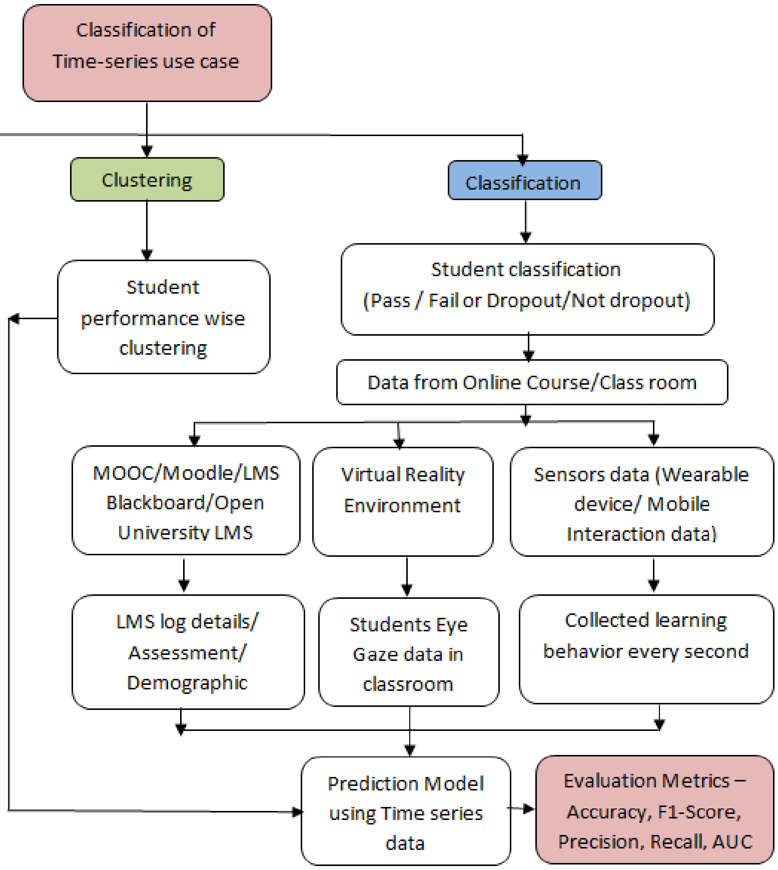

Classification of Time-series use case
Regression
Clustering
Classification
Forecasting Student Enrollment
Forecasting Student Score/GPA
University past enrollment data
University student mark data
Prediction Model
Evaluation Metrics - RMSE, MSE, MAE. R^2
Student performance wise clustering
Student classification (Pass / Fail or Dropout/Not dropout)
Data from Online Course/Class room
MOOC/Moodle/LMS Blackboard/Open University LMS
Virtual Reality Environment
Sensors data (Wearable device/ Mobile Interaction data)
LMS log details/ Assessment/ Demographic
Students Eye Gaze data in classroom
Collected learning behavior every second
Prediction Model using Time series data
Evaluation Metrics – Accuracy, F1-Score, Precision, Recall, AUC

Fig. 4. Classification of time series problem and workflow found in this review.

B. The Architecture of Time Series Model and the Difference between Traditional Approach

This section provides the history and technical background of Recurrent Neural Networks. Even though a few studies used other models (CNN and hybrid models) for time series problems, those are excluded and not specific to handle temporal data.

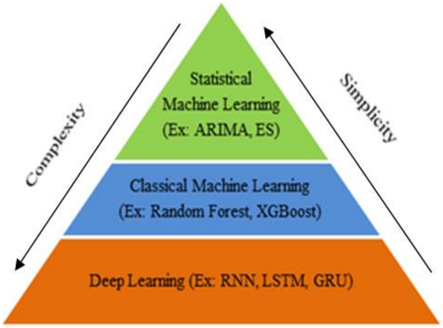

Statistical Machine Learning (Ex: ARIMA, ES)
Classical Machine Learning (Ex: Random Forest, XGBoost)
Deep Learning (Ex: RNN, LSTM, GRU)
Complexity
Simplicity

Fig. 5. Classification of time-series algorithms.

Initially, statistical methods are beneficial in predicting time series problems, but they are ineffective in handling nonlinear data. Therefore deep learning came into existence to overcome the liabilities of conventional time series algorithms such as ARIMA and Exponential smoothing techniques [9], [10], [20]. Similarly, few classical machine learning algorithms (XGBoost) apply to time series problems. Fig. 5 illustrates the complexity and the simplicity level of different time series algorithms.

The properties of time series data are Trend, Seasonal, Cyclic, and Irregular. Fig. 6 describes the pictorial representation of each property.

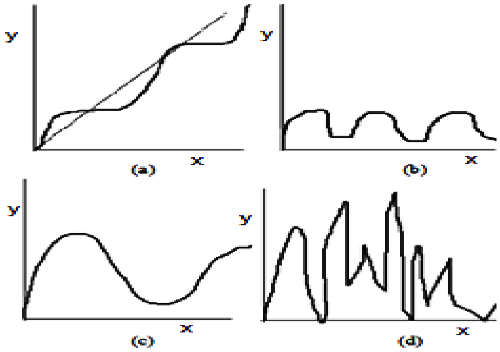


Fig. 6. Time series properties (a). Trend (b). Seasonal (c). Cyclic (d). Irregular.

Here ‘x’ denotes any time unit such as minutes, hours, months, years, etc. And the ‘y’ represents the numerical value such as weight, height, price, quantity, etc. Fig. 6 explains how the ‘y’ value is changed based on time. The appropriate algorithm has to prefer built on the type of the dataset. RNN has introduced around the 1980s. However, it got renowned after the invention of LSTM in 1990 to overcome the weaknesses of RNN. The most common use case for RNN is time series problems [11] and natural language processing [12]. Fig. 7 depicts the workflow difference between traditional Feed Forward Neural Networks (FFNN) and RNN.

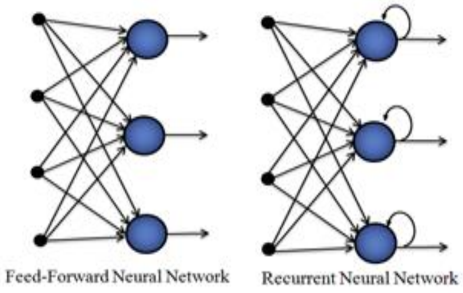


Fig. 7. FFNN vs RNN.

1) Recurrent Neural Network (RNN): RNN is a type of Artificial Neural Network (ANN) specially designed to capture sequential information with the aid of memory cells. This memory cell retains the previous report for further processing, and the decision is based on the prior and current state. RNN shares the same weight parameters within each layer, whereas the traditional neural network shares different weights. There are three crucial components in RNN Input, hidden neuron, and activation function, as described in Fig. 8 and 9.

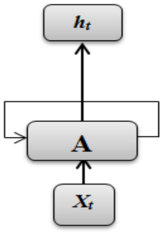


Fig. 8. Simple RNN.

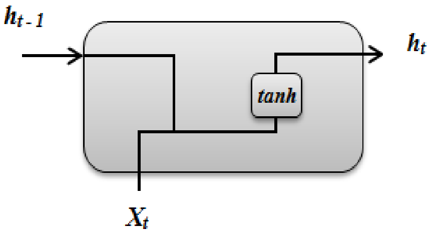


Fig. 9. Internal architecture of simple RNN.

$$h_t = tanh(\mathrm{U}.x_t + W.h_{t-1}) \tag{1}$$

Eq. (1) calculates the hidden state where h_t is a hidden neuron at time t, x_t is the input at time *t, U* is the weight of the hidden layer and *W* is the transition weight of the hidden layer. The input and previous state informations are combined to go through the *tanh*activation function to produce a new hidden state. RNN suffers from the vanishing gradient problem while handling long sequence data. But it is rectified by Long Short-Term Memory[19], another variant of RNN.

2) Long Short Term Memory (LSTM): LSTM is capable of processing long-term dependency data. It manages the previous context more effectively than RNN using three gates. They are the input gate, forget gate, and output gate, as depicted in Fig. 10. The input gate updates the memory cell, forget gate decides whether the information has to be kept or not. The output gate is responsible for determining the next hidden state.The loop structure of RNN and LSTM helps to choose the better weight parameter. The formula for each variable in LSTM is defined below:

$$f_t = \sigma(W_f.X_t + U_f.h_{t-1} + b_f) \tag{2}$$

$$i_t = \sigma(W_i.X_t + U_i.h_{t-1} + b_i) \tag{3}$$

$$S_t = tanh(W_c.X_t + U_c.h_{t-1} + b_c) \tag{4}$$

$$C_t = i_t * S_t + f_t * S_{t-1} \tag{5}$$

$$o_t = \sigma(W_o.X_t + U_o.h_{t-1} + V_o.C_t + b_o) \tag{6}$$

$$h_t = o_t * tanh(C_t) \tag{7}$$

where i_t, f_t, o_t refers to the input gate, forget gate, and out gate respectively. *W, U,* and *V* are the weight matrices, *b* is the bias vectors, X_t is the input vector to the memory cell at time t, h_t is the value of the memory cell at time t, and C_t, S_t are the candidate state and state of the memory cell at time *t*, respectively. Here sigmoid (*σ*) and *tanh* are the activation functions.

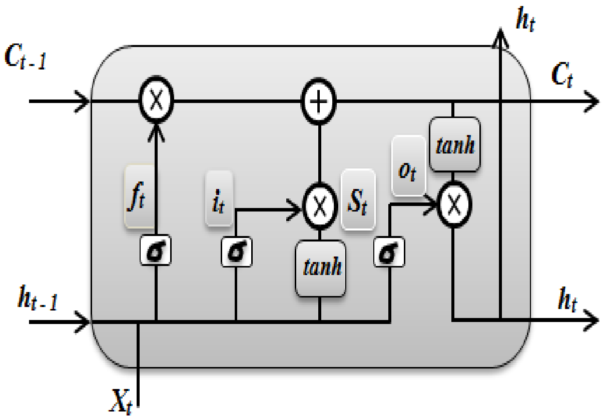


Fig. 10. LSTM architecture.

3) Gated Recurrent Unit (GRU): GRU is a simple version of RNN in terms of architecture. It is uncomplicated to implement and has a quick performance than LSTM, but the functionalities of both architectures are identical. GRU uses fewer parameters, so it requires less hardware and training time. Therefore, GRU attracts the user to involve in many applications. The three gates are reduced into two gates update and reset gate, defined in Fig. 11.

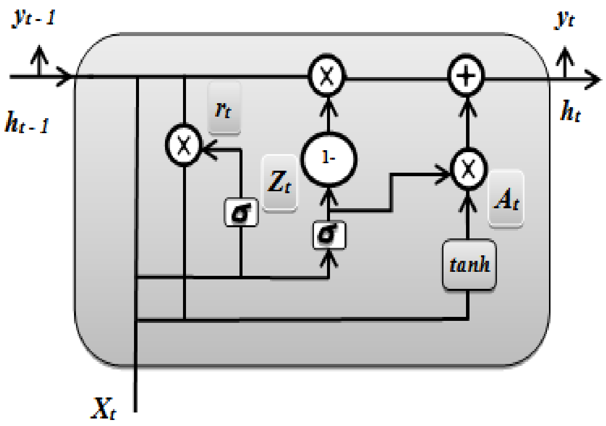


Fig. 11. GRU architecture.

The update gate is a combination of the input, and a forget gate in LSTM. It decides whether the particular information has to be kept or discarded. The reset gate will determine the amount of data that should forget. The following formula defines each variable in GRU:

$$r_t = \sigma(W_r . X_t + U_r . h_{t-1} + b_r) \quad (8)$$

$$Z_t = \sigma(W_z . X_t + U_z . h_{t-1} + b_z) \quad (9)$$

$$A_t = tanh(W_h . X_t + U_h . (r_t * h_{t-1}) + b_h) \quad (10)$$

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * A_t \quad (11)$$

where r_t and Z_t are the two gates for reset and update respectively. A_t is memory content, h_t is the final memory of the current time step and the σ and *tanh* are the activation functions. The two gates have values between 0 and 1 through the sigmoid function (σ). While doing, the memory content (A_t), using the reset gate store the significant information from the previous value between the range −1 to 1 over *tanh*.

4) Metrics used for time-series data: Choosing the right metric is essential to evaluating the model's performance. All the decision, such as tuning the hyper-parameter and selecting the suitable model, is made on the result only. Here the notable thing is before deciding the metrics, need to check the following entities: the nature of the dataset, the values going to handle, and whether there is any need to compare other datasets. If so, are they all on the same scale or different ones? Table III and Table IV illustrate the various metrics available for the time series problem [14]-[18]. Fig. 12 shows the percentage of performance metrics reported in this study.

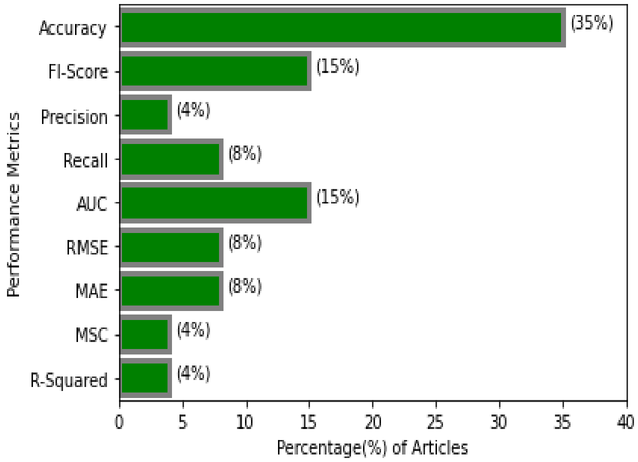


Fig. 12. Percentage of performance metrics applied in this review.

TABLE III. TYPES OF PERFORMANCE METRICS IN REGRESSION PROBLEM

Metrics for Regression			
Scale-Dependent	**Percentage-Error**	**Relative-Error**	**Scale-Free Error**
• Mean Absolute Error(MAE) • Mean Squared Error (MSE) • Root Mean SquaredError (RMSE)	• Mean Absolute Percentage Error(MAPE) • Symmetric Mean Absolute Percentage Error (SMAPE)	• Median Relative Absolute Error (MdRAE) • Geometric Mean Relative Absolute Error (GMRAE)	• Mean Absolute Scaled Error (MASE)
Description			
Error metrics are articulated in the units of the underlying data (Example: Dollars, Inches, etc.)	Scale independent and used to compare forecast performance between different time series	Compare your model's performance with the baseline or benchmark model**.**	Scale the error based on the in-sample MAE from a random walkforecast method
Advantage			
Easy to calculate and interpret	Scale- independency and easy interpretability	Scale-independence	Scale free and suitable metric for time series data with zeros
Disadvantage			
Scale dependency	- Infinite or undefined values for zero or close-to-zero actual values - Heavier penalty on negative than on positive errors - Cannot be used when using percentages make no sense.	When the calculated errors are small it leads to division by zero error.	Not to use when all historical observations are equal or all of the actual values during the in-sample period were zeros.

TABLE IV. TYPES OF PERFORMANCE METRICS IN CLASSIFICATION PROBLEM

Metrics for Classification			
Name	**Description**	**Advantage**	**Disadvantage**
AUC/ROC	The **AUC** measures the entire two-dimensional area under the curve at all possible classification thresholds.ROC is a plot to explain the true and false positive rates.	Using graph representation to show the trade-off between the TPR and FPR.	Not suitable for the highly imbalanced dataset and concentrates only on TPR and FPR.
Confusion Matrix	Identify the model correctness all the way. The Four elements of this table are TP, TN, FP, and FN, which helps to derive the following metrics.	Find the issue where the model failed to understand.	Interpreting the result is complex.
Accuracy	The degree of model correctness. ***Accuracy=(TP+TN)/(TP+FN+TN+FP)***	Easy to interpret	Misleading the result where the sample of minority class is very less.
Precision	Ability of the model to identify only the relevant data points. ***P=TP/(TP+FP)***	Identify the proportion of correct positive identifications	It doesn't consider the type II classification error.
Recall (Sensitivity)	Ability of the model to find all the relevant data points. ***R=TP/(TP+FN)***	Identify the proportion of correct actual positives.	It doesn't consider the type I classification error.
F1-Score	A single score that balances both the concerns of precision and recall in one number. ***F1-Score = 2 *(P*R)/(P+R)***	The harmonic mean of precision and recall value	It is a combined result of precision and recall, so a bit harder to interpret.

TABLE V. LIST OF HYPER-PARAMETERS

Hyper-Parameters	Description
Train/Test ratio	Splitting the dataset into train and test. (Example: 80:20)
Hidden Layer	The layer between input and output and it determines the depth of the neural network (Usually 1 or 2 layers).
Optimizer	It is an algorithm used to update the weight of each layer after each iteration (Example: Gradient descent, Adam [26,27,39,43,45,29,31,32,33])
Learning Rate	It defines how quickly the network updates its parameters (0.0-0.1)
Activation function	Allowing deep learning models to learn non-linear prediction boundaries [22] (Example: Sigmoid [28, 39], ReLU[24,28,32], Tanh[26,32], Leaky Relu[45])
Number of Epochs	Number of iterations to pass the whole dataset in training.
Batch-Size	Number of sample that the network used to update the weights
Momentum	It speeds up the learning process by preventing the oscillation in the convergence of the method.
Weight initialization	It defines the starting point of the optimization.
Dropout	It helps to avoid over-fitting by eliminating the randomly selected neurons in training [26, 27, 28, 41,43, 45].
Regularization	It prevents over-fitting by stopping the weights that are too high(L1,L2) [26]
Units	It determines the level of knowledge that is extracted by each layer.

C. Factors Affecting the Time Series Model Accuracy

There are several factors affecting the model performance which are the techniques used for pre-processing,train-test ratio, and the selection of model hyper-parameters. Table V represents the hyper-parameters that are affecting the model accuracy [23].

1) Pre-processing: Removing unwanted data and filling in the missing values are the initial step inpre-processing. Several methods are available for imputation, such as mean, median,mode, interpolation, weighted average [24], and k-nearest neighbor[43]. Mean and Weighted Average is the widelyused techniques. The first one returns the average value of the feature column, and the second substitute the average of the most frequent information.

The next step is to encode all the categorical information into numerical value for model understanding using any technique such as label encoding or one hot encoding [39]. Each method has merits and demerits of its own. After encoding, re-scaling the data (feature) is very important since it makes the model less sensitive to the scale of features and allows converging with better weights. There are two significant types of scaling: Standardization(z-score) and Normalization (min-max scalar). Standardization assumes that the values are in Gaussian distribution and centered on the zero mean with unit standard deviation. It is less sensitive to outliers, so Karim et al., [34] used z-score normalization to handle the outlier values, and Wang et al., used batch normalization for scaling in [24]. It is specific to each layer and batch of input in the neural network.

2) Train-test split: Normally, 80:20 is the suggested ratio for a train-test split if the samples are distributed evenly across the dataset. Wu et al., [27,39] split the dataset into 80:20 for training and testing. Shin et al., carry the exact ratio in [40], but 10 % for validation from test data. In researches [25,38, 31, 32], the percentage used for training/testing is 70:30; the remaining studies [43, 26, 28, 29] use slightly different ratios.

3) Imbalanced dataset: Most of the real-time dataset is imbalanced and should be handled appropriately to avoid classification errors. In studies [35, 39, 43, 33], the authors used synthetic samples (SMOTE) to balance the target class count. But Mubarak et al.,[28] introduced a cost-sensitive technique in the loss function to avoid type 2 classification error[28]. Dimension reduction is also another issue where the feature count is vast. Waheed et al.,[25] using Singular Value Decomposition (SVD) method to find the top 30 efficient features.

4) Hyper-parameters: The number of hidden layers is significant in deep learning because it shows the complexity of the problem. Bousnguar et al., [41] used three LSTM layers and 50 cells for each layer. Qiu et al.,[42] involved two convolutional and two fully connected layers for binary classification with the sigmoid activation function. Aljohani et al.,[44] applied three LSTM layers, and each layer is assigned 100 to 300 units of neurons. Deep ANN is appliedin [25,45] and uses a minimum of three layers and up to seven hidden layers. Next to hidden layers, select the suitable optimizer to update the weight for every iteration. The Adam optimizer is majorly used [39, 43, 45, 26, 27, 29, 31, 32, 33] among others, such as gradient descent, stochastic gradient descent, and RMSProp.

Then the learning rate (0.0-0.1) assigns the speed of the network parameter update. Frequently used values are 0.0025 [27], 0.001 [29, 32, 33], and 0.1 [31]. The activation function is another hyper-parameter that helps to predict complex non-linear data. This parameter differentiates neural networks compared with machine learning models. Relu, Leaky Relu, tanh, and sigmoid are the activation functions equally used in all the papers. After fitting these parameters training the model by mentioning the number of epochs is mandatory. Sometimes less training, such as 15, 20, and 25, gives better performance than massive iteration [27, 31]. The Dropout is the last layer of the neural network to avoid overfitting, so it is majorly used in all the experiments. The frequent values are 0.1, 0.2, 0.3, and a maximum of 0.5 by Mubarak et al., in [28].

Concerning batch size, the authors adjusted the value to improve the accuracy by doing several experiments. Waheed et al., found the batch size from the value of 64 increased the model performance for all the weeks, but when the batch size was increased additionally from 1364, the model performance degraded with AUC decreasing by a value of 0.04. Regularization is rarely used [26] in the experiment. The model setup does not explain the other parameters, such as Weight initialization and Momentum.

IV. Discussion

This section discuss the contribution of this paper referred to in the introduction.

A. *Finding the Impact of Deep Learning in Educational Time Series Problem*

To answer RQ1, this SLR proved the success of deep learning models in educational time series data retrieved from various sources. Hernández et al.,[21] also confirmed in their review the number of publications recently enriched after raising the application of the DL model. But the first publication commenced in 2015. Section III describes the previous work, and all the information is summarized in Table II to explain the types of models used, the paper's findings, individuality, and the dataset details. The CNN-LSTM is the majorly used hybrid technique, and the LSTM is the widely used single model in many research works. In Education, student performance prediction is the typical use case executed in multiple investigations. Moreover, clustering the time sequence data was also applied to categorize the students based on their performance. Due to its sequential nature, most of the work was done on MOOC online data than the offline mode to predict student dropout.

B. *Identify the Architecture of Time Series Model and How it Differs from the Traditional Approach*

To answer RQ2, Section IV describes the internal structure of RNN, LSTM, and GRU using the required diagrams and formulas. It represents the improvement and differences between each other. Numerous investigations involve LSTM rather than RNN and GRU though the architecture is intricate. Also, LSTM merged with CNN to retrieve spatiotemporal features effectively. Self-connected neurons helps to maintain the previous information, and this is different from feed-forward neural network.

C. *Discover the Significant Factors Affecting the Time Series Model Accuracy*

To answer RQ3, Table V provides information on the factors influencing the model accuracy. Tuning the Neural Network is necessary because it improves the model's performance. The number of hidden layers, epochs, batch size, dropout layer, and optimizers are the commonly used hyper-parameters due to their high impact on the outcome. Most authors use manual selection to pick the best hyperparameters instead of any optimization technique, such as grid search and Bayesian method.

V. Conclusion

Understanding the DL methodology and the previous work done in a particular domain is fundamental before implementing the research idea. This study is the first work that gives a background for young researchers who want to involve Deep Learning in the Education time series problem. Accessed Google Scholar and IEEE Xplore scientific websites to collect relevant research papers. Then the collected documents(n=291) are analyzed manually and selected twenty-two (n=22) papers for this SLR by following PRISMA methodology. The essence of this survey is deep learning applies widely, but the hybrid model gave the highest accuracy than the individual model. Student classification, clustering, forcasting the student enrolment/grade, and dropout prediction using online course log data are the normally used problem statement. Large sequential data are rarely used compared with other domains which helps to avoid complex models. Finally discussed the RNN architecture, types of metrics, and the factors influencing the model accuracy.

VI. Future Perspective

This deliberation clearly explains the previous work done in the educational domain using time series data and will involve all this learning in the implementation work to fill the research gap identified.

References

[1] R. Singh and S. Srivastava, “Stock prediction using deep learning,” *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18569–18584, Dec. 2016, doi: 10.1007/s11042-016-4159-7.

[2] S. Ji, J. Kim, and H. Im, “A Comparative Study of Bitcoin Price Prediction Using Deep Learning,” *Mathematics*, vol. 7, no. 10, p. 898, Sep. 2019, doi: 10.3390/math7100898.

[3] M. O. Alassafi, M. Jarrah, and R. Alotaibi, “Time series predicting of COVID-19 based on deep learning,” Neurocomputing, vol. 468, pp. 335–344, Jan. 2022, doi: 10.1016/j.neucom.2021.10.035.

[4] H. T. Rauf et al., “Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks,” Personal and Ubiquitous Computing, Jan. 2021, doi: 10.1007/s00779-020-01494-0.

[5] P. Hewage, M. Trovati, E. Pereira, and A. Behera, “Deep learning-based effective fine-grained weather forecasting model,” Pattern Analysis and Applications, Jun. 2020, doi: 10.1007/s10044-020-00898-1.

[6] D. Sun, J. Wu, H. Huang, R. Wang, F. Liang, and H. Xinhua, “Prediction of Short-Time Rainfall Based on Deep Learning,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–8, Mar. 2021, doi: 10.1155/2021/6664413.

[7] P. P. Sarkar, P. Janardhan, and P. Roy, “Prediction of sea surface temperatures using deep learning neural networks,” *SN Applied Sciences*, vol. 2, no. 8, Jul. 2020, doi: 10.1007/s42452-020-03239-3.

[8] S. Biswas and M. Sinha, “Performances of deep learning models for Indian Ocean wind speed prediction,” *Modeling Earth Systems and Environment*, vol. 7, no. 2, pp. 809–831, Sep. 2020, doi: 10.1007/s40808-020-00974-9.

[9] A. P, “Higher Education Institution (HEI) Enrollment Forecasting Using Data Mining Technique,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 2060–2064, Apr. 2020, doi: 10.30534/ijatcse/2020/179922020.

[10] B. Siregar, I. A. Butar-Butar, R. Rahmat, U. Andayani, and F. Fahmi, “Comparison of Exponential Smoothing Methods in Forecasting Palm Oil Real Production,” *Journal of Physics: Conference Series*, vol. 801, p. 012004, Jan. 2017, doi: 10.1088/1742-6596/801/1/012004.

[11] A. Graves,”Long short-term memory,” *Supervised sequence labelling with recurrent neural networks*, pp. 37-45, 2012.

[12] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, “Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Oct. 2019, doi: 10.2478/jaiscr-2019-0006.

[13] R. Jayashree, “Enhanced Classification Using Restricted Boltzmann Machine Method in Deep Learning for COVID-19,” *Understanding COVID-19: The Role of Computational Intelligence*, pp. 425–446, Jul. 2021, doi: 10.1007/978-3-030-74761-9_19.

[14] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and practice. Melbourne: OTexts, 2021.

[15] C. Chen, J. Twycross, and J. M. Garibaldi, “A new accuracy measure based on bounded relative error for time series forecasting,” *PLOS ONE*, vol. 12, no. 3, p. e0174202, Mar. 2017, doi: 10.1371/journal.pone.0174202.

[16] V. R. Jose, “Percentage and relative error measures in forecast evaluation,” Operations Research, vol. 65, no. 1, pp. 200–211, 2017, doi:10.1287/opre.2016.1550

[17] S. Kim and H. Kim, “A new metric of absolute percentage error for intermittent demand forecasts,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, Jul. 2016, doi: 10.1016/j.ijforecast.2015.12.003.

[18] V. Cerqueira, L. Torgo, J. Smailović and I. Mozetič, "A Comparative Study of Performance Estimation Methods for Time Series Forecasting," 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 2017, pp. 529-538, doi: 10.1109/DSAA.2017.7.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] S. Vanitha and R. Jayashree, “A Prediction On Educational Time Series Data Using Statistical Machine Learning Model -An Experimental Analysis,” Journal of Theoretical and Applied Information Technology, vol. 100, no. 14, pp. 5189–5200, Jul. 2022.

[21] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, “A Systematic Review of Deep Learning Approaches to Educational Data Mining,” *Complexity*, vol. 2019, pp. 1–22, May 2019, doi: 10.1155/2019/1306039.

[22] A. Farzad, H. Mashayekhi, and H. Hassanpour, “A comparative performance analysis of different activation functions in LSTM networks for classification,” *Neural Computing and Applications*, vol. 31, no.7, pp. 2507–2521, Oct. 2017, doi: 10.1007/s00521-017-32106.

[23] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, “Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework,” *IEEE Access*, vol. 6, pp. 49325–49338, 2018, doi: 10.1109/access.2018.2868361.

[24] X. Wang, P. Wu, G. Liu, Q. Huang, X. Hu, and H. Xu, “Learning performance prediction via convolutional GRU and explainable neural networks in e-learning environments,” *Computing*, vol. 101, no. 6, pp. 587–604, Jan. 2019, doi: 10.1007/s00607-018-00699-9.

[25] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, “Predicting academic performance of students from VLE big data using deep learning models,” *Computers in Human Behavior*, vol. 104, p. 106189, Mar. 2020, doi: 10.1016/j.chb.2019.106189.

[26] A. A. Mubarak, H. Cao, and S. A. M. Ahmed, “Predictive learning analytics using deep learning model in MOOCs’ courses videos,” *Education and Information Technologies*, vol. 26, no. 1, pp. 371–392, Jul. 2020, doi: 10.1007/s10639-020-10273-6.

[27] Q. Fu, Z. Gao, J. Zhou, and Y. Zheng, “CLSA: A novel deep learning model for MOOC dropout prediction,” *Computers & Electrical Engineering*, vol. 94, p. 107315, Sep. 2021, doi: 10.1016/j.compeleceng.2021.107315.

[28] A. A. Mubarak, H. Cao, and I. M. Hezam, “Deep analytic model for student dropout prediction in massive open online courses,” *Computers & Electrical Engineering*, vol. 93, p. 107271, Jul. 2021, doi: 10.1016/j.compeleceng.2021.107271.

[29] H. Waheed, S. U. Hassan, R. Nawaz, N. R. Aljohani, G. Chen, and D. Gasevic, “Early prediction of learners at risk in self-paced education: a neural network approach,” *Expert Systems with Applications*, vol. 213, no. Part A, p. 118868, Mar. 2023, doi: 10.1016/j.eswa.2022.118868.

[30] M. Zhang, X. Du, K. Rice, J.-L. Hung, and H. Li, “Revealing at-risk learning patterns and corresponding self-regulated strategies via LSTM encoder and time-series clustering,” *Information Discovery and Delivery*, vol. 50, no. 2, pp. 206–216, Jun. 2021, doi: 10.1108/idd-12-2020-0160.

[31] K. He and K. Gao, “Analysis of Concentration in English Education Learning Based on CNN Model,” *Scientific Programming*, vol. 2022, pp. 1–10, Jul. 2022, doi: 10.1155/2022/1489832.

[32] A. S. Aljaloud *et al.*, “A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM,” *IEEE Access*, vol. 10, pp. 85255–85265, 2022, doi: 10.1109/access.2022.3196784.

[33] H.-C. Chen *et al.*, “Week-Wise Student Performance Early Prediction in Virtual Learning Environment Using a Deep Explainable Artificial Intelligence,” *Applied Sciences*, vol. 12, no. 4, p. 1885, Feb. 2022, doi: 10.3390/app12041885.

[34] F. Karim, S. Majumdar, and H. Darabi, “Insights Into LSTM Fully Convolutional Networks for Time Series Classification,” *IEEE Access*, vol. 7, pp. 67718–67725, 2019, doi: 10.1109/access.2019.2916828.

[35] F. Chen and Y. Cui, “Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course

Performance," *Journal of Learning Analytics*, vol. 7, no. 2, pp. 1–17, Sep. 2020, doi: 10.18608/jla.2020.72.1.

[36] S. Li and T. Liu, "Performance Prediction for Higher Education Students Using Deep Learning," *Complexity*, vol. 2021, pp. 1–10, Jul. 2021, doi: 10.1155/2021/9958203.

[37] H. Prabowo, A. A. Hidayat, T. W. Cenggoro, R. Rahutomo, K. Purwandari, and B. Pardamean, "Aggregating Time Series and Tabular Data in Deep Learning Model for University Students' GPA Prediction," *IEEE Access*, vol. 9, pp. 87370–87377, 2021, doi: 10.1109/access.2021.3088152.

[38] Asish, S.M., Hossain, E., Kulshreshth, A.K. and Borst, C.W., "Deep Learning on Eye Gaze Data to Classify Student Distraction Level in an Educational VR Environment", In *ICAT-EGVE 2021-International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments,* Vol. 20211326, *https://doi. org/10.2312/egve*.

[39] N. Wu, L. Zhang, Y. Gao, M. Zhang, X. Sun, and J. Feng, "CLMS-Net: dropout prediction in MOOCs with deep learning," *Proceedings of the ACM Turing Celebration Conference - China*, pp.1-6, 2019.

[40] J. Shin, F. Chen, C. Lu, and O. Bulut, "Analyzing students' performance in computerized formative assessments to optimize teachers' test administration decisions using deep learning frameworks," *Journal of Computers in Education*, Aug. 2021, doi: 10.1007/s40692-021-00196-7.

[41] H. Bousnguar, L. Najdi, and A. Battou, "Forecasting approaches in a higher education setting," *Education and Information Technologies*, vol. 27, no. 2, pp. 1993–2011, Aug. 2021, doi: 10.1007/s10639-021-10684-z.

[42] L. Qiu, Y. Liu, Q. Hu, and Y. Liu, "Student dropout prediction in massive open online courses by convolutional neural networks," *Soft Computing*, vol. 23, no. 20, pp. 10287–10301, Oct. 2018, doi: 10.1007/s00500-018-3581-3.

[43] F. Chen and Y. Cui, "Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance," *Journal of Learning Analytics*, vol. 7, no. 2, pp. 1–17, Sep. 2020, doi: 10.18608/jla.2020.72.1.

[44] N. R. Aljohani, A. Fayoumi, and S.-U. Hassan, "Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment," *Sustainability*, vol. 11, no. 24, p. 7238, Dec. 2019, doi: 10.3390/su11247238.

[45] Y. He *et al.*, "Online At-Risk Student Identification using RNN-GRU Joint Neural Networks," *Information*, vol. 11, no. 10, p. 474, Oct. 2020, doi: 10.3390/info11100474.

AI in Tourism: Leveraging Machine Learning in Predicting Tourist Arrivals in Philippines using Artificial Neural Network

Noelyn M. De Jesus[1], Benjie R. Samonte[2]
Assistant Professor, College of Informatics and Computing Sciences[1, 2]
Batangas State University ARASOF-Nasugbu, Nasugbu, Batangas, Philippines [1, 2]

***Abstract*—Tourism is one of the most prominent and rapidly expanding sectors that contribute significantly to the growth of a country's economy. However, the tourism industry has been most adversely affected during the coronavirus pandemic. Thus, a reliable and accurate time series prediction of tourist arrivals is necessary in making decisions and strategies to develop the competitiveness and economic growth of the tourism industry. In this sense, this research aims to examine the predictive capability of artificial neural networks model, a popular machine learning technique, using the actual tourism statistics of the Philippines from 2008-2022. The model was trained using three distinct data compositions and was evaluated utilizing different time series evaluation metrics, to identify the factors affecting the model performance and determine its accuracy in predicting arrivals. The findings revealed that the ANN model is reliable in predicting tourist arrivals, with an *R*-squared value and MAPE of 0.926 and 13.9%, respectively. Furthermore, it was determined that adding training sets that contain the unexpected phenomenon, like COVID-19 pandemic, increased the prediction model's accuracy and learning process. As the technique proves it prediction accuracy, it would be a useful tool for the government, tourism stakeholders, and investors among others, to enhance strategic and investment decisions.**

Keywords—Tourist arrivals; machine learning; predictive analytics; artificial neural network; ANN; time series prediction

I. Introduction

Big data has been utilized by many academics and industry professionals to make their own predictions and forecasting in a variety of fields, including the travel and tourism industries. Tourism is one of the most prominent and rapidly expanding sectors that contribute significantly to the growth of a country's economy. In the Philippines, travel and tourism contributed 12.7%, 5.4%, and 5.2% in the Gross Domestic Product (GDP) in 2019, 2020, and 2021, respectively [1]. This year, the tourism sector achieved another remarkable recognition when it won three significant awards at the 29th World Travel Awards (WTA). According to [2], the country was named "Asia's top tourist attraction," "dive destination," and "top beach destination" among Asian countries. The [3] names dive tourism as one of the important industries that can have a favorable impact on industry growth in terms of more visitors, longer stays, and increased tourism revenue.

However, the direct gross value contributed of the tourism industry has decreased by almost 50% over the last two years as a result of the closing of international borders and national lockdowns.

Thus, a reliable and accurate prediction of tourist arrival is necessary in making decisions and strategies to develop the competitiveness and economic growth of the tourism industry. In the previous study of [4], they investigated the connection between the tourist arrivals from the quantity of COVID-19 cases during summer. The analysis uses three models: the simple linear model, the negative binomial regression model, and the Cognitive Artificial Neural Network (ANN) model. The findings showed that tourism is a significant factor of the increase in Covid-19 cases. The researchers also concluded that the ANN model can make the most accurate predictions among the three models.

The random forest (RF), artificial neural networks (ANN) [5], and support vector machine are three of the most popular AI-based models for time-series prediction. However, many publications, including [6,7,8], claim that there is no one technique that consistently delivers the best tourism forecasts; rather, the results rely on the model and technique applied, the amount of observations, features of the data set, and predicted duration.

This study's objective is to forecast future tourist arrivals using historical data rather than to examine demand for tourist attractions. As has already been mentioned, seasonal times-series forecasting is crucial for making strategic decisions and organizing upcoming tasks. The researchers want to raise concerns about neural network models' capability for predicting seasonal tourist arrivals, which is a challenge that shows frequently in a variety of applications. In addition, this study aims to identify the factors affecting the model performance and determine its accuracy in tourist arrivals' prediction.

The following are the primary contributions of this study:

- An exploratory data analysis that delivers more meaningful data through visualization using a dataset that includes information about Philippine tourist arrivals.
- In contrast to other studies that concentrated on and employed a single data composition in training and testing sets, the researchers explore with several data

compositions to assess the predictive capability of ANN models.

- Leveraging machine learning, particularly artificial neural networks model in alternative with other popular data mining techniques like random forest and support vector machine, in predicting tourist arrivals in the Philippines and perform model performance evaluation using various time series evaluation metrics.

II. Related Works

Machine learning and data science researchers have been working on time series forecasting. There is a growing need for precise and effective forecasting techniques as time series data from numerous industries, including banking, healthcare, and energy, become more widely available. In this study, the researchers highlight a few recent developments in time series forecasting and talk about their advantages and disadvantages.

Deep learning is one of the most well-liked methods for time series forecasting and has demonstrated promising outcomes in a number of fields. In order to capture long-term relationships in the data, [9] introduced a unique deep learning architecture for time series forecasting dubbed Longformer-TS. This design makes use of the self-attention mechanism. On numerous benchmark datasets, they showed that Longformer-TS outperformed a number of cutting-edge models.

The introduction of probabilistic forecasting techniques, which offer a distribution of potential future values rather than a single point estimate, is another new development in time series forecasting. This might be helpful when making decisions that call for quantifying uncertainty and assessing risk. In order to model the uncertainty in the data, [10] for instance suggested a deep probabilistic forecasting framework that makes use of a Bayesian neural network. On real-world electrical load forecasting tests, they demonstrated that their system beat numerous conventional and deep learning approaches.

There has been an increase in interest in applying meta-learning techniques for time series forecasting in addition to deep learning and probabilistic approaches. By utilizing prior experience on related tasks, meta-learning seeks to discover the best algorithm or hyperparameters for a current task. Using a neural network to train a set of hyperparameters that can adapt to various forms of time series data, [11] suggested a meta-learning framework for time series forecasting, for instance. On a number of benchmark datasets, they showed that their method delivered state-of-the-art performance.

For predicting tourist arrivals, [12] introduced a neural network model termed the Deep Travel Demand Forecasting Network (DTDNet). A prediction module and a feature learning module make up the model's two primary parts. While the prediction module employs a Long Short-Term Memory (LSTM) network to identify temporal connections and generate predictions, the feature learning module employs a deep convolutional neural network (CNN) to extract pertinent features from the raw data. On three tourism datasets from various areas, the authors examined the DTDNet model and found that it beat other cutting-edge models, including ARIMA, SARIMA, and VAR.

A hybrid neural network model that combines a convolutional neural network (CNN) and a recurrent neural network (RNN) for forecasting tourist arrivals was proposed by [13]. The data is processed using the CNN to extract spatial features, and the RNN to identify temporal connections. The model was tested on a Chinese tourism dataset, and the results demonstrated that it beat numerous conventional models like ARIMA and exponential smoothing.

Another study, [14] suggested a neural network model for anticipating tourist arrivals termed the Multivariate Attention-based Temporal Convolutional Network (MATCN). The model combines a temporal convolutional network (TCN) and an attention mechanism to identify both short- and long-term dependencies in the input. On a Thai tourist dataset, the authors tested the MATCN model and found that it performed better than numerous conventional models, including ARIMA and Holt-Winters.

III. Methodology

Datasets for the study are mainly collected from three Philippines government organizations, i.e., the Tourism Department and Development Planning of Department of Tourism, Research and Information Management, and Statistics, Economic Analysis, and Information Management Division. The data collected from Philippines's tourism demand statistics include the annual number of inbound tourists' arrivals to Philippines from January 2008 to October 2022 [15].

A. Descriptive Analysis

Inbound travel to the Philippines increased dramatically between 2008 and 2021 (Fig. 1). From 3.14 million in 2008 to 8.26 million in 2019, there were more than twice as many foreign and overseas Filipino tourists visiting the country [16].

Despite the fact that the graph indicates an increase in inbound tourists' arrivals to the Philippines over the past years, as forecast indicate, there were notable drops in tourist arrival in 2009, 2014 and 2020. In 2009, there was a decrease in the number of arrivals by 1.5% compared to 2008 [17]. Several upsetting incidents occurred in the global tourism industry in 2014, such as the emergence of SARS, avian influenza, Ebola, and MERS-CoV infections [18]. Evidently, a sudden decline in tourist arrivals and subsequent drop in tourism demand of the country caused by coronavirus pandemic in the year 2020, which led to the loss of millions of jobs, severe economic hardship, and the demise of many enterprises [19]. Several jobs and businesses, particularly the micro small, and medium-sized ones that catered to tourists or were related industries, were at risk due to the border closures, main entry points, and hotels as well as mass gatherings restrictions, land travel, and related services worldwide.

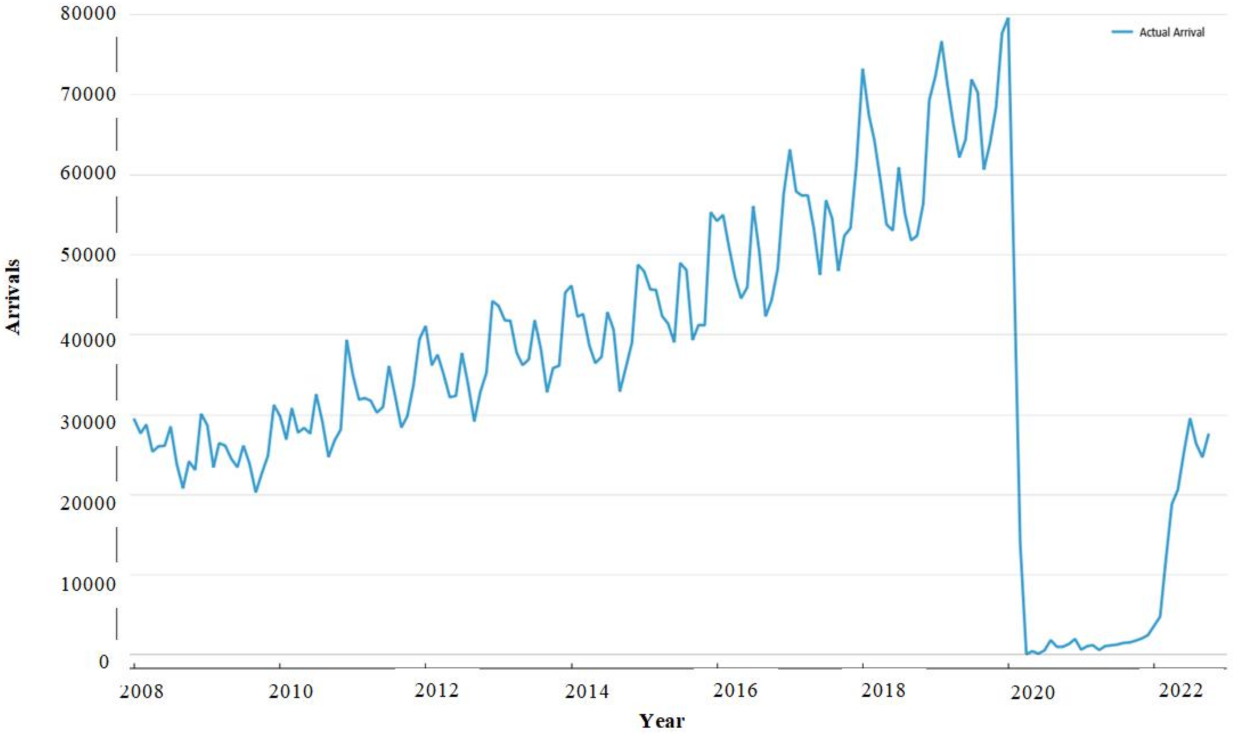


Fig. 1. Number of inbound tourists' arrivals to Philippines and annual grow rate in arrivals (2008-2022). Source: Statistics, Economic Analysis and Information Management Division (SEAIMD) and Department of Tourism (DoT).

Travel and tourism competitiveness in the Philippines continues to rise globally, continuing a trend that has been present for the past eleven years [20]. In 2019, 8.3 million tourists (including non-resident overseas Filipinos) visited the country, primarily from Korea, China, the USA, Japan, and Taiwan. These made up 70% of all visitors during the year. Tourist arrivals in the Philippines, however, fell by 80% in the first quarter of 2020 and by about 90% in 2021 for both foreign and overseas Filipinos, these are the periods when most countries started implementing travel restrictions and lockdowns [21].

B. Methodology

Fig. 2 shows the adapted and modified research framework of the study [22]. It involves collection and preprocessing of data, feature extraction, model development, and its performance evaluation. The researchers first gathered tourist arrival statistics from the Department of Tourism's website. Then, the data was preprocessed in the second phase to extract valuable and target information. Third, train and test the time series machine learning model, and finally, evaluate the model performance.

1) Data collection: To perform this study, the researchers collected the actual inbound tourist arrivals to Philippines between 2008-2022 from the Department of Tourism's official website. The datasets were analyzed and visualized using exploratory data analysis in Python programming. This was done to determine what the tourism data could reveal beyond the formal modeling [23]. Table I presents the descriptive statistics of annual inbound tourist arrivals in the Philippines from 2008 to 2022.

TABLE I. DESCRIPTIVE STATISTICS OF THE DATASETS

Variable	Year	Mean	Std. Dev.	Min	Max
Tourist arrivals	2008	261,618.50	28,386.64	208,167	301,175
	2009	251,424.92	28,750.71	202,822	312,132
	2010	293,372.58	37,521.50	247,191	393,585
	2011	326,454.50	30,288.81	284,040	394,567
	2012	356,067.58	41,311.19	291,637	442,088
	2013	390,108.92	37,505.80	328,114	452,650
	2014	402,780.67	45,120.36	328,981	487,654
	2015	446,723.50	48,193.41	390,486	553,002
	2016	497,250.42	51,174.38	422,943	576,638
	2017	552,489.58	47,494.22	474,854	631,639
	2018	597,372.25	73,046.16	518,041	732,506
	2019	688,409.42	55,021.46	606,553	776,798
	2020	123,544.58	249,612.96	948	796,164
	2021	13,656.58	5,339.33	6,036	24,353
	2022	157,918.67	117,900.48	-	295,650

Source: Statistics, Economic Analysis and Information Management Division (SEAIMD) and Department of Tourism (DOT)

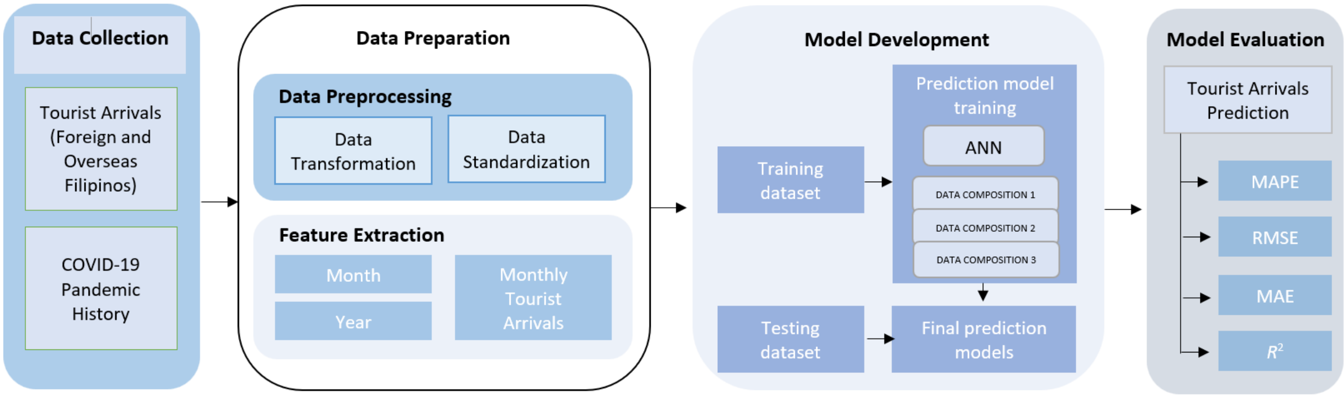


Fig. 2. Adapted research framework of the study.

2) Data preparation: Two temporal features were extracted in the preprocessing stage: month and year. The extracted data was pre-processed using Long-Short-Term Memory (LSTM) as the extraction feature technique, capturing the complex dependencies between the past and current tourist arrivals. The study conducted the time series analysis following the theory of Autoregressive model (Equation 1). The model uses observations to predict a value based on prior time steps as an input. The order of the model is determined by the number of samples utilized for prediction, n_r [24].

$$x\,[k] = e[k] \sum_{r=1}^{n_r} a_r x[k = r] \tag{1}$$

AR model was translated into a neural network, where there is a need of inputs that are lagged. In order to that, the data set was pre-processed, in such as a way that input values was created as lagged values. From the original dataset having a month and tourist arrivals column, the last 12 months was taken from the seasonal time series of the dataset and was used as lags. It will be used as inputs to predict the current month. The first twelve months, January through December 2008, were eliminated, and a new dataset that began in January 2009 and ended in October 2022 was established with twelve additional dummy variables. Then, a traditional neural network was formed, where there are twelve features and the target variable which is the number of tourist arrivals.

3) Model development: To build and develop the model, train and test sets were split into two parts. The criteria for data decomposition in this study is the trend where the tourist arrivals show an upward and downward movements, particularly during the COVID-19 pandemic. In the training process, the segments of the training dataset were decomposed into three partitions such as: (a) January 2008 – January 2020 with 82% as the training data, and 18% as the testing data, (b) January 2008 – March 2020 with 83% as the training data, and 17% as the testing data, and (c) January 2008 – December 2020 with 88% as the training data, and 12% as the testing data (Fig. 3):

- January 2020 (*the period when the government confirmed the first case of new coronavirus*)

In the middle of January 2020, the Philippines reported its first coronavirus case [25] [26].

- March 2020 (*the period when the government implemented suspension of arrivals*)

According to the Bureau of Immigration (BI), entry into the Philippines is prohibited as of March 22, 2020, with the exception of foreign spouses and children of Filipino citizens, diplomats, international organizations' employees and officials [27].

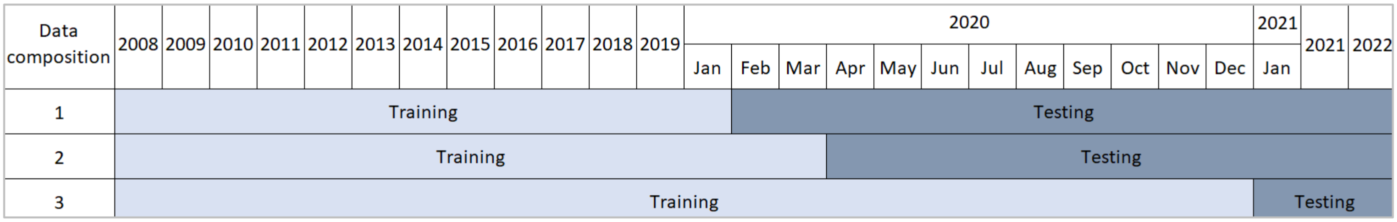
- December 2020 (*the period when the restrictions to main entry point amid new COVID-19 strain was implemented by the government*).

The prohibition on travelers from 20 countries was made public by the Bureau of Immigration (BI) on December 29, 2020, as an additional step to stop the spread of the COVID-19 virus's alleged new strain [28].

The datasets were loaded into Orange Data Mining tool for time series prediction using artificial neural network model based on the architecture of a MLP network. Orange is an open-source data mining toolkit which provides a platform for data visualization and predictive modeling [29].

The researchers adapted the architecture of multi-layer perceptron (MLP), which serve as a supplement to the feed forward neural network. As depicted in Fig. 4, it has three distinct types of layers such as input, hidden, and output layers; receiving the input signal for processing takes place at the input layer. The output layer completes the necessary task, such as classification and prediction [30].

The multilayer perceptron is a feedforward-style design that is based on the perceptron neuron model, which is one of the most popular topologies for forecasting time series [31].



Data composition	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020												2021	2021	2022
													Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan		
1	Training													Testing													
2	Training															Testing											
3	Training																								Testing		

Fig. 3. Composition of fixed training and test sets.

4) Model evaluation: The trained network's predictions were compared against the test forecast set to assess the error. On the test dataset, the network was then simulated.

Evaluation of the Artificial Neural Networks model with three distinct data composition is shown in this section. Each model's prediction accuracy was assessed using mean absolute percentage error (MAPE: Equation (2)), coefficient of determination (R^2: Equation (3)), mean absolute error (MAE: Equation (4)), and root mean squared error (RMSE: Equation (5)) [32, 33]. The MAPE is used to check how close estimates or forecasts are to actual values. Both RMSE and MAE calculate the size of a set of predictions' errors. The primary purpose of the R^2 is to assess how similar the predicted and actual time series are [34].

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (2)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3)$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|Y_i - X_i|}{n} \quad (4)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i)^2} \quad (5)$$

The MSE, RMSE, MAE, MAPE and R^2 determined with the predicted values over the test set are presented for the three data compositions. Furthermore, the selections are based on the best R^2 and lowest MAPE because it shows whether or not the model is a good fit for the observed values, as well as how good of a fit is.

IV. Results and Discussions

Considering the results presented in Table II, the adapted Artificial Neural Networks (ANN) model with the third data composition was the best performing model for predicting tourist arrivals across different data compositions, which clearly shows that ANN is a reliable model in time series prediction [35].

The results show that the prediction models trained with the third training set have the highest coefficient of determination of 0.926 and the lowest MAPE with 13.9%, meaning the forecasted value is closer to the actual one [36]. These results show that the models that were trained with enough data to cover unexpected events like corona virus pandemic, will improve the model accuracy. [37] As the last study pointed out, researchers need to come up with predicting models that can take into account of unplanned scenarios [38]. Overall, the best predictor is the ANN model that uses the data composition which consists of the data of COVID-19 pandemic period.

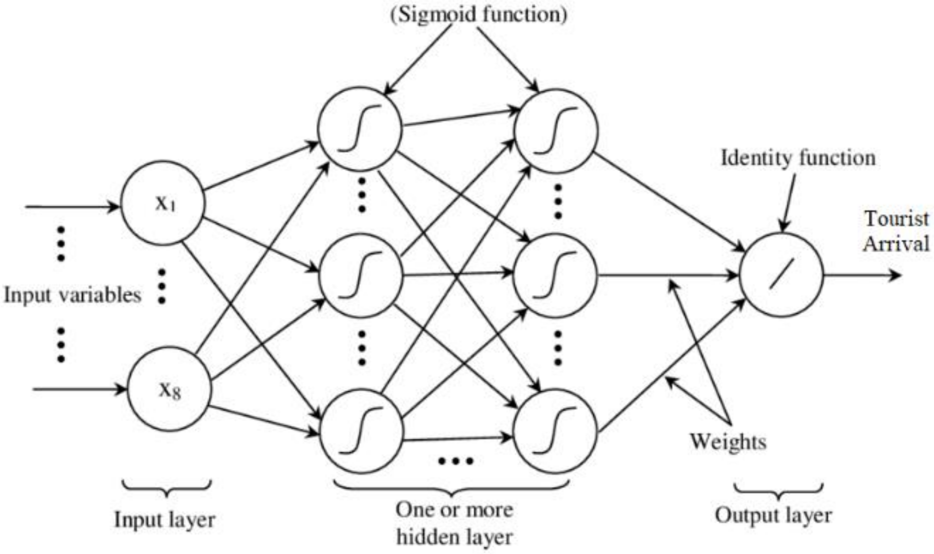


Fig. 4. Multilayer perceptron – artificial neural networks architecture.

TABLE II. COMPARISON OF THE ANN MODEL PERFORMANCE USING DISTINCT DATA COMPOSITIONS

Data Composition	RMSE	MAE	MAPE	R^2
Data Composition 1	333,830.43	255,944.55	543.8%	-6.857
Data Composition 2	156,101.00	113,327.23	250.2%	-1.414
Data Composition 3*	29,814.31	25,254.74	*13.9%*	*0.926*

Note(s): The italicized figures show the results with the best fit across various data compositions, and the * denotes the model with the best performance across various data compositions.

Fig. 5 to 7 portrays the comparison of actual and predicted values of monthly number of inbound tourist arrivals with all data compositions using Artificial Neural Networks model.

The blue line represents the actual tourist arrival, while the predicted value is represented by a red line. Similar to the result in the study of [39], the outputs portrayed in Fig. 7 confirm the competence of ANN model in predicting tourist arrivals in the Philippines. Another researcher conforms that the ANN model has the capacity to outperform other time series machine learning techniques, like ARIMA models [40, 41], Box-Jenkins and Exponential Smoothing models [42].

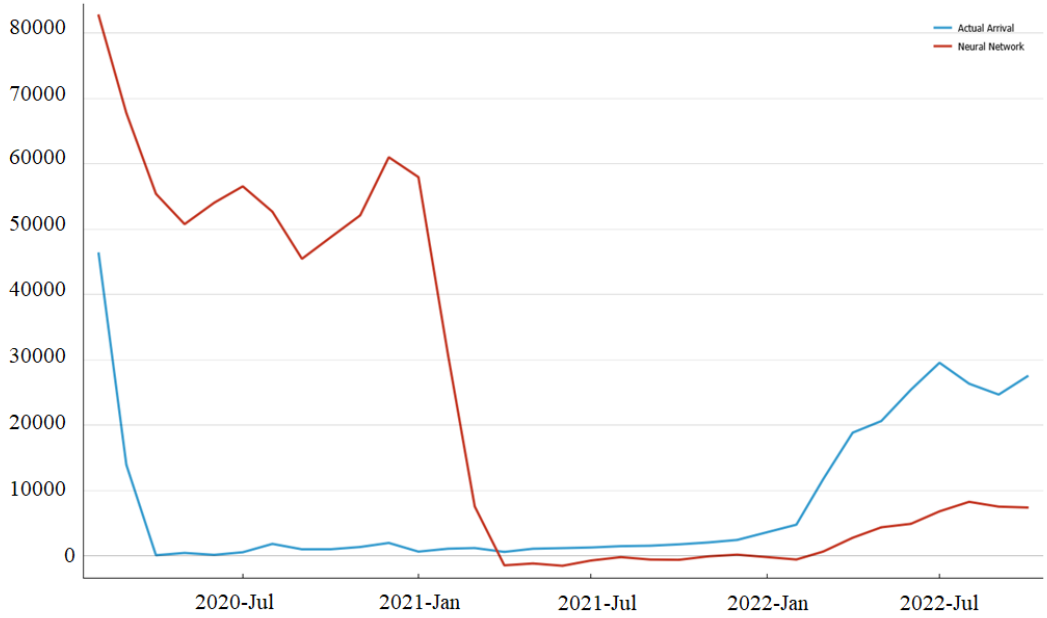


Fig. 5. The actual and predicted values of tourists' arrivals using ANN of first data composition. Source: Author's elaboration.

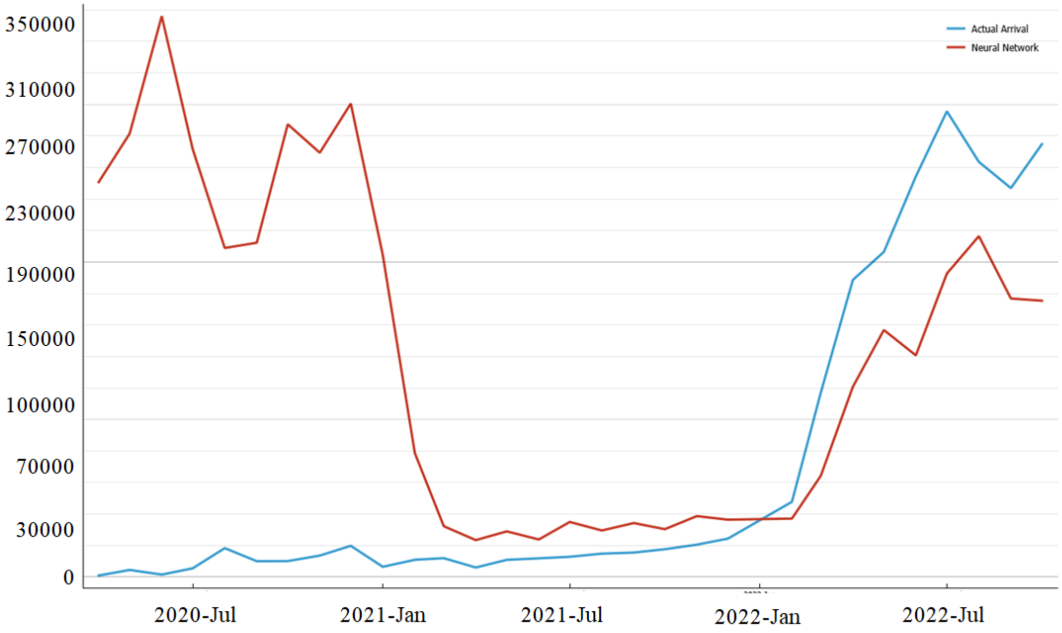


Fig. 6. The actual and predicted values of tourists' arrivals using ANN of second data composition. Source: Author's elaboration .

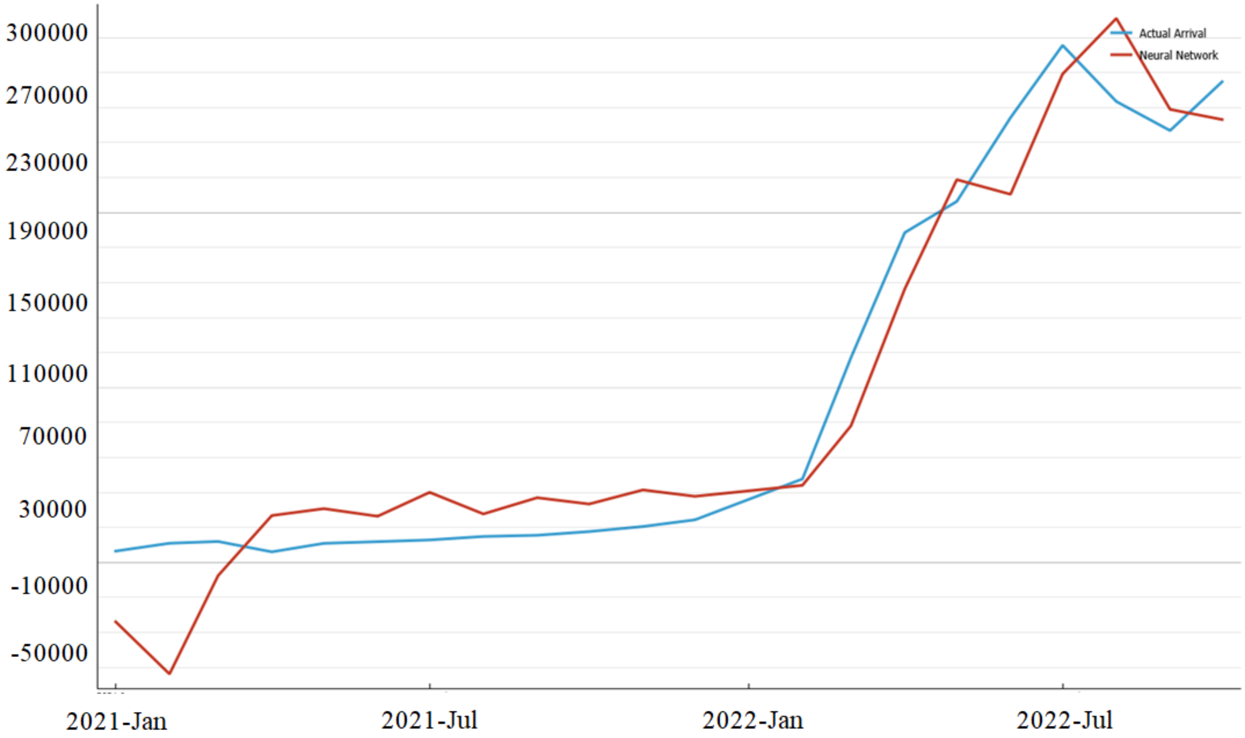


Fig. 7. The actual and predicted values of tourists' arrivals using ANN of third data composition. Source: Author's elaboration.

V. Conclusion

This study compares three distinct data composition in training data sets of Philippines' tourism demand from 2008-2022 using Artificial Neural Network model. The first training set consists of the period when the government confirmed the first case of new coronavirus; the second set is the period when the government implemented suspension of arrivals, and finally, the period when the restrictions to main entry point amid new COVID-19 strain was implemented by the government. To determine the best model and to see whether or not the model is a good fit for the observed values, as well as how good of a fit is in predicting and forecasting the arrivals, it was evaluated utilizing different time series evaluation metrics namely, mean absolute percentage error (MAPE), coefficient of determination (R^2), mean absolute error (MAE), and root mean squared error (RMSE).

The findings showed that the ANN model is reliable in predicting tourist arrivals, with an R-squared value and MAPE of 0.926 and 13.9%, respectively. Furthermore, it was determined that adding training sets that contain the unexpected phenomenon, like COVID-19 pandemic, increased the prediction model's accuracy and learning process. As the technique proves it prediction accuracy, it would be a useful tool for the government, tourism stakeholders, and investors among others, to enhance strategic and investment decisions.

VI. Future Work

To further the study, a combination of neural networks with fuzzy logic or other time series forecasting models for more reliable and accurate results. In addition, other external factors like online forums, reviews in travel Apps, and social media posts, can be added to further enrich the dataset.

Acknowledgment

The authors would like to extend their profound gratitude and sincere appreciation to the Editors and anonymous reviewers for their valuable and constructive feedbacks to further improve the study.

References

[1] Share of Tourism to GDP Is 5.2 Percent in 2021 | *Philippine Statistics Authority*.

[2] "PH Outclasses Other Asian Countries in Tourism." *Manila Bulletin*, 10 Sept. 2022.

[3] Department of Tourism. *National Tourism Development Plan 2016-2022*. 2010. Accessed November 5, 2022.

[4] Casini, Luca, and Marco Roccetti. "A Cross-Regional Analysis of the COVID-19 Spread during the 2020 Italian Vacation Period: Results from Three Computational Models Are Compared." *Sensors*, vol. 20, no. 24, Dec. 2020, p. 7319. *DOI.org (Crossref)*, https://doi.org/10.3390/s20247319.

[5] Atienza, John Robert D., et al. "A Deep Neural Network in a Web-based Career Track Recommender System for Lower Secondary Education." *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*. IEEE, 2022.

[6] Önder, I. (2017). Forecasting Tourism Demand with Google Trends: Accuracy Comparison of Countries versus Cities. International Journal of Tourism Research 19, 648–60. doi: 10.1002/jtr.2137

[7] Zhang, Y., Li, G., Muskat, B. & Law, R. (2020). Tourism Demand Forecasting: A Decomposed Deep Learning Approach. Journal of Travel Research, 1-17. doi: 10.1177/0047287520919522

[8] Höpken, W., Eberle, T., Fuchs, M. & Lexhagen, M. (2020). Improving Tourist Arrival Prediction: A Big Data and Artificial Neural Network Approach. Journal of Travel Research, 1–20. doi:10.1177/0047287520921244

[9] Zhang, J., Xie, W., Ma, X., Liu, Y., Liu, W., & Zhao, J. (2021). Longformer-TS: A transformer-based architecture for time series forecasting. arXiv preprint arXiv:2104.11178.

[10] Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2021). Deep probabilistic forecasting with Bayesian neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(4), 1434-1446.

[11] Ravi, S., & Larochelle, H. (2021). Meta-learning for time series forecasting. arXiv preprint arXiv:2103.02951.

[12] Zhou, H., & Yu, J. (2021). DTDNet: Deep travel demand forecasting network for tourism demand prediction. IEEE Transactions on Neural Networks and Learning Systems, 32(3), 1001-1013.

[13] Chen, W., Zhou, Y., & Wang, Y. (2020). A hybrid neural network model for tourist arrivals forecasting. Journal of Travel Research, 59(7), 1199-1215.

[14] Hu, Q., Xu, J., Hu, L., & Zhang, J. (2020). Forecasting tourism demand with multivariate attention-based temporal convolutional network. Tourism Management, 81, 104130.

[15] Statistics, Economic Analysis, and Information Management Division, The Office of Tourism Development Planning, Research and Information Management, and The Tourism Department of Department of Tourism. Tourism Demand Statistics. Accessed November 5, 2022.

[16] Statistics, Economic Analysis, and Information Management Division. *2019 Economic Newsletter: Philippine Tourism Statistics*.; 2019.

[17] Background (TMTR) - Arangkada Philippines. Accessed November 5, 2022.

[18] Office of the President of the Philippines. Manila: Malacañang Records Office. Executive Order No. 168, s. 2014 | GOVPH. Official Gazette of the Republic of the Philippines. Published 2014. Accessed November 5, 2022.

[19] Helble M, Fink A. *Reviving Tourism amid COVID-19 Pandemic*. 0 ed. Asian Development Bank; 2020. doi:10.22617/BRF200245-2

[20] Lauren Uppink Calderwood, Maksim Soshkin. *The Travel & Tourism Competitiveness Report 2019: Travel and Tourism at a Tipping Point*. World Economic Forum's Platform for Shaping the Future of Mobility; 2019.

[21] Kristhel Anne M. Caynila, Katherine T. Luna, Sarah Amabelle A. Milla. Economic News Letter No. 22-02 - THE PHILIPPINE TOURISM SECTOR AMID THE PANDEMIC: Developments and Prospects.; 2022.

[22] Andariesta, Dinda Thalia, and Meditya Wasesa. "Machine Learning Models for Predicting International Tourist Arrivals in Indonesia during the COVID-19 Pandemic: A Multisource Internet Data Approach." *Journal of Tourism Futures*, Jan. 2022. *DOI.org (Crossref)*, https://doi.org/10.1108/JTF-10-2021-0239.

[23] Exploratory data analysis. In: *Wikipedia*.; 2022. Accessed December 3, 2022.

[24] Gandhi V. Interfacing Brain and Machine. In: *Brain-Computer Interfacing for Assistive Robotics*. Elsevier; 2015:7-63. doi:10.1016/B978-0-12-801543-8.00002-8

[25] ABS-CBN News. Philippines confirms first case of new coronavirus. Accessed November 5, 2022.

[26] GMA News Online. DOH recommends declaration of public health emergency after COVID-19 local transmission. Accessed November 5, 2022.

[27] Philippines: Bureau of Immigration. Press Release: BI to Implement DFA Circular on Suspension of Arrivals.; 2020.

[28] Philippines: Bureau of Immigration. Press Release: BI to Implement Extension of Travel Ban amid New COVID-19 Strain.; 2020.

[29] Demsar J, Curk T, Erjavec A, et al. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*. 2013; 14:2349-2353.

[30] Abirami, S., and P. Chitra. "Energy-Efficient Edge Based Real-Time Healthcare Support System." *Advances in Computers*, vol. 117, Elsevier, 2020, pp. 339–68. *DOI.org (Crossref)*, https://doi.org/10.1016/bs.adcom.2019.09.007.

[31] Borghi, Pedro Henrique, et al. "A COVID-19 Time Series Forecasting Model Based on MLP ANN." *Procedia Computer Science*, vol. 181, 2021, pp. 940–47. *DOI.org (Crossref)*, https://doi.org/10.1016/j.procs.2021.01.250.

[32] Durgapal, Ayushman, and Vrince Vimal. "Prediction of Stock Price Using Statistical and Ensemble Learning Models: A Comparative Study." *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, 2021, pp. 1–6. *DOI.org (Crossref)*, https://doi.org/10.1109/UPCON52273.2021.9667644.

[33] Panda, Manaswinee Madhumita, et al. "Forecasting Foreign Currency Exchange Rate Using Convolutional Neural Network." *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022. *DOI.org (Crossref)*, https://doi.org/10.14569/IJACSA.2022.0130272.

[34] Lendave V. A Guide to Different Evaluation Metrics for Time Series Forecasting Models. Analytics India Magazine. Published November 1, 2021. Accessed November 5, 2022. https://analyticsindiamag.com/a-guide-to-different-evaluation-metrics-for-time-series-forecasting-models/

[35] Abraham ER, Mendes dos Reis JG, Vendrametto O, et al. Time Series Prediction with Artificial Neural Networks: An Analysis Using Brazilian Soybean Production. *Agriculture*. 2020;10(10):475. doi:10.3390/agriculture10100475

[36] Maliberan, Ruby Mae Ebuna. "Forecasting Tourist Arrival in the Province of Surigao Del Sur, Philippines Using Time Series Analysis." *JOIV: International Journal on Informatics Visualization*, vol. 3, no. 3, Aug. 2019. *DOI.org (Crossref)*, https://doi.org/10.30630/joiv.3.3.268.

[37] Qiu RTR, Wu DC, Dropsy V, Petit S, Pratt S, Ohe Y. Visitor arrivals forecasts amid COVID-19: A perspective from the Asia and Pacific team. *Annals of Tourism Research*. 2021; 88:103155. doi: 10.1016/j.annals.2021.103155.

[38] Nguyen, Le Quyen, et al. "Analyzing and Forecasting Tourism Demand in Vietnam with Artificial Neural Networks." *Forecasting*, vol. 4, no. 1, Dec. 2021, pp. 36–50. *DOI.org (Crossref)*, https://doi.org/10.3390/forecast4010003.

[39] Adebiyi, Ayodele Ariyo, et al. "Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction." *Journal of Applied Mathematics*, vol. 2014, 2014, pp. 1–7. *DOI.org (Crossref)*, https://doi.org/10.1155/2014/614342.

[40] Ma, Qihang. "Comparison of ARIMA, ANN and LSTM for Stock Price Prediction." *E3S Web of Conferences*, edited by S.O. Oladokun and S. Lu, vol. 218, 2020, p. 01026. *DOI.org (Crossref)*, https://doi.org/10.1051/e3sconf/202021801026.

[41] Siami-Namini, Sima, et al. "A Comparison of ARIMA and LSTM in Forecasting Time Series." *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2018, pp. 1394–401. *DOI.org (Crossref)*, https://doi.org/10.1109/ICMLA.2018.00227.

[42] Ensafi, Yasaman, et al. "Time-Series Forecasting of Seasonal Items Sales Using Machine Learning – A Comparative Analysis." *International Journal of Information Management Data Insights*, vol. 2, no. 1, Apr. 2022, p. 100058. *DOI.org (Crossref)*, https://doi.org/10.1016/j.jjimei.2022.100058.

A Cloud and IoT-enabled Workload-aware Healthcare Framework using Ant Colony Optimization Algorithm

Lu Zhong[1], Xiaoke Deng[2]
Chongqing Youth Vocational &Technical College; Chongqing, 400712, China[1]
Chongqing Communication Design Institute Co.,Ltd; Chongqing , 400050 , China[2]

***Abstract*—In recent years, smart cities have gained in popularity due to their potential to improve the quality of life for urban residents. In many smart city services, particularly those in the field of smart healthcare, big healthcare data is analyzed, processed, and shared in real time. Products and services related to healthcare are essential to the industry's current state, which increases its viability for all parties involved. With the increasing popularity of cloud-based services, it is imperative to develop new approaches for discovering and selecting these services. This paper follows a two-stage process. The first step involves designing and implementing an Internet-enabled healthcare system incorporating wearable devices. A new load-balancing algorithm is presented in the second stage, based on Ant Colony Optimization (ACO). ACO distributes tasks across virtual machines to maximize resource utilization and minimize makespan time. In terms of both makespan time and processing time, the proposed method appears to be more efficient than previous approaches based on statistical analysis.**

Keywords—Internet of things; cloud computing; healthcare; load balancing

I. INTRODUCTION

Recent advancements in Internet of Things (IoT) [1, 2], Blockchain [3], 5G connectivity [4, 5], cloud computing [6], smart grids [7], optical networks [8, 9], machine learning [10], and artificial intelligence [11, 12] have resulted in exponential growth in data usage across a wide variety of engineering and commerce fields. Embedded systems and Internet-enabled devices have given rise to a new technology called the IoT [13]. IoT allows physical objects, data, and virtual environments to interact. In a prosperous digital society, IoT is used in various applications for data acquisition, such as smart communications, smart healthcare, and smart cities [14]. IoT-enabled medical sensors and tools pose several potential research opportunities in medical and health care [15]. There is a mutual dependency between IoT and cloud computing. By combining both, they provide real-time health data to medical professionals and caregivers at the remote site [16]. Cloud resources and capabilities enable IoT to overcome technological limitations. In contrast, IoT can benefit the cloud by adding real-life items to its scope and distributing and dynamically delivering many new services [17].

Smart environments can be extended to develop new applications and services using IoT-centric-cloud architectures. Combining cloud computing with IoT technology is far more efficient than regular cloud-based applications. This combination of technologies can be used in emerging medical, military, and banking applications [18]. Cloud-based IoT is particularly useful in medical applications that require tracking and remote access to records [19]. IoT-driven healthcare applications collect data on frequent changes in health parameters and continuously update their severity. The IoT devices and sensor readings associated with medical parameters will also be used to diagnose diseases on time before they become severe [20]. There is an increasing amount of data being created and stored for a long period of time today [21].

IoT plays a vital role in healthcare applications. Three phases are included in this criterion: treatment, monitoring remotely, and awareness of context. Patients in the hospital, especially those in intensive care, require constant monitoring and close attention to respond to potential crises and save lives. Physiological data is collected using sensors, analyzed in the cloud, and sent to caregivers (parents, nurses) for further analysis [22]. The sensors collect and analyze the flow of data collected by many health professionals to examine patients according to their specialties. This makes identifying the emergency conditions of patients at risk (Urgent or emergent surgery patients, cardiac patients, etc.) much easier [23]. Many real-world applications rely on remote monitoring. People worldwide are currently suffering from the lack of effective healthcare monitoring. People with chronic illnesses, such as the elderly and children, must be examined almost daily. Remote monitoring eliminates the need to make rounds to the hospital for checkups. Caregivers can diagnose early and intervene earlier with remote access sensors. Healthcare IoT applications require context awareness as a key criterion. As it can determine the condition of the patient and the environment, this technology is extremely useful in comprehending changes in the health status of these patients. Furthermore, changing the patient's physical state could increase his/her vulnerability to diseases and lead to health deterioration [24].

The paper is divided into two stages. In the first stage, an Internet-based healthcare model is designed and implemented, which involves wearable devices attached to the patients' bodies. The second stage proposes a new method based on Ant Colony Optimization (ACO) algorithm to solve the load balancing problem. In order to maximize resource utilization and minimize makespan time, the ACO distributes tasks over the available virtual machines. IoT provides a basis for next-generation applications and services by supporting and developing foundational technologies. Likewise, cloud computing provides the potential to support these applications, allowing for broad network connectivity, distributed computation, and real-time data sharing. Using cloud computing and IoT together provides enhanced performance and security. Still, they could also be used to develop innovative new applications. Cloud computing allocates tasks to virtual machines (VMs) with varying lengths, launch times, and processing times. A key factor in this process is balancing the loads among VMs. Cloud computing presents a number of challenges concerning load balancing. Since cloud workloads can fluctuate depending on users' requirements, it is difficult to arrange these resources in the cloud. By using task scheduling, load balancing allows the requests to be split between different machines so that more jobs are processed in less time, and the VMs' performance can be monitored. A key goal of load balancing is to reduce makespan times and increase cloud resource utilization. The rest of the paper is arranged in the following manner. Related works are reviewed in Section II. Section III gives a background of the problem. Section IV presents the proposed method. Section V concludes the paper.

II. Related Works

Hassan, El Desouky [25] has developed an intelligent and context-aware hybrid model for assisting patients at home under close supervision. Cloud-based systems monitor chronic illness patients at home and store and process massive amounts of data generated by ambient assisted living systems. The local component monitors patients in the event of a cloud system failure or internet disconnect. Real-time health status can be determined using context-aware methods that continuously monitor ambient conditions, physiological signals, and patient behavior. The effectiveness of the model for monitoring and identifying emergencies in an unbalanced dataset has been demonstrated in an experimental case study, including patients with blood pressure disorders. However, the case study only included a small sample size of patients. In order to truly be effective, the local component would need to be tested on a larger, more diverse group of patients.

Kumar, Lokesh [26] propose a mobile healthcare IoT and cloud-based application. By combining the medical sensors with the UCI repository dataset, the researchers have created diabetes datasets for predicting people with severe diabetes. Additionally, a new algorithm has been developed that diagnoses and estimates the severity of diseases using fuzzy rule-based neural classifiers. Health records from various hospitals and UCI Repository datasets were used for some experiments. Compared to existing disease prediction systems, the proposed mechanism is more effective. However, there are some potential drawbacks to this system. First, it is reliant on accurate and up-to-date data from both the medical sensors and the UCI repository. If either of these sources is inaccurate or out-of-date, the predictions made by the system will be less accurate. Additionally, the system is only as effective as the algorithms used to make the predictions. If the algorithms are not accurate, the predictions will also be inaccurate.

Using edge-cognitive computing, Chen, Li [27] have developed an intelligent healthcare system. This system monitors and assesses users' physical health using cognitive computing. Using a health risk assessment table, a user's health risk grade is determined under various health statuses and then distributed across the entire edge computing network based on the user's health risk grade. As a result, the network has a deeper cognitive intelligence that is application-aware. Experiments show the proposed method increases patient survival rates in an emergency while improving user experience and maximizing computing resource utilization. However, there are also potential risks associated with this health risk assessment table. For example, if the health risk grade is incorrectly determined, it could lead to devastating consequences for the user. In addition, if the table is not updated frequently, it may not be accurate in predicting health risks.

In addition, Garbhapu and Gopalan [28] have created a device using the IoT to monitor patients simultaneously while wirelessly transmitting data to physicians across the globe. Hub and spoke diagrams illustrate the interaction between the sensor nodes in the instrument, which include a microprocessor MSP430 and a wireless transceiver nRF24L01. The wristband has a sensor that continuously monitors the user's vital signs. Data collected by the MSP430 is stored and wirelessly transmitted for future treatment by doctors. However, there are also potential risks associated with using such technology. For example, if the data collected by the wristband is not properly encrypted, it could be hacked and used to exploit the user's personal information. Additionally, if the wristband malfunctions, it could give false readings that could lead to incorrect diagnosis and treatment by doctors.

An IoT-based mechanism for authenticating health monitoring signals was developed by Hossain and Muhammad [29]. Since ECG monitoring is an essential evaluation tool, it was used as a case study. ECG signals can be constantly monitored by a medical expert to detect illness and prescribe medications to minimize avoidable deaths. ECG signals are captured by a portable ECG recording device and sent via Bluetooth to a smartphone or computer. To ensure security and authentication, the recorded signal is processed with a simple algorithm to remove unwanted noise. After extracting and classifying the temporal and spectral features of the watermarked ECG signal, a one-class support vector machine classifier is used. A suitable healthcare expert evaluates the watermarked ECG and classification decision. The expert then enters a prescription and a conclusion into the cloud server. Eventually, the patient is notified by the cloud. Cloud-based ECG health monitoring experiments and simulations has demonstrated the applicability of this method. However, while this system may be convenient for some, there are also potential privacy concerns that come with storing this sensitive information in the cloud. There is also the potential for system failure or human error that could lead to misdiagnosis or delayed diagnosis.

An overview of ubiquitous and intelligent healthcare systems is provided by Motwani, Shukla [30]. A novel mechanism to monitor and recommend smart patients was developed using deep learning and cloud-based analytics. The proposed mechanism monitors and forecasts patients' health status using vital signs and activity context, generated by ambient assisted living devices and calls for assistance. The experimental study used an unbalanced dataset compiled from a case study of blood pressure patients and predicted the patients' actual health status. Preventive and therapeutic care can be provided in real-time without Internet access or cloud services. However, the accuracy of the predictions made by the mechanism may not be accurate enough to provide reliable preventive or therapeutic care to patients in real-time.

Using blockchain technology and attribute-based encryption Mubarakali [31] has proposed a method for securely transmitting healthcare data in cloud environments by addressing privacy and security concerns. A centralized healthcare system collects patient data using wearable devices. Patient conditions are observed while sleeping, hearing heartbeats, and walking. Servers in the cloud receive the gathered data. Doctors review the patient's observation report, genetic information, and clinical test results before prescribing the appropriate medicine and precautions to speed up recovery. Based on experimental results, the approach has shown better success rates, execution times, and delay times than previous methods. However, there are also potential privacy concerns that come along with this method of data collection and storage. If patient data is not properly secured, there is a risk that it could be accessed and used without the patient's consent or knowledge. There is also the potential for security breaches that could lead to the release of sensitive patient information.

III. Backgrounds

With the rapid growth of the IoT and the popularity of wearable devices, the relationship between healthcare and technology has significantly changed in recent years. This results in personalized healthcare, increasing access to healthcare like never and allowing for customization like never before. IoT-enabled healthcare models require an intensive collaboration with cutting-edge technologies to build a flexible, efficient and secure platform for developing new services. A major objective of IoT in healthcare is to replace the traditional approach with an electronic approach, which means linking the patient and doctor in real-time via a smartphone so that doctors can continuously track the patient's health and provide him with recommendations.

Cloud computing provides a cost-effectively scalable and resilient computing resources, enabling large-scale applications to be deployed quickly and efficiently. Large-scale applications handle a large volume of work. There are three types of cloud computing services: infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS). SaaS cooperative models allow users to access cloud applications via web browsers on desktops or workstations. Using PaaS, developers can improve, evaluate, and accommodate their applications. The IaaS model enables large-scale applications to be run on an elastic and scalable computing platform.

VMs offer users’ preconfigured CPU processors, bandwidth, memory, and storage capacities in a virtualized computing environment. Virtualization solutions are available at various prices to meet customers' varied needs, allowing them to manage computer resources efficiently. IaaS offers end users three primary benefits. Users pay for the infrastructure usage as they would for essential services. Resources can be contracted or increased depending on the needs of the application. Additionally, IaaS cloud computing improves application development by providing direct access to resources. Lastly, according to users' requirements, renting resources is possible anytime and anywhere. In IaaS clouds, finding enough resources to handle a wide range of massive-scale tasks is still an open and well-known challenge. The healthcare field can benefit from cloud computing techniques for maintaining patients' records and monitoring health outcomes more efficiently and effectively. Healthcare systems can benefit from cloud computing in the following ways:

- Cost reductions: Hardware and operating systems do not have to be monopolized by healthcare systems.
- Security and privacy: Cloud providers must meet several privacy requirements, including the Health Insurance Law and Issue.
- Exchange of health information: Healthcare organizations can exchange information through cloud computing.
- Telemedicine: As technology disperses and technologically advanced devices spread, telemedicine encompasses teleoperations, electronic health records, video conferences, and the provision of health care at home.
- Speed: Patients and healthcare providers can access confidential information quickly via cloud services.

As the number of cloud providers grows and the load on the cloud servers also increases, the scheduling problem in this environment becomes a key issue. Scheduling tasks on VMs may result in some VMs needing more utilization while others are underutilized. An efficient scheduler is needed to organize the scheduling process and balance the server loads. Load balancing has been an efficient way to distribute a system's load among the involved VMs. CPU, memory, and network loads are all possible types of loads. Cloud computing environments rely on the following metrics for load balancing.

- Resource utilization refers to the mechanism's effectiveness in employing resources for the given tasks.
- Throughput: It indicates the number of customer requests handled per unit of time.
- Makespan: It represents the time when the last submitted task was completed.
- Performance: It measures system efficiency after executing a load-balancing algorithm.
- Response time refers to the combined arrival and waiting time in a task queue.

IV. Proposed Mechanism

Ant Colony Optimization (ACO) is a meta-heuristic used to solve complex combinatorial optimization problems. The basic ACO algorithm relies on laying pheromone trails inspired by how biological ants communicate using pheromones. Artificial ants communicate with each other via artificial pheromones within a colony, similar to the biological example. Pheromone bands induce indirect communication between simple agents called artificial ants. Pheromone traces act as digital and distributed information in ACO that ants use to create solutions to problems to be solved and to adapt to them as a reflection of their search experience during the execution of the algorithm, which is why ACO differs from other construction heuristics in many ways [32, 33]. The ACO key steps are described in the following.

A. *Initialization*

All pheromone variables and parameters are set in this step.

B. *Constructing and Solutions*

The ants use the pheromone values and other information to solve the problem. Depending on the particular construction mechanism, a partial solution may not be extended and still be feasible, or a complete solution may be constructed that is impossible. If an infeasible solution violates constraint requirements to a greater or lesser extent, it may be penalized. Each step of the construction process involves selecting a solution component based on probabilities. Probability distributions are defined in various ways. The most commonly used rule is described by Eq. 1.

$$p\left(c_i^j \middle| S_p\right) = \frac{\mathcal{T}_{ij}^{a}.\left[\eta(C_i^j)\right]^{\beta}}{\sum_{C_i^l \in N(S_p)} \mathcal{T}_{il}^{a}.\left[\eta(C_i^l)\right]^{\beta}}, \forall c_i^j \in N(S_p) \tag{1}$$

C. *Local Search*

Using local search algorithms, completed candidate solutions can be further improved. ACO algorithms perform best when coupled with local search algorithms to solve combinatorial optimization problems. The local search operation can be viewed as a daemon activity that performs centralized or problem-specific functions that cannot be handled by an ant individually.

D. *Global Pheromone Update*

As pheromones are updated, subsequent iterations will be more attracted to good solutions. This goal can be achieved through two mechanisms. Firstly, pheromone deposits boost the pheromone level in selected solution components. It led to making these solution items more appealing to ants in subsequent iterations. Another mechanism is pheromone trail evaporation, which causes the pheromones left behind by previous ants to diminish over time. Practically, pheromone evaporation is necessary to prevent too rapid convergence of the algorithm towards a suboptimal region. It facilitates exploring new search spaces by implementing an effective form of forgetting. Pheromone updates are implemented by Eq. 2.

$$\mathcal{T}_{ij} = (1-\rho)_{\mathcal{T}_{ij}} + \sum_{s \in S_{upd} \middle| c_i^j \in s} g(s) \tag{2}$$

As illustrated in Fig. 1, two stages are involved in the proposed mechanism: developing an IoT-based healthcare system and solving the cloud load balancing problem using the ACO algorithm. The idea behind it in healthcare is moving from a conventional way of monitoring to a modern way called real-time remote monitoring. To build the health care system, wearable devices (medical sensors) connect to patients' bodies. These devices transfer data throughout the day through a wireless network attached to the patient's body. Servers in a cloud centre have characteristics, and virtual machines with similar characteristics surround them. A key challenge in cloud computing is load balancing, and a new algorithm is proposed to resolve it. In this design, rapid response reduces costs and improves patient care. This mechanism includes the following components:

- Doctors: They monitor patients, perform diagnoses and treatments, and prescribe medications for patients.
- VMs: Cloud resources are provisioned to customers through VMs.
- ACO algorithm: It chooses optimal VMs based on their fitness values for load distribution.
- Cloud broker: In a nutshell, it is an intermediary between the ACO algorithm and the VM manager that assigns the VMs to different loads.
- VM manager: The tool extracts tasks from the queue and provides relevant data regarding the machines regarding their activities, efficiency, number of hosts, number of virtual machines, and task counts and lengths.
- Task queue: It is utilized to keep track of the number of tasks that need to be completed by the manager.
- IoT gateway: This device bridges the sensor and the task queue. Gateways transfer data from on-premises to the cloud and vice versa.
- Patients: There are several diseases afflicting a group of patients. Data such as heart rate and temperature are sent to an IoT gateway by sensors attached to patients' bodies.

The gateway receives tasks from sensor devices and sends them to the task queue. The VM manager collects queued tasks and forwards them to the cloud broker once sufficient VMs are available. As a load balancer, the cloud broker allocates tasks to virtual machines based on their fitness values using the ACO algorithm. Each VM is given a fitness value based on the three following criteria:

E. *Execution Time*

It refers to the amount of time a VM takes to complete the execution of a given task, calculated by Eq. 3, in which U denotes the CPU utilization and V stands for the VM load. The VM load is calculated using Eq. 4, in which M refers to MIPS for the VM, L indicates the task length, and N reflects the number of tasks.

$$E = {}^{V}\!/_{U} \tag{3}$$

$$V = {}^{(N \times L)}/_{M} \qquad (4)$$

F. *Storage Utilization*

It is the proportion of occupied storage space versus total storage space, calculated by Eq. 5, in which T, A, and R stand for the total memory of VM, available memory, and the amount of memory remaining before implementing a task in the VM, respectively.

$$S = R + ({}^{A}/_{T}) \times 100\% \qquad (5)$$

G. *CPU Utilization*

It refers to the amount of work that a CPU handle using its processing resources, calculated by Eq. 6, where U indicates the CPU usage of a task.

$$C = 100 - ({}^{U}/_{N}) \qquad (6)$$

The past state of VMs and the task's duration are considered during load distribution. The search space's starting point is the matrix's centre, where each position represents a VM. In each iteration, an ant calculates the fitness value of its present position and neighbors and moves on to the next hope with the highest fitness value.

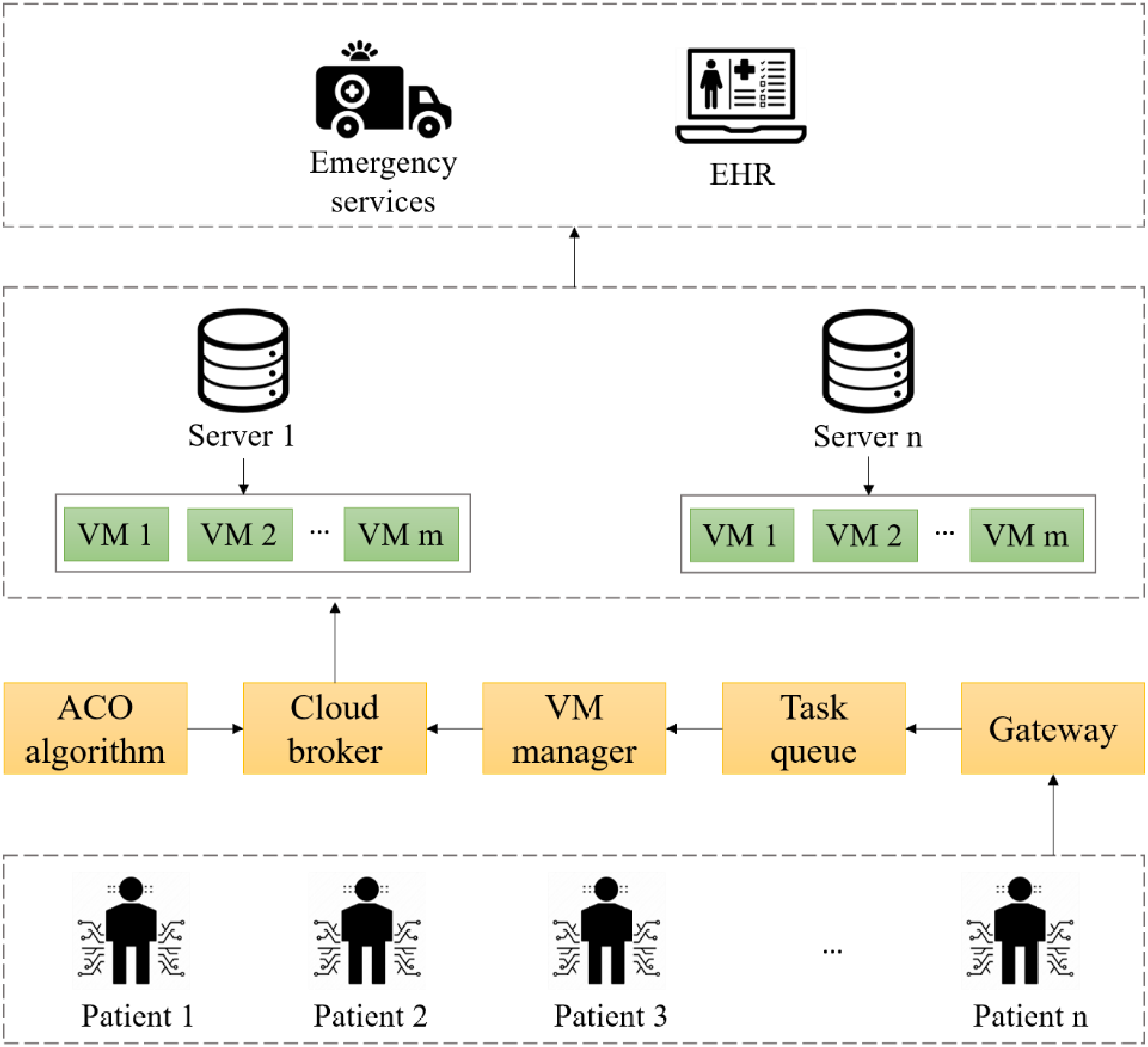


Fig. 1. Structure of proposed mechanism.

V. Experimental Results

The proposed system was evaluated experimentally and estimated using the CloudSim simulator. Parameters and assumptions are summarized in Table I. VMs differ in their ability (fitness) to execute tasks (Table II). Different VMs and tasks were used to compare the proposed mechanism with other approaches. Fitness (capability) varies from VM to VM. In this situation, load balancing is necessary to achieve the preferred performance. Thus, the ACO algorithm divides workload among virtual machines according to their capability (fitness) and task length, in which longer tasks are assigned to VMs with a high fitness level. VMs selected by the ACO are shown in Fig. 2, along with the shape of their movements, with red representing the ants' starting points and green representing the VMs selected. Table III and Fig. 3 illustrate the VMs selected and the fineness.

For each iteration, the time elapsed can be seen in Table IV. Increasing the number of iterations leads to an increase in time, as shown in Fig. 4. The ACO chooses optimal VMs for the given task based on the VM's status and length in the cloud environment. The ACO reduces makespan time as well as balances the whole system load. Table V compares the makespan time of our mechanism with those of other heuristic algorithms. When the number of tasks increases, the suggested mechanism performs faster than others; this proves that it is more effective than its competitors. As shown in Fig. 5, the proposed algorithm has a shorter makespan time. Eq. 7 calculates the processing time of an algorithm to perform a given task, in which Np represents the number of processors within the current virtual machine. A comparison of the total processing time of the proposed mechanism and recent ones are presented in Table VI. According to Fig. 6, our mechanism outperforms others regarding processing time.

$$Processing\ time = {}^{TL}\!/_{(Vm * Np)} \tag{7}$$

The implementation of a healthcare system to enhance the interactivity between patients and physicians has been enhanced with the integration of IoT with wearable devices. Specifically, the ACO algorithm has been applied to load balance in cloud computing, which assigns tasks based on their length and fitness values to the appropriate VMs. By assigning tasks to the most efficient VMs, the ACO algorithm can ensure that the tasks are completed in a timely manner, while also reducing latency and improving the overall user experience. Furthermore, the use of wearables enables the collection of data that can be used to track and monitor patient progress over time, allowing for more accurate diagnosis and treatment. A primary aim of load balancing is to eliminate the overloaded tasks from VMs and allocate them to underloaded VMs, improve the runtime of loads (tasks), decrease the overall makespan time, maximize resource utilization, and balance the load in each VM.

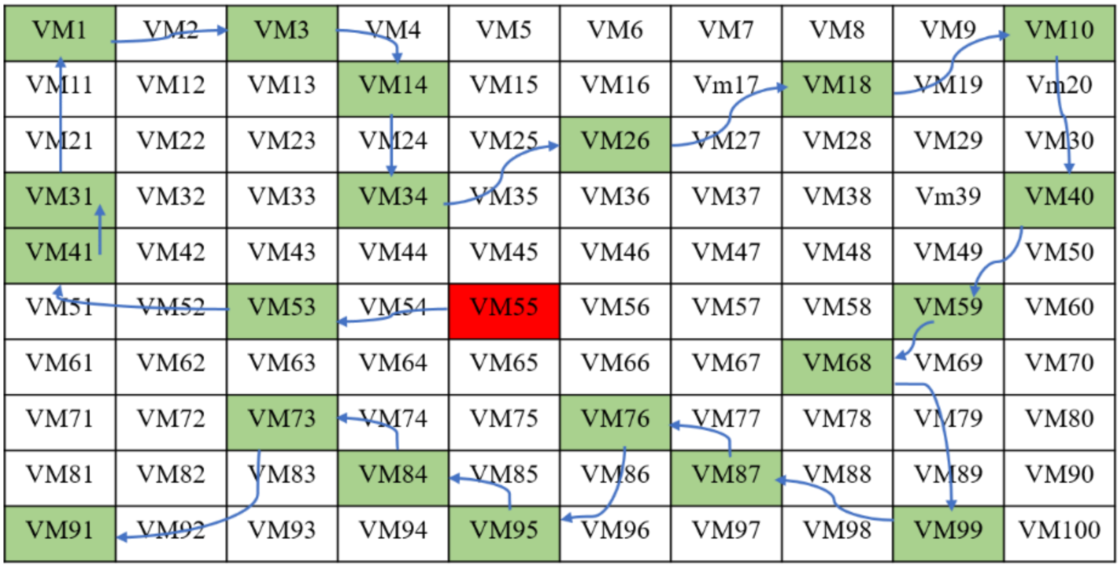
This study demonstrated that the ACO algorithm is more effective than other meta-heuristic algorithms in providing strong optimal results. In most cases, it provides better solutions, is faster in convergence, and is more computationally efficient than other approaches. When compared to prior work, this work is effective in reducing the makespan time and the processing time. In all cases, resource utilization increases when the number of tasks increases. Furthermore, the degree of imbalance is measured in order to assess the effectiveness of the proposed method. The measurements indicate that the proposed method performs better for a large number of cloudlets, estimated at 177.767.

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Number of hosts	10
VMs memory	512-2048
VMs MIPS	500-2000
Number of VMs	100
Length of tasks	1000-5000 MI
Number of tasks	100

TABLE II. VMS AND FITNESS VALUES

VM number	Fitness	VM number	Fitness	VM number	Fitness	VM number	Fitness
1	95.1	26	91.2	51	76	76	92.6
2	71.4	27	81.5	52	81.1	77	88.1
3	96.6	28	82.4	53	91.4	78	80.1
4	83.7	29	77.3	54	81.2	79	65.3
5	73.2	30	69.6	55	89.8	80	77.1
6	67.6	31	93.5	56	84.3	81	69.2
7	78.5	32	85.4	57	79.1	82	79.4
8	83.4	33	87.7	58	79.4	83	70.6
9	77.3	34	93.4	59	92.5	84	91.3
10	92.1	35	81.0	60	77.5	85	71.2
11	87.5	36	78.4	61	85.4	86	86.3
12	64.9	37	86.4	62	81.2	87	96.2
13	68.6	38	69.8	63	85.0	88	81.0
14	94.3	39	73.6	64	68.9	89	69.4
15	67.4	40	94.0	65	71.5	90	73.3
16	75.8	41	84.7	66	77.6	91	92.9
17	68.3	42	81.5	67	82.3	92	90.1
18	95.9	43	83.4	68	91.0	93	86.3
19	89.7	44	75.6	69	78.3	94	87.6
20	74.2	45	87.4	70	68.4	95	95.2
21	87.5	46	83.3	71	66.5	96	74.3
22	81.2	47	85.8	72	79.3	97	79.1
23	74.3	48	86.5	73	94.4	98	82.5
24	78.5	49	73.4	74	88.1	99	93.7
25	82.2	50	88.5	75	81.5	100	88.3



VM1	VM2	VM3	VM4	VM5	VM6	VM7	VM8	VM9	VM10
VM11	VM12	VM13	VM14	VM15	VM16	Vm17	VM18	VM19	Vm20
VM21	VM22	VM23	VM24	VM25	VM26	VM27	VM28	VM29	VM30
VM31	VM32	VM33	VM34	VM35	VM36	VM37	VM38	Vm39	VM40
VM41	VM42	VM43	VM44	VM45	VM46	VM47	VM48	VM49	VM50
VM51	VM52	VM53	VM54	VM55	VM56	VM57	VM58	VM59	VM60
VM61	VM62	VM63	VM64	VM65	VM66	VM67	VM68	VM69	VM70
VM71	VM72	VM73	VM74	VM75	VM76	VM77	VM78	VM79	VM80
VM81	VM82	VM83	VM84	VM85	VM86	VM87	VM88	VM89	VM90
VM91	VM92	VM93	VM94	VM95	VM96	VM97	VM98	VM99	VM100

Fig. 2. VMs selected by the ACO algorithm.

TABLE III. VMS AND FINENESS VALUES

Iteration	VM number	Fitness	Iteration	VM number	Fitness
1	53	91.4	11	40	94.0
2	41	84.7	12	59	92.5
3	31	93.5	13	68	91.0
4	1	95.1	14	99	93.7
5	3	96.6	15	87	96.2
6	14	94.3	16	76	92.6
7	34	93.4	17	95	95.2
8	26	91.2	18	84	91.3
9	18	95.9	19	73	94.4
10	10	92.1	20	91	92.9

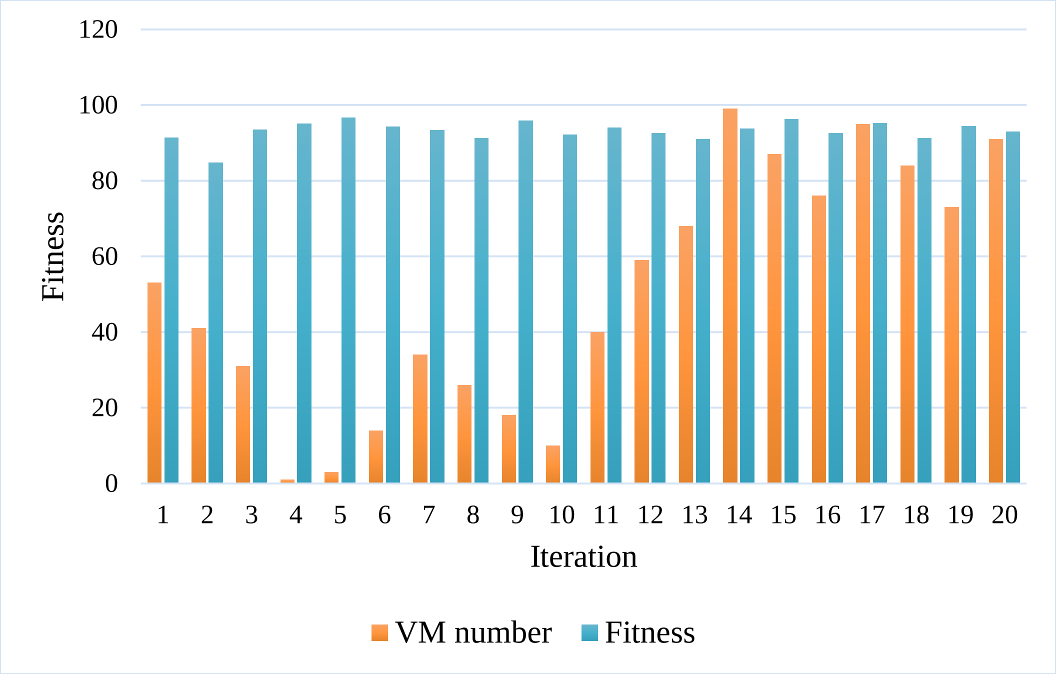


Fig. 3. VMs and fitness values.

TABLE IV. CONSUMED TIME IN EACH ITERATION

Iteration	Time (s)	Iteration	Time (s)
1	0.22	11	1.42
2	0.31	12	1.88
3	0.46	13	2.2
4	0.54	14	2.48
5	0.63	15	3.1
6	0.76	16	3.31
7	0.83	17	3.54
8	0.95	18	3.87
9	1.2	19	4.22
10	1.31	20	4.8

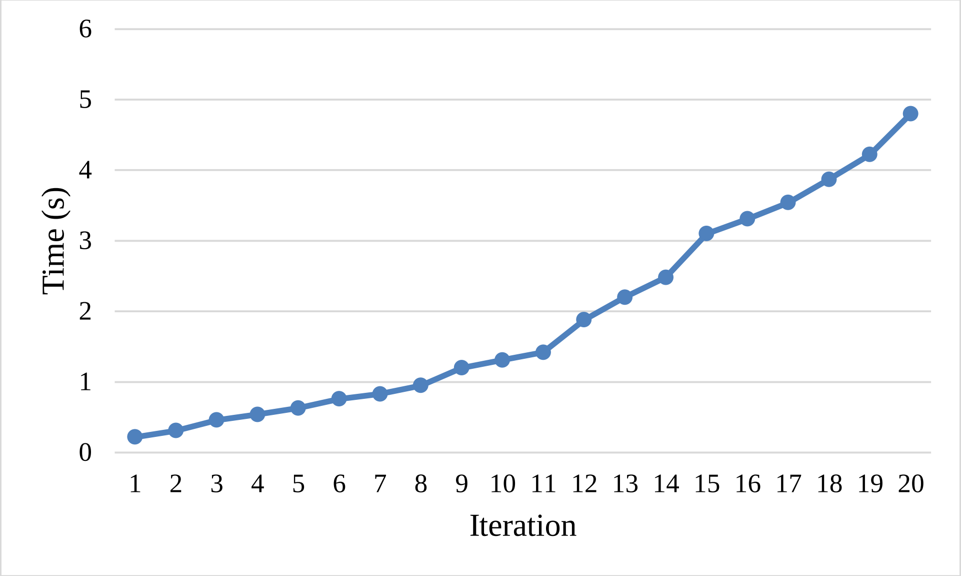


Fig. 4. Consumed time in each iteration.

TABLE V. MAKESPAN COMPARISON

Number of tasks	Hyper_load [34]	GA-load [34]	PSO_load [34]	LB_PSO [35]	Proposed method
100	4.0	6.81	15.8	3.51	2.81
200	4.87	12.75	46.1	4.71	4.07
300	13.65	22.59	95.86	10.6	9.7
400	13.68	35.5	170.6	12.22	12.33
500	24.5	47.7	271.2	23.25	22.53

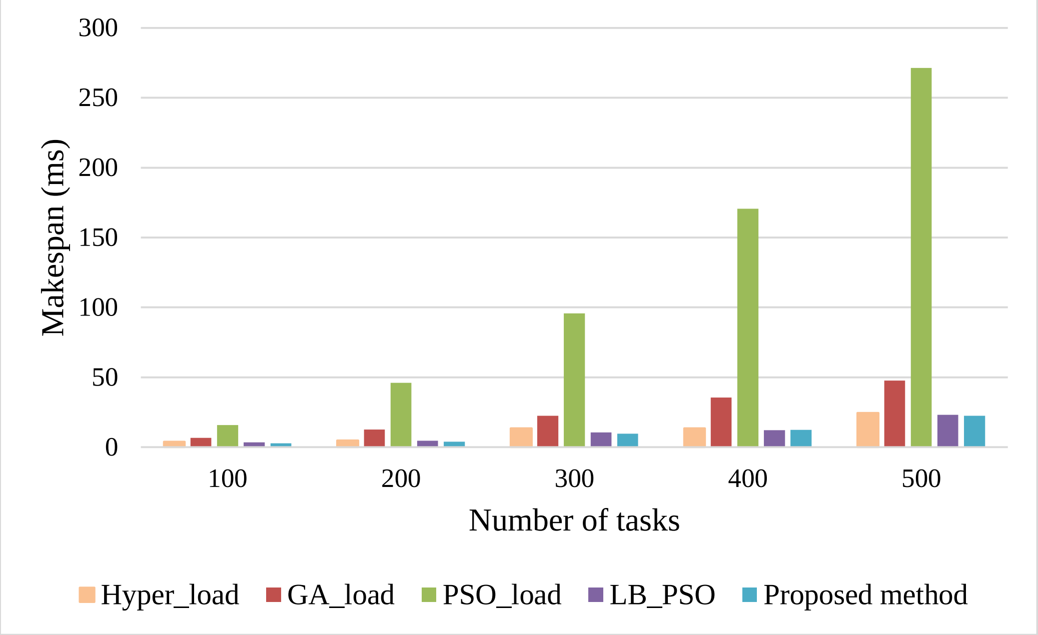


Fig. 5. Makespan comparison.

TABLE VI. PROCESSING TIME COMPARISON

Number of tasks	Hyper_load [34]	GA-load [34]	Proposed method
100	99.3	96.4	94.5
200	239.6	234.7	231.2
300	418.2	412.9	411.6
400	571.8	630.5	534.8
500	904.3	984.1	881.2

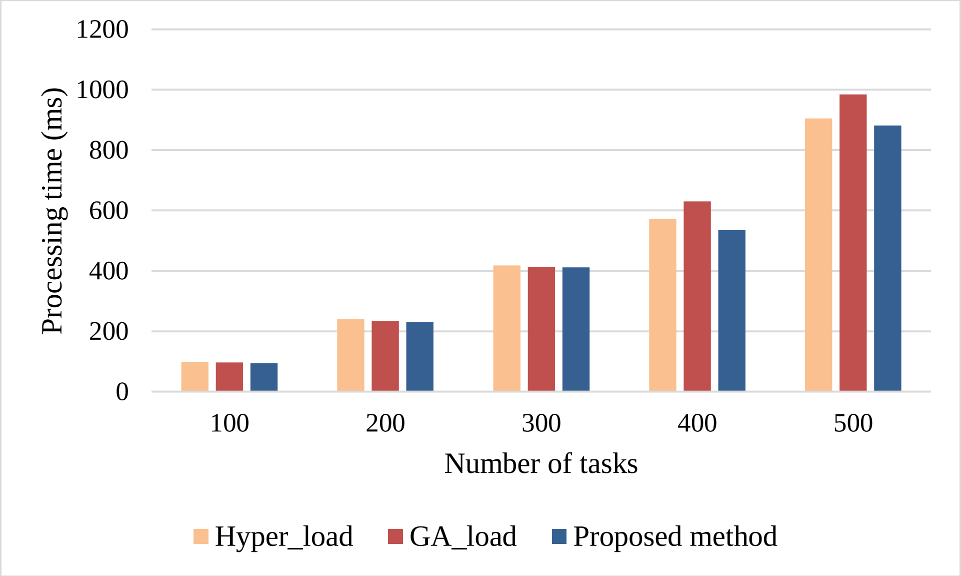


Fig. 6. Processing time comparison.

VI. CONCLUSION

A healthcare system has been implemented by connecting IoT with wearable devices and cloud computing to enhance patient-doctor interaction. Cloud computing load balancing is solved with the ACO algorithm, which assigns tasks to suitable virtual machines based on fitness values and lengths of tasks. Load balancing focuses on eliminating overloaded tasks from the VMs, reallocating them to underloaded VMs, increasing load time (tasks), reducing makespan times, and increasing resource utilization. In comparison to other meta-heuristic algorithms, the ACO algorithm provides strong optimal results, provides better solutions, is faster in convergence than other methods, and is more computationally efficient than previous ones. The proposed method may be improved or enhanced in the future by improving the ACO algorithm in order to reduce the search time for candidate VMs and solve the local optimization problem. Moreover, the performance and efficiency of the system can be enhanced by integrating or utilizing other intelligent algorithms. Messages can be given priority based on their importance. Additionally, the proposed method could be extended to incorporate additional constraints such as cost, availability, or energy efficiency. Furthermore, the proposed method could be adapted to work with different types of clouds, such as public, private, and hybrid clouds, to offer even greater scalability and flexibility.

REFERENCES

[1] Pourghebleh, B., et al., A roadmap towards energy-efficient data fusion methods in the Internet of Things. Concurrency and Computation: Practice and Experience, 2022: p. e6959.

[2] Kumar, A., et al., Smart power consumption management and alert system using IoT on big data. Sustainable Energy Technologies and Assessments, 2022: p. 102555.

[3] Meisami, S., M. Beheshti-Atashgah, and M.R. Aref, Using Blockchain to Achieve Decentralized Privacy In IoT Healthcare. arXiv preprint arXiv:2109.14812, 2021.

[4] He, P., et al., Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review. Sustainable Computing: Informatics and Systems, 2022. 36: p. 100822.

[5] Ataie, I., et al. D 2 FO: Distributed Dynamic Offloading Mechanism for Time-Sensitive Tasks in Fog-Cloud IoT-based Systems. in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC). 2022. IEEE.

[6] Taami, T., S. Krug, and M. O'Nils. Experimental characterization of latency in distributed iot systems with cloud fog offloading. in 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS). 2019. IEEE.

[7] Haghshenas, S.H., M.A. Hasnat, and M. Naeini, A Temporal Graph Neural Network for Cyber Attack Detection and Localization in Smart Grids. arXiv preprint arXiv:2212.03390, 2022.

[8] Khosravi, F., et al. Implementation of an Elastic Reconfigurable Optical Add/Drop Multiplexer based on Subcarriers for Application in Optical Multichannel Networks. in 2022 International Conference on Electronics, Information, and Communication (ICEIC). 2022. IEEE.

[9] Khosravi, F., et al., Improving the performance of three level code division multiplexing using the optimization of signal level spacing. Optik, 2014. 125(18): p. 5037-5040.

[10] Akhavan, J. and S. Manoochehri. Sensory data fusion using machine learning methods for in-situ defect registration in additive manufacturing: a review. in 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). 2022. IEEE.

[11] Vahedifard, F., et al., Artificial intelligence for radiomics; diagnostic biomarkers for neuro-oncology. World Journal of Advanced Research and Reviews, 2022. 14(3): p. 304-310.

[12] Saeidi, S.A., et al. A novel neuromorphic processors realization of spiking deep reinforcement learning for portfolio management. in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022. IEEE.

[13] Mehbodniya, A., et al. Gibbs Sampling Aided Throughput Improvement for Next-Generation Wi-Fi. in 2018 IEEE Globecom Workshops (GC Wkshps). 2018. IEEE.

[14] Pourghebleh, B., V. Hayyolalam, and A.A. Anvigh, Service discovery in the Internet of Things: review of current trends and research challenges. Wireless Networks, 2020. 26(7): p. 5371-5391.

[15] Pourghebleh, B., K. Wakil, and N.J. Navimipour, A comprehensive study on the trust management techniques in the Internet of Things. IEEE Internet of Things Journal, 2019. 6(6): p. 9326-9337.

[16] Pourghebleh, B. and V. Hayyolalam, A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things. Cluster Computing, 2019: p. 1-21.

[17] Pourghebleh, B., et al., The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments. Cluster Computing, 2021: p. 1-24.

[18] Mohseni, M., F. Amirghafouri, and B. Pourghebleh, CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic. Peer-to-Peer Networking and Applications, 2022: p. 1-21.

[19] Aswini, J., et al., An efficient cloud-based healthcare services paradigm for chronic kidney disease prediction application using boosted support vector machine. Concurrency and Computation: Practice and Experience, 2022. 34(10): p. e6722.

[20] Javaid, M., et al., Evolutionary Trends in Progressive Cloud Computing based Healthcare: Ideas, Enablers, and Barriers. International Journal of Cognitive Computing in Engineering, 2022.

[21] Pourghebleh, B. and N.J. Navimipour, Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research. Journal of Network and Computer Applications, 2017. 97: p. 23-34.

[22] Kamalov, F., et al., Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective. Sustainability, 2023. 15(4): p. 3317.

[23] Farid, F., et al., A smart biometric identity management framework for personalised IoT and cloud computing-based healthcare services. Sensors, 2021. 21(2): p. 552.

[24] Hajipour Khire Masjidi, B., et al., CT-ML: diagnosis of breast cancer based on ultrasound images and time-dependent feature extraction methods using contourlet transformation and machine learning. Computational Intelligence and Neuroscience, 2022. 2022.

[25] Hassan, M.K., et al., Intelligent hybrid remote patient-monitoring model with cloud-based framework for knowledge discovery. Computers & Electrical Engineering, 2018. 70: p. 1034-1048.

[26] Kumar, P.M., et al., Cloud and IoT based disease prediction and diagnosis system for healthcare using Fuzzy neural classifier. Future Generation Computer Systems, 2018. 86: p. 527-534.

[27] Chen, M., et al., Edge cognitive computing based smart healthcare system. Future Generation Computer Systems, 2018. 86: p. 403-411.

[28] Garbhapu, V.V. and S. Gopalan, IoT based low cost single sensor node remote health monitoring system. Procedia computer science, 2017. 113: p. 408-415.

[29] Hossain, M.S. and G. Muhammad, Cloud-assisted industrial internet of things (iiot)–enabled framework for health monitoring. Computer Networks, 2016. 101: p. 192-202.

[30] Motwani, A., P.K. Shukla, and M. Pawar, Novel framework based on deep learning and cloud analytics for smart patient monitoring and recommendation (SPMR). Journal of Ambient Intelligence and Humanized Computing, 2021: p. 1-16.

[31] Mubarakali, A., Healthcare services monitoring in cloud using secure and robust healthcare-based BLOCKCHAIN (SRHB) approach. Mobile Networks and Applications, 2020. 25(4): p. 1330-1337.

[32] Wang, Q., et al., Continuous space ant colony algorithm for automatic selection of orthophoto mosaic seamline network. ISPRS Journal of Photogrammetry and Remote Sensing, 2022. 186: p. 201-217.

[33] Xiao, H., Z. Hu, and K. Li, Multi-objective VM consolidation based on thresholds and ant colony system in cloud computing. IEEE Access, 2019. 7: p. 53441-53453.

[34] Gupta, A., H.S. Bhadauria, and A. Singh, Load balancing based hyper heuristic algorithm for cloud task scheduling. Journal of Ambient Intelligence and Humanized Computing, 2021. 12(6): p. 5845-5852.

[35] Pradhan, A. and S.K. Bisoy, A novel load balancing technique for cloud computing platform based on PSO. Journal of King Saud University-Computer and Information Sciences, 2020.

Multi-String Missing Characters Restoration for Automatic License Plate Recognition System

Ishtiaq Rasool KHAN[1], Syed Talha Abid ALI[2], Asif SIDDIQ[3], Seong-O SHIM[4]
College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia[1, 4]
Dept. Electrical Engineering, Pakistan Institute of Engineering and Technology, Multan, Pakistan[2, 3]

***Abstract*—Developing a license plate recognition system that can cope with unconstrained real-time scenarios is very challenging. Additional cues, such as the color and dimensions of the plate, and font of the text, can be useful in improving the system's accuracy. This paper presents a deep learning-based plate recognition system that can take advantage of the bilingual text in the license plates, as used in many countries, including Saudi Arabia. We train and test the model using a custom dataset generated from real-time traffic videos in Saudi Arabia. Using the English alphanumeric alone, the accuracy of our system was on par with the existing state-of-the-art algorithms. However, it increased significantly when the additional information from the detection of Arabic text was utilized. We propose a new algorithm to restore noise-affected missing or misidentified characters in the plate. We generated a new test dataset of license plates to test how the proposed system performs in challenging scenarios. The results show a clear advantage of the proposed system over several commercially available solutions, including Open ALPR, Plate Recognizer, and Sighthound.**

Keywords—Automatic License Plate Recognition (ALPR); Intelligent Transportation System (ITS); Optical Character Recognition (OCR); Deep Convolutional Neural Network; You Only Look Once (YOLO)

I. INTRODUCTION

In recent times, smart cities have been a popular trend, resulting in the pacing of the development of several enabling technologies. One of them is robust automatic license plate recognition (ALPR) [1], which offers a computer vision based solution for intelligent transportation systems (ITS) [2]. In this regard, a passive mesh of cameras is generally installed at road intersections and other suitable locations to observe vehicle routing through urban environments [3]. ITS can improve safety and mobility and help law enforcement agencies monitor traffic effectively. There have been several systems proposed to detect and recognize license plates (LPs) for various applications like toll fees collection [4], monitoring for the speed of a car on the road [5], traffic volume estimation on [6], detection of illegal parking [7], highway surveillance [8], and border control [9]. However, most systems tackling any of these aspects often work well only when operated in a restricted environment where camera distance and angle are fixed [10]. In an unconstrained environment, factors like low image resolution, dynamic background, motion blur [11], and variable lighting conditions, can degrade the image quality. This makes recognizing LP characters in unconstrained environments a very challenging task.

Some solutions that have been proposed to address these variabilities include Laplacian gradient-based partial character segmentation [12], Generative Adversarial Network (GAN) [13], super-resolution (SR) image reconstruction using maximum a posteriori (MAP) [14], SR reconstruction [15], and Recurrent Neural Network (RNN) [16]. Although these approaches can help restore the deteriorated or missed characters within LP, they do it at the expense of higher computational complexity due to the additional machine learning model used for the restoration task. Similarly, methods like multiscale adaptive thresholding [17] and matrix edge information based adaptive thresholding [18] depend on edge information. They can produce false positives due to weak edges, especially in the presence of high noise.

We propose a simple and fast technique for detecting and recognizing the contents of LP in unconstrained environments. You Only Look Once Version 5 (YOLOv5) network is used for detection, while a custom-built convolutional neural network (CNN) recognizes the contents in the detected LP. We collected a custom dataset from real-time traffic in Saudi Arabia, where the number plates are bilingual, using Arabic text on top and English at the bottom. The unique feature of the proposed method compared to the existing works is that it tries to recognize alphanumerics in both languages and then combines both results to improve the overall accuracy of LP recognition. We propose a new algorithm, which can address several scenarios of unrecognized and wrongly recognized alphanumerics. It uses computationally simple and efficient techniques, implemented without using a separate deep learning model, and does not depend on the deteriorated character's edge information either to restore the noise-affected missed character. Our results show that the proposed model outperforms commercially available server-based ALPR systems of Sighthound [19], Plate Recognizer [20], and Open ALPR [21].

The rest of the paper is organized as follows. Section II gives an overview of the related work. Section III describes training dataset preparation steps, including extracting LP from the captured video streams and synthetically generating and augmenting more plates to add to the data. The steps of detecting LP, recognizing Arabic and English alphanumeric characters, and restoration of missed and wrongly recognized characters are discussed in Section IV. Experimental evaluation is reported in Section V, while some conclusions and future research directions are given in Section VI.

II. Related Work

Over the years, numerous designs based on different architecture and technologies have been proposed to solve the detection and recognition problems of ALPR. For the detection of LPs, some works use image processing-based solutions. Chen and Luo [22] proposed an LP localization method using an improved version of the Prewitt arithmetic operator. It extracts the exact location from vertical and horizontal projections. However, the method requires prior knowledge of all the characters' textures which become difficult to obtain while working in an unconstrained real-time environment. Vijeta et al. [12] used stroke pair candidate detection by using Laplacian and gradient information. They find pixels that represent the stroke width of characters on a given LP. However, this method usually fails while dealing with complex backgrounds in real time due to unsymmetrical features within the captured frames.

YOLO models have been used to detect LPs in more recent works. Alghyaline [11] used a modified YOLOv3 model. Since the LP size is much smaller than the captured frame, they used only 15 layers instead of the original 75. It made the process fast but at the expense of low accuracy when detecting smaller characters within LPs. Ju-Yeong Sung [23] used YOLOv4, which can detect smaller objects, both LP and the characters within it, but at the expense of higher training costs. Khan et al. [24] used the YOLOv5 model to detect LP and its characters in a real-time environment. The YOLOv5 model uses path aggregators to speed up the detection process in real time. It also uses the mosaic data augmentation technique, which helps detect very small objects in challenging environments. Moreover, it also uses auto-learning bounding box anchors, which help make bounding boxes around objects despite the challenging backgrounds within a frame.

Some works use image processing-based techniques to reconstruct deteriorated or missed characters for the recognition task. Vijeta [12] proposed partial character reconstruction with the Tesseract optical character recognition (OCR) library, which uses the characteristics of stroke widths in the Laplacian and gradient domains. The method first enhances high-contrast information at the edges using symmetrical features of incomplete characters by suppressing background information. Then it uses stroke width properties to reconstruct the complete shape of the deteriorated characters. This method, however, proves to be not very effective for restoring completely deteriorated characters. Khoshki et al. [17] proposed a multiscale adaptive thresholding method for ALPR, which is used to find the candidates matching the LP characters affected by noise. However, the effectivity of this method lessens under varying illumination conditions. Moreover, it is computationally intensive and requires significant time to process input images. Mokji et al. [18] proposed an algorithm that incorporates matrix edge information, which enhances foreground objects in relation to the neighborhood pixels. This method does not work well when the edges of objects are weak and disconnected. Moreover, in an extremely degraded image, strong interfering patterns can generate false edges.

Some works used SR algorithms along with machine learning models. Lin et al. [13] proposed an SR image reconstruction method using GAN, an unsupervised learning model. It preprocesses the affected input image and extracts image features using a residual dense network. The method then uses sampling to restore more information using a larger receptive field. A Markovian discriminator is used to accurately guide the generator to reconstruct high-quality restored images. Despite good results, the method increases the difficulty level of training such a highly complex model. Moreover, it shows edge artifacts as well. Zhan et al. [14] proposed a MAP-based SR image reconstruction approach. It helps to estimate distribution and model parameters that best explain an observed dataset. It then uses a Huber Markov random field (HMRF) along with an ALPR system to measure image smoothness. The technique improves the recognition rate by restoring and improving the quality of LP image. However, quality enhancement is limited to single-frame reconstruction instead of a sequence of video frames.

Chen et al. [15] proposed an SR reconstruction algorithm. It uses the mechanism of attention along with a feature map to reconstruct multiscale SR images from original low-resolution images. It uses a combined feature map of multiple channels before applying a reconstruction module. It also uses interdependence to adaptively adjust the characteristics of the channel to restore details before generating high-resolution images at different scales. Despite reconstructing images, it increases weight parameters which eventually increases training time. Hui Li et al. [16] extracted and recognized the sequential features from the whole LP using a recurrent neural network with long short-term memory. However, a large number of labeled LPs are required for training. Duan et al. [25] used inception structure, which utilizes computational resources effectively by image dimensionality reduction, in an end-to-end CNN. However, this can affect the recognition accuracy of noisy images taken in real-time scenarios. Ergun et al. [26] used a statistical method in which refined characters are stretched/reduced to a given size and matched in a labeled database. This method has a limitation as it requires exact correspondence. Moreover, a slight deterioration of characters can affect the recognition accuracy.

There are several commercial ALPR solutions also available. OpenALPR [21], PlateRecognizer [20], and Sighthound[19] are open-source systems that can recognize LPs in a given input frame. These systems can be accessed using their cloud API services that predominantly use OpenCV and Tesseract OCR libraries. Moreover, reconstruction algorithms are also part of these available systems, but they require incomplete strokes of digits or characters in a noiseless background for good performance. Otherwise, these APIs fail to restore the completely missing alphanumerics in noise effected license plates.

III. The Proposed Method

The proposed system uses several steps, including extraction of frames from the videos of traffic, annotating the frames, augmenting the dataset, training the network, and testing its performance. The complete pipeline is shown in Fig. 1, and its different modules are explained in this section.

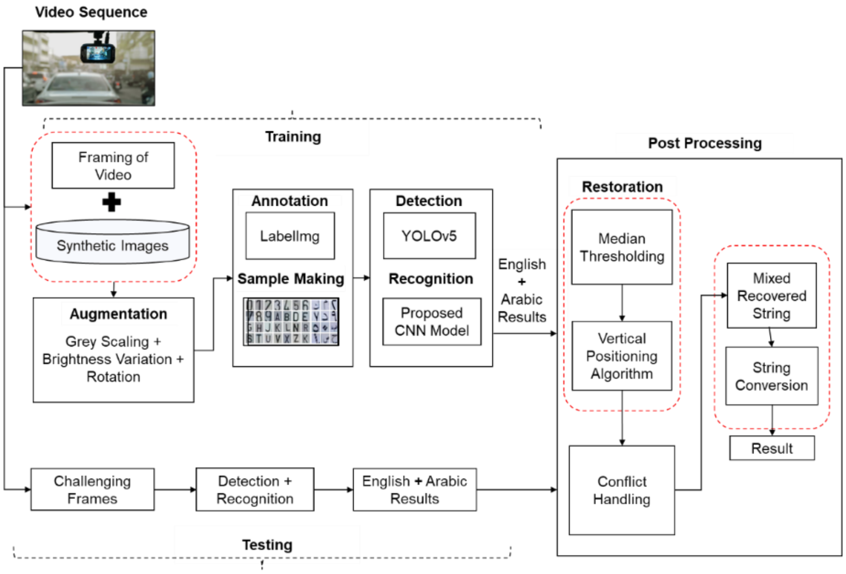


Fig. 1. The pipeline of proposed multi-string ALPR system.

A. Preparation of Dataset

1) Data acquisition: Designs of the most commonly used LPs in Saudi Arabia are shown in Fig. 2. The standard size of these plates is 32 cm in width and 16 cm in height. The color schemes indicate the usage of vehicles. We captured many videos of live traffic, with a sufficient number of samples of each type. We examined each frame of these videos manually and extracted 2600 frames in total. The rest of the frames that had no vehicle or were very similar to one of the previously selected frames were discarded. The frames were stored as color images in PNG format at FHD resolution (1920×1080 pixels).

2) Synthetic Saudi Arabian LP Generation: No Saudi LP dataset is available in the literature with sufficient size (number of images) and variety of scenarios to support robust training of LP detection and recognition models. Moreover, dataset preparation is a very time-consuming process. Hence, we increased the size of our dataset by adding some synthetically generated license plates. For this, our algorithm first generates random but mutually mapping strings of officially used English and Arabic alphanumerics. These random texts are then appended on synthetically generated layouts. The synthetic layouts contain immutable elements such as the country name, the grid, and other official symbols on the right side of every plate. The mapping coordinates are determined based on the positional analysis of different characters in the actual plates. A total of 200 synthetic images were generated this way and included in the dataset. Some examples are shown in Fig. 3. In a real-world scenario, the license plates may be dirty and less readable than the synthetically generated plates. Thus, to create more realistic data, noise is added. Afterward, all these plates are appended on some selective frames of actual data, as shown in Fig. 4.

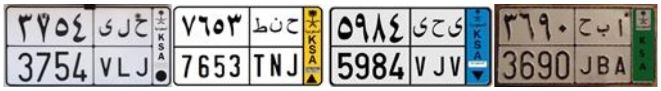


Fig. 2. Types of Saudi Arabian license plates [27].

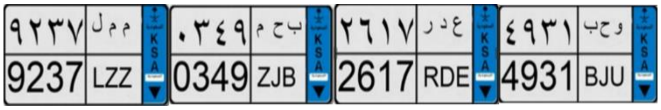


Fig. 3. Examples of synthetically generated Saudi Arabian license plates.

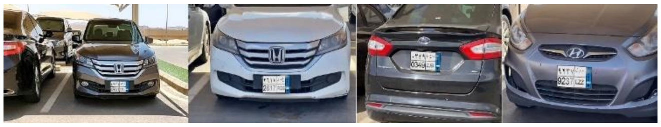


Fig. 4. Examples of appending synthetically generated plates in different video frames.

B. Training Samples Generation

All Saudi Arabian LPs have three alphabet and four digits in English and Arabic both. For training of the custom CNN model, samples for each English and Arabic Alphanumeric glyph are required. For this, an algorithm was designed to automate certain steps of this process, avoiding human intervention. YOLOv5 [28] model initially trained on the available custom dataset is used to detect LP in each frame along with all Arabic and English alphanumerics in it. Coordinates of the detected bounding boxes are used to extract the alphanumerics from the LP automatically. As the training of the network is improved, more samples can be extracted more accurately with minimal manual supervision. The process is depicted in Fig. 5.

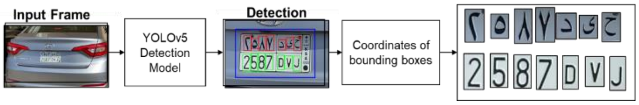


Fig. 5. Generation of training samples.

Each extracted sample is saved in its respective folder. There are 54 classes in total, including ten numerical digits (0-9) and 17 alphabets among (A-Z) that are officially used in LPs, and another 27 Arabic counterparts, containing ten digits (٠ – ٩) and 17 alphabets among (ا – ى).

C. Data Augmentation

Deep learning models require large datasets for increased accuracy. In comparison, our proprietary data is relatively limited, having 2800 frames in total, including 2600 frames extracted from the actual traffic videos and 200 generated synthetically. We split them into training and testing sets containing 2600 and 200 frames, respectively. The testing set includes challenging situations of deteriorated and completely missed alphanumeric characters in Arabic and English strings.

To increase the training dataset and its variability, data augmentation techniques were used, including gray scaling, brightness variation, and rotation. As a result, we got 6500 frames and 15000 alphanumeric characters in total. They vary considerably in size and cannot be used for training without preprocessing. We scale each of them 128x128 pixels. Through augmentation, we could increase the variability in the dataset, which helps in better training of networks, improving their accuracies, and avoiding overfitting. The numbers of frames and samples are shown in Table I.

TABLE I. SUMMARY OF THE TRAINING DATASET

Dataset	Real	Real + Augmented
Frames	2600	6500
Arabic + English Characters	9000	15000

D. Data Annotation

We used a graphical image annotation tool, LabelImg [29], to label the dataset. The coordinates of plates as well as the individual characters, are marked, as shown in Fig. 6. Since we are using YOLOv5 for detection, which accepts annotations in extensible markup language (XML) files, we save the annotations in this format.

E. Detection of License Plates

Due to their high efficiency, the detectors in the YOLO family are often used to work in real-time scenarios [30] [31] [32]. Since the release of its first version, many updates and new versions of YOLO have been developed. The most recent is YOLOv5 [28] [33], which outperforms its predecessors in terms of computational complexity and accuracy of detection. The structure of the YOLOv5 model, a relatively new family member, is shown in Fig. 7.

YOLOv5 is pre-trained on a large dataset; therefore, we use transfer learning [34] using our proprietary dataset. Transfer learning is a well-known technique to fine-tune a previously trained neural network to perform a new task. In our case, YOLOv5 is trained to perform general object detection, and we tune it for LPs. This process prevents the need to complete network training from scratch, which would have required an extensive dataset. For training YOLOv5, our training dataset is split at a ratio of 70 to 30 as shown in Table II.

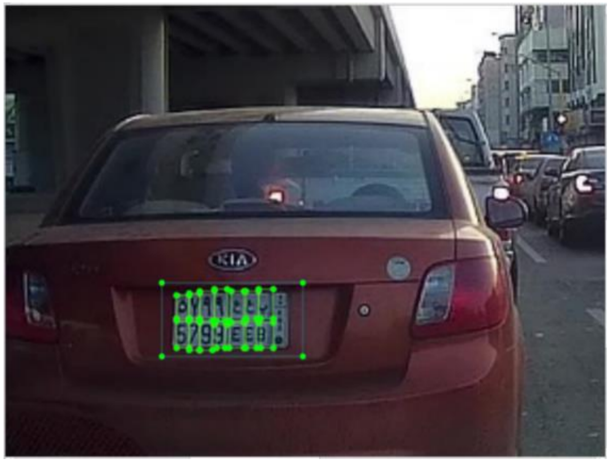


Fig. 6. Annotation of the LP and Arabic and English alphanumeric in an LP using LabelImg software tool.

TABLE II. DATASET SPLIT FOR DETECTION MODEL

Dataset	Split Percentage	Number of Frames
Training	70%	4550
Validation	30%	1950

For the training of YOLOv5, we use Google Collab notebook [28], which provides free access to powerful GPUs. The trained model takes an input frame at the dimensions of 416x416x3 and detects LP in it. The detected LP is cropped automatically, resized to the same input dimensions of 416x416x3, and fed again to the YOLOv5 model to detect its alphanumerics. Since the original frame had an FHD resolution of 1920 x 1080 pixels, the coordinates of the bounding boxes are upscaled to find the exact locations in the original frame. Some results of the extracted LPs are shown in Fig. 8 by marking the bounding boxes with red colored rectangles.

F. Custom Network for Recognition of Arabic and English Alphanumerics

The architecture of the proposed CNN for the recognition phase is shown in Fig. 9. The input of this network is the bounding box detected by the YOLOv5 in the previous step. The proposed CNN's first layer is a convolution layer with dimensions of 224x224x3; the output of YOLOv5 is resized to match it. A maxpool layer after each convolution layer uses 2x2 pixels for pooling. The convolution layers have different filter sizes (64, 128, 256, 512), and the convolution kernel size is 3x3. There are two fully connected (FC) layers, in the end, to keep the model end-to-end trainable. "Same Padding" is used to handle the convolution near the boundaries of the image, and the stride size is 1. Adam optimizer alpha is used for its desirable characteristics in non-convex optimization problems [35].

Alternating convolutional and non-linear activation layers extract rich features of given alphanumerics. The activation function of Rectified Linear Unit (ReLU) with a stride size of 1 pixel for convolutional layers and 2 pixels for the maxpool layers is used. Each of the output FC layers contains 1000 neurons. The final decision is made by using SoftMax about the recognition (classification) of alphanumeric characters.

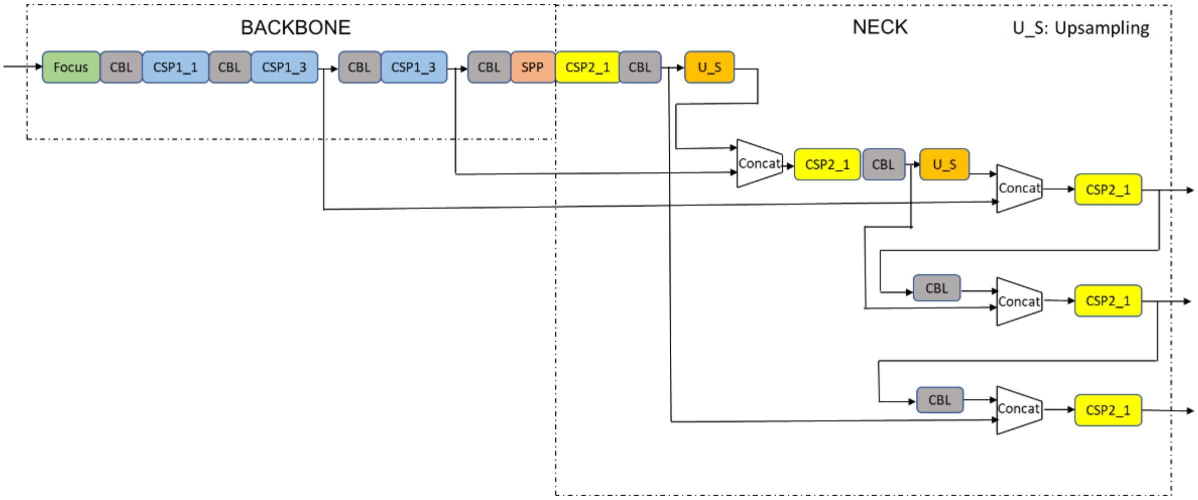


Fig. 7. CSP structure for YOLOv5 model.

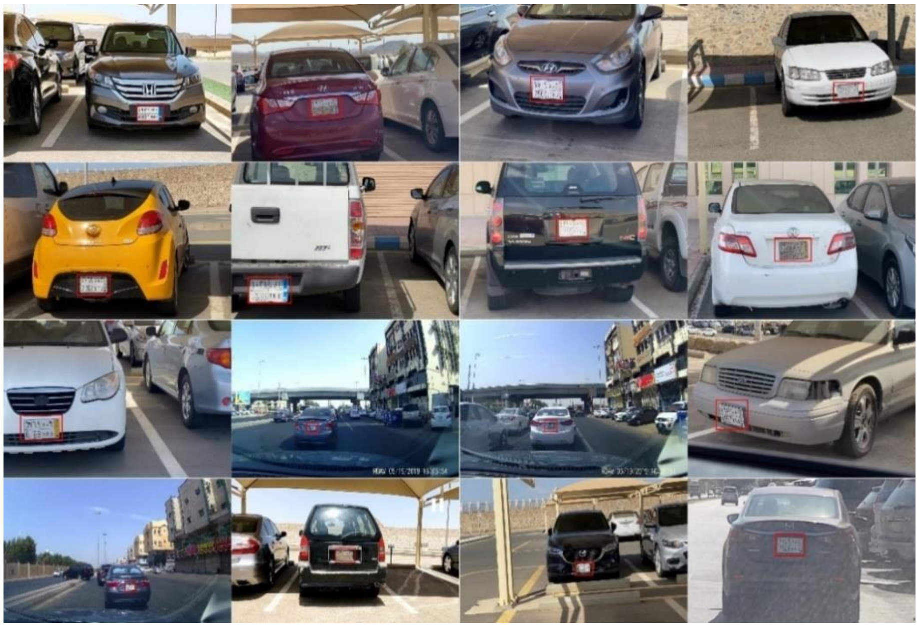


Fig. 8. YOLOv5 detection of license plates.

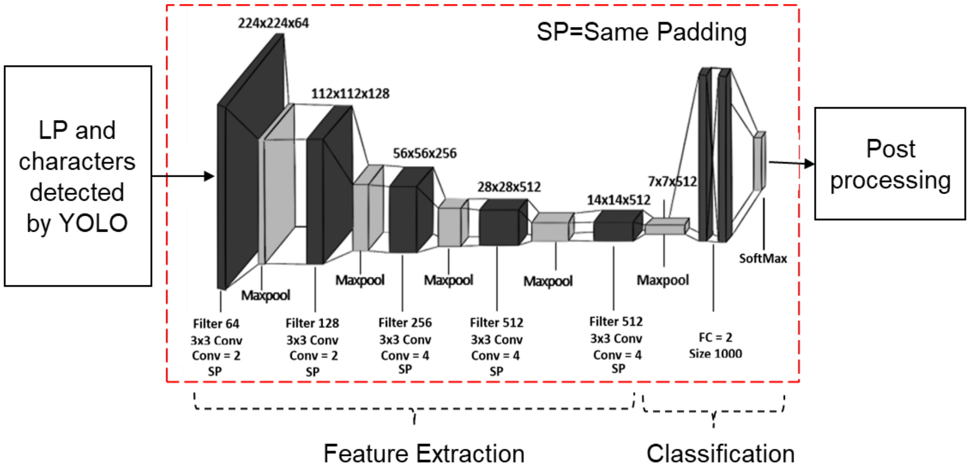

LP and characters detected by YOLO
224x224x64
SP=Same Padding
112x112x128
56x56x256
28x28x512
14x14x512
7x7x512
Maxpool
Maxpool
Maxpool
Maxpool
Maxpool
SoftMax
Filter 64 3x3 Conv Conv = 2 SP
Filter 128 3x3 Conv Conv = 2 SP
Filter 256 3x3 Conv Conv = 4 SP
Filter 512 3x3 Conv Conv = 4 SP
Filter 512 3x3 Conv Conv = 4 SP
FC = 2 Size 1000
Post processing
Feature Extraction
Classification

Fig. 9. The complete achitecture of the proposed pipeline.

We use 15000 alphanumeric characters, including those taken from original plates and others obtained by augmentation. These are split into training, validation, and testing sets, as given in Table III.

TABLE III. DATASET SPLIT OF ARABIC AND ENGLISH ALPHANUMERIC CHARACTERS

Dataset	Split Percentage	Number of Samples
Training	70%	4550
Validation	20%	1950
Testing	10%	1500

G. Post Processing

The dataset used for testing includes some challenging scenarios. In some instances, characters are badly affected by noise, or the plate has a poor condition. Some examples are shown in Fig. 10.

The reconstruction techniques in the existing literature, as discussed earlier, can be used to restore missing and noise-affected individual characters either in English or Arabic string. However, using a separate trained restoration model will increase the whole system's computational cost. We propose a fast and simple solution based on the fact that there is redundant information available in the form of Arabic and English text, and correct recognition of the individual alphanumerics in either of them can lead to overall correct recognition of the LP.

The first step for restoration is to find the location of missing characters in Arabic and English text, and we use a median thresholding based approach for this. It measures the position of each alphanumeric bounding box in both strings relative to the extreme left of the LP and calculates the difference between the successive bounding boxes. If a character is missing, the difference between its left and right detected neighboring boxes would be large. Due to the different shapes of characters, the distance between each pair of neighbors is not the same; however, a missing character makes it much large. We calculate the median value of the distance between detected neighboring boxes and set a threshold of 1.5 times the median distance to detect the missing characters. If the distance is more than 2.5 times the median distance, we assume two characters are missing. However, to explain the algorithm, we assume that only one character is missing between two successfully detected characters. For this case, different steps involved in the restoration process are explained in Fig. 11.

The next step is to restore the missing digit or letter on the spots identified by the median thresholding algorithms above. The exact location of the missing character is assumed to be in the middle of the neighboring characters as shown in Fig. 12.

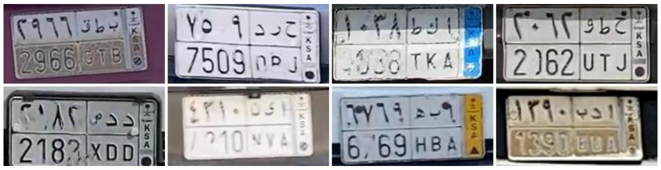


Fig. 10. Badly noise affected license plates.

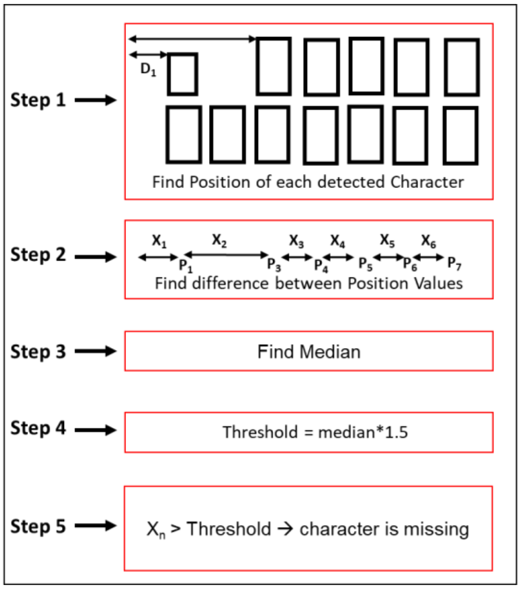

Step 1
D1
Find Position of each detected Character
Step 2
X1 X2 X3 X4 X5 X6
P1 P3 P4 P5 P6 P7
Find difference between Position Values
Step 3
Find Median
Step 4
Threshold = median*1.5
Step 5
Xn > Threshold → character is missing

Fig. 11. Working of median threshold-based algorithm for locating missing characters.

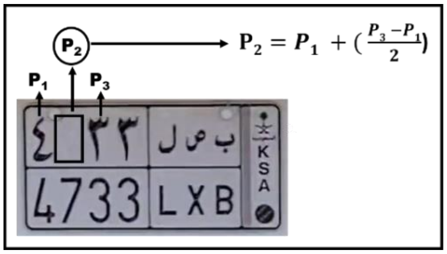
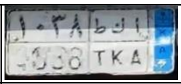


Fig. 12. Finding position of the missing character.

This way, we get seven bounding boxes in each string, regardless the character is detected or missing. The empty boxes are filled by the corresponding vertically mapping characters in the other string. Except for a rare scenario where the same character is missing in both English and Arabic strings, the algorithm can successfully recognize the plate quite accurately, as shown in the experimental results in the next section.

1) Handling conflicts between two strings: Arabic and English alphanumerics have one-to-one matches. To have recognition results with greater precision, we pick the alphanumeric, which is recognized with a higher confidence level by CNN. The string obtained this way is a mix of Arabic and English characters as shown in Table IV. We convert the entire string to the desired language, which is English in our case as shown in the table.

TABLE IV. EXAMPLES OF IMPROVED RECOGNITION RESULTS OF THE PROPOSED ALGORITHM

License Plates			
Mixed Strings	١٠٣٨TKA	1٨٠٦XHJ	21٨٢XDD
Converted Strings	1038TKA	1806XHJ	2182XDD

2) Handling misrecognition due to bolts on the lp: Among many challenging scenarios we encountered in our dataset, one worth mentioning is the wrong recognition of alphanumerics due to the bolts used to attach the plate to the car. There are two bolts used on the top-right and top-left regions of the plate. The location of the right bolt is such that if there is the Arabic character "Alif" (Equivalent of English character A) in the rightmost position, it is read as English "9", as shown in Fig. 13. This is, however, easy to correct, knowing that the rightmost alphanumeric cannot be a digit and it must be a character. So, if a nine is detected at this position, there is a good chance it should be "Alif". If the English string does not detect anything at the rightmost location, we take it as "Alif" or "A". However, if the English string detects something different, we include that in the final result.

Correcting the left bolt is not that straightforward, which happens to be between the first two characters in the Arabic text. If a "9" is detected as the second digit from the left, it could be actually "9" or "1". We examined several cases and found that the head of true "9" has a hollow space, whereas the head formed by the bolt is solid black. We have shown both cases side by side in Fig. 14. Our algorithm crops the head when a "9" is detected at the second place from the left and examines it further. A simple count of black pixels in the binarised image of the head can reveal if it was all filled with back pixels or had a hollow white space.

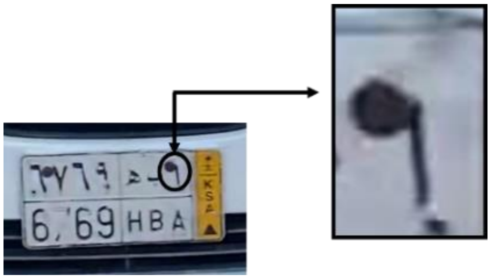


Fig. 13. Bolt Overlapping onto Arabic corresponding character of A.

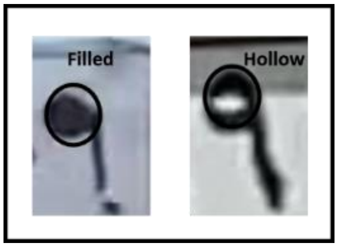


Fig. 14. One misread as nine, shown on the left, and a true 9 in Arabic shown on the right.

IV. Experimental Evaluation

Training and validation are done using the official notebook repository of YOLOv5, which provides a powerful GPU for fast processing. The YOLOv5 model has 476 layers. The batch size and epoch values were set to 35 and 100, respectively. The accuracy of detection can be determined by the overlap between the annotated (ground truth) mask and detected plates. The ratio is called Intersection Over Union (IOU). We considered different values of IOU and the optimal detection results are achieved at IOU>0.5. The rectangles predicted by the YOLOv5 model below this value are discarded.

For training the proposed CNN model, a computer with moderate specifications – GPU GeForce GTX 1080 GPU, 8 GB memory – running on the Linux operating system of Ubuntu 20.04.3 LTS was used. The learning rate was set to 0.001.

The proposed system achieved significantly higher accuracy when both Arabic and English alphanumerics of the LPs were used as discussed above, compared to the accuracy of either string recognized individually. We tested a few commercially available tools as well. On good quality images, all of them achieved high accuracy. However, on challenging cases consisting of real traffic scenarios in different conditions, commercially available tools performed very poorly, and our proposed model outperformed them by a clear margin. We show some results in Tables V to VII. In the shown LP text, the red color shows a wrongly recognized character, the black color shows a correctly read character, and the blue color shows a successfully restored character.

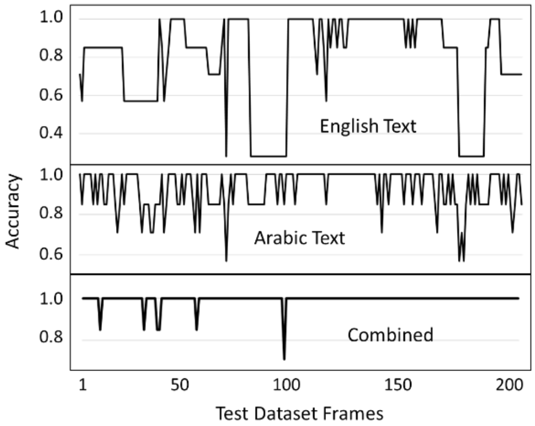


Fig. 15. Accuracy of English and Arabic strings recognition and the combined results for each LP in our test dataset.

A. *Plates with Extremely Poor Visibility*

This test dataset consists of cases where visibility of alphanumeric in the LP plate was poor due to shadows or glare. Many of the alphanumerics were wrongly recognized, but when we considered both resultant strings, the actual LP text was successfully restored. Some examples are shown in Table V A and V B. Note that all three commercial software failed in all these cases, whereas the proposed successfully recognized all plates.

TABLE V. A. Results of Extremely Poor Visibility Cases

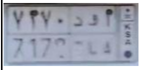
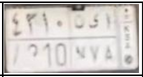
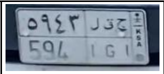
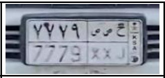
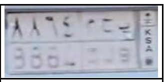
Model				
Sighthound	7172LA	8302KEJ	3636NHB	4810VVA
OpenALPR	71727LA	6302KEJ	3636NHB	4810VA
Plate Recognizer	7172ZGJ	6302KEJ	3636NHB	4310VA
Proposed	7170DLA	5802KEJ	8636NHB	4210NVA

TABLE V. B. MORE RESULTS OF EXTREMELY POOR VISIBILITY CASES

Model			
Sighthound	5941G1	7779KXJ	3864B
OpenALPR	594LG1	7778KXJ	386B
Plate Recognizer	584LG1	7779XXJ	8664JB
Proposed	5943LGJ	7779XXJ	8864ZJB

B. Deteriorated Alphanumeric Characters

Our dataset contains some cases where alphanumerics are degraded partially or fully causing missing a few alphanumerics at the recognition stage. We show some such cases in Table VI. Again, the proposed system outperforms the commercial systems. The missing characters are marked by an asterisk in red color.

C. Mud Affected Plates

There are certain LPs badly affected due to aging or covered by mud, and plate detection becomes challenging in such cases. We show some examples in Table VII. Our system not only detects these plates but also recognizes the text. Only in the last plate, which is challenging even for the human eye, our system made one mistake out of seven characters. The other methods could not even detect that plate.

It would be interesting to see the performance of our system on English and Arabic parts of the LPs separately and combined. Fig. 15 shows the accuracy in all three scenarios for each of the 200 plates in the test dataset. Note that one plate has seven alphanumerics. If all of them are correctly recognized, we consider the accuracy to be 1. If 6 of them are recognized correctly, the accuracy is 6/7, and so on. It can be seen in the figure that the combined results, except for a few cases, could recognize all the characters correctly.

We also present the results of the individual alphanumeric recognition in Table VIII. The first column shows the method used for recognition. In 200 plates in our test dataset, there are 1400 characters in total. The second column shows how many of these were recognized correctly. The third column shows the number of cases when all seven alphanumerics in the plate were correctly recognized. This also gives the system accuracy in perfectly recognizing plates, which is shown in the last column as a percentage. The rest of the columns further break down the cases when less than seven alphanumerics were recognized correctly. It can be seen that the proposed system identified 194 plates out of 200 perfectly while making one mistake in five plates and two mistakes in one plate. The other methods performed much poorly in comparison. The closest competitor Sighthound could recognize only 110 plates.

TABLE VI. RESULTS OF DETERIORATED ALPHANUMERICS

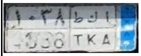
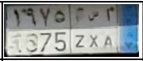
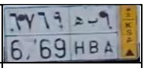
Model				
Sighthound	*7806HJ	*038TKA	*875ZXA	6*69HBA
OpenALPR	*806ZHJ	**38TKA	*375ZXA	6769HBA
Plate Recognizer	*806AHJ	*588TKA	*375ZXA	6*69HBA
Proposed	1806XHJ	1038TKA	1975ZXA	6769HBA

TABLE VII. RESULTS OF MUD AFFECTED PLATES

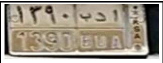
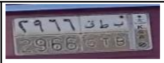
Model			
Sighthound	LP Not Detected	LP Not Detected	LP Not Detected
OpenALPR	LP Not Detected	LP Not Detected	LP Not Detected
Plate Recognizer	LP Not Detected	P9666VB	LP Not Detected
Proposed	1390BDA	2966GTB	79B8*UA

TABLE VIII. OVERALL RECOGNITION RESULTS USING CHALLENGING DATASET OF 200 LPS

Method	Correctly recognized characters (out of 200x7=1400)	Character recognition accuracy	Correctly recognized plates (out of 200)	Plate recognition accuracy
Proposed (English Only)	1106	**79%**	83	41.5%
Proposed (Arabic Only)	1303	**93.1%**	120	60%
Proposed (Combined)	**1393**	**99.5%**	**194**	**97%**
OpenALPR	910	**65%**	55	27.5%
Plate Recognizer	1030	**73.6%**	78	39.5%
Sighthound	1015	**72.5**	110	55%

V. CONCLUSION

This paper proposed a CNN model that can recognize English and Arabic text in the license plates used in Saudi Arabia. The two were combined using a proposed algorithm to correctly restore missing or wrongly read alphanumerics in either of the strings. These methods successfully recognized the license plates where commercially available solutions OpenALPR, Plate Recognizer, and Sighthound failed. The proposed system is computationally efficient and works in real-world unconstrained situations.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number MoE-IF-G-20-12.

REFERENCES

[1] V. R. Greati, V. C. T. Ribeiro, I. M. D. da Silva, and A. de Medeiros Martins, "A Brazilian license plate recognition method for applications in smart cities," in 2017 IEEE First Summer School on Smart Cities (S3C), 2017, pp. 43-48.

[2] A. K. Haghighat, V. Ravichandra-Mouli, P. Chakraborty, Y. Esfandiari, S. Arabi, and A. Sharma, "Applications of deep learning in intelligent transportation systems," Journal of Big Data Analytics in Transportation, vol. 2, no. 2, pp. 115-145, 2020.

[3] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Robust license plate recognition using neural networks trained on synthetic images," Pattern Recognition, vol. 93, pp. 134-146, 2019.

[4] R. Laroca et al., "A robust real-time automatic license plate recognition based on the YOLO detector," in 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-10.

[5] M. Spanu et al., "Smart Cities Mobility Monitoring through Automatic License Plate Recognition and Vehicle Discrimination," in 2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2021, pp. 1-6.

[6] J. Li, H. Van Zuylen, Y. Deng, and Y. Zhou, "Urban travel time data cleaning and analysis for Automatic Number Plate Recognition," Transportation Research Procedia, vol. 47, pp. 712-719, 2020.

[7] Z. Li et al., "License Plate Detection and Recognition Technology for Complex Real Scenarios," in International Conference on Intelligent Computing, 2020, pp. 241-256.

[8] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated license plate recognition: a survey on methods and techniques," IEEE Access, vol. 9, pp. 11203-11225, 2020.

[9] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time Bhutanese license plate localization using YOLO," ICT Express, vol. 6, no. 2, pp. 121-124, 2020.

[10] S. Montazzolli and C. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in 2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), 2017, pp. 55-62.

[11] S. Alghyaline, "Real-time Jordanian license plate recognition using deep learning," Journal of King Saud University-Computer Information Sciences, 2020.

[12] V. Khare et al., "A novel character segmentation-reconstruction approach for license plate recognition," Expert Systems with Applications, vol. 131, pp. 219-239, 2019.

[13] M. Lin, L. Liu, F. Wang, J. Li, and J. Pan, "License Plate Image Reconstruction Based on Generative Adversarial Networks," Remote Sensing, vol. 13, no. 15, p. 3018, 2021.

[14] Z. Li, G. Han, S. Xiao, and X. Chen, "MAP-based single-frame super-resolution image reconstruction for license plate recognition," in 2009 International Conference on Computational Intelligence and Software Engineering, 2009, pp. 1-5.

[15] Y. Chen et al., "Image super-resolution reconstruction based on feature map attention mechanism," Applied Intelligence, pp. 1-14, 2021.

[16] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMs," arXiv preprint arXiv:.05610, 2016.

[17] R. M. Khoshki and S. Ganesan, "Multiscale adaptive nick thresholding method for alpr system," Entropy, vol. 4, no. 10, 2015.

[18] M. M. Mokji and S. A. Bakar, "Adaptive thresholding based on co-occurrence matrix edge information," in First Asia International Conference on Modelling & Simulation (AMS'07), 2007, pp. 444-450.

[19] SightHound. (September 2, 2020). Commercially available tool. Available: https://www.sighthound.com/.

[20] PlateRecognizer. (November 21, 2020). Commercially Available Tool. Available: https://platerecognizer.com/.

[21] OpenALPR. (December 22, 2020). Commercially Available Tool. Available: https://github.com/openalpr/openalpr.

[22] R. Chen and Y. Luo, "An improved license plate location method based on edge detection," Physics Procedia, vol. 24, pp. 1350-1356, 2012.

[23] J.-Y. Sung and S.-B. Yu, "Real-time Automatic License Plate Recognition System using YOLOv4," in 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2020, pp. 1-3.

[24] I. R. Khan, S. T. A. Ali, A. Siddiq, M. M. Khan, M. U. Ilyas, S, Alshomrani, S. Rahardja, "Automatic License Plate Recognition in Real-World Traffic Videos Captured in Unconstrained Environment by a Mobile Camera," Electronics, vol. 11, no. 9, pp. 1408, 2022.

[25] I. Kilic and G. Aydin, "Turkish vehicle license plate recognition using deep learning," in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-5.

[26] S. E. Ozbay Ergun, "Automatic vehicle identification by plate recognition," World Academy of Science, Engineering Technology, vol. 9, no. 41, pp. 222-225, 2005.

[27] Steve. (October 20, 2019). Types of License Plates in Saudi Arabia. Available: https://lifeinsaudiarabia.net/types-of-number-plates-in-saudi-arabia/.

[28] G. Jocher. (July, 2019). YOLOV5, An Open-Source Detection Model Repository Available: https://github.com/ultralytics/yolov5.

[29] T.Lin. (March 29, 2018). LabelImg, Annotation Tool. Available: https://github.com/tzutalin/labelImg.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[31] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781-10790.

[32] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390-391.

[33] ProgrammerSought. (July, 2020). YOLOV5 learning summary.

[34] J. Wen-ping and J. Zhen-cun, "Research on early fire detection of Yolo V5 based on multiple transfer learning," Fire Science Technology, vol. 40, no. 1, p. 109, 2021.

[35] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," Artificial Intelligence Review, vol. 53, no. 8, pp. 5455-5516, 2020.

A Real-time ECG CTG based Ensemble Feature Extraction and Unsupervised Learning based Classification Framework for Multi-class Abnormality Prediction

Aditya.Y[1], Dr. S.Suganthi Devi[2], Dr. B.D.C.N Prasad[3]
Research Scholar, Department of Computer Science and Engineering-Faculty of Engineering and Technology, Annamalai University[1]
Lecturer, Srinivasa Subbaraya Polytechnic College, Puttur, Tamil Nadu[2]
Professor, Department of Computer Applications, VR Siddhartha Engineering College, Vijayawada[3]

***Abstract*—Cardiovascular diseases (CVDs) are a leading cause of death worldwide. Early detection and diagnosis of these diseases can greatly reduce complications and improve outcomes for high-risk individuals. One method for detecting CVDs is through the use of electrocardiogram (ECG) monitoring systems, which use various technologies such as the Internet of Things (IoT), mobile applications, wireless sensor networks (WSN), and wearable devices to acquire and analyze ECG data for early diagnosis. However, despite the prevalence of these systems in the literature, there is a need for further optimization and improvement of their classification accuracy. In an effort to address this challenge, a novel heterogeneous unsupervised learning model for real-time ECG classification was proposed. The main goal of this work was to reduce the error rate and improve the classification accuracy of the system. This study presents a framework for the classification of multi-class abnormalities in electrocardiograms (ECGs) using an ensemble feature extraction technique and unsupervised learning. The framework utilizes a real-time electrocardiogram-cardiotocography (ECG-CTG) system to extract features from the ECG signal, and then employs an ensemble of feature extraction techniques to enhance the discrimination of the extracted features. The extracted features are then used in an unsupervised learning-based classification algorithm to classify the ECG signals into different classes of abnormalities. The proposed framework is evaluated on a dataset of ECG signals and the results show that it can effectively classify ECG signals with high accuracy and low computational complexity.**

Keywords—Ensemble; feature ranking; improved inter quartile range; outlier detection; heterogeneous optimized k-nearest neighbor; unsupervised learning

I. Introduction

According to research data from the National Family Health Survey, people living in rural areas of India are disproportionately affected by cardiovascular diseases (CVDs) compared to those in urban areas. This is due to factors such as lower income and lack of access to healthcare infrastructure. To address this issue, there is growing interest in developing low-cost tools and techniques for detecting CVDs in a timely and accurate manner. The utilization of IoT and machine learning in healthcare presents a promising solution, enabling remote diagnosis of patients and identifying patterns in vast amounts of medical data. Nonetheless, there is still room for improvement in accurately diagnosing patients by classifying ECG signals. This research aims to address this issue by developing a reliable ECG monitoring system that utilizes IoT and signal classification to enhance diagnosis rates. The system utilizes an AD8232 biopotential sensor to capture real-time ECG data, which is then transmitted to an AWS IoT core through a NodeMCU ESP8266 gateway and MQTT protocol. In monitoring fetal well-being during pregnancy, fetal cardiotocogram (CTG) and fetal electrocardiogram (FECG) are two critical tools. CTG, a non-invasive technique, measures fetal heart rate (FHR) and uterine contractions through the use of a tocodynamometer and an ultrasound transducer placed on the mother's abdomen. It is typically performed during the latter part of pregnancy to evaluate fetal well-being and detect abnormalities, such as fetuses at risk for distress, which can lead to poor outcomes such as stillbirth or neonatal death. On the other hand, FECG is an invasive technique that records the electrical activity of the fetal heart and is usually performed during the third trimester of pregnancy [1]. The process of fetal electrocardiogram (FECG) involves inserting a small electrode into the amniotic fluid surrounding the fetus, which records the electrical activity of the fetal heart to detect any abnormalities in the fetal heart rate (FHR). Compared to cardiotocogram (CTG), FECG is considered to be a more accurate method of assessing fetal well-being as it can detect subtle changes in the FHR that may not be visible on a CTG trace. Both CTG and FECG have their own advantages and limitations. While CTG is a non-invasive technique that is easy to perform and does not pose any risks to the mother or fetus, it is not as accurate as FECG in detecting fetal distress. FECG, on the other hand, is a more accurate method of assessing fetal well-being, but it is invasive and carries a small risk of infection or bleeding.

In conclusion, CTG and FECG are two important tools used in the monitoring of fetal well-being during pregnancy. CTG is a non-invasive technique that is easy to perform and does not pose any risks to the mother or fetus, while FECG is a more accurate method of assessing fetal well-being, but it is invasive and carries a small risk of infection or bleeding. Both techniques play an important role in the assessment of fetal well-being, and when used together, they can provide a more comprehensive picture of the fetus's health [2].

The collected data is preprocessed to remove outliers, and features are extracted using statistical and advanced filtering techniques. An ensemble learning model is then employed to optimize the prediction rate on the segmented classes. Regarding the research approach, different methods such as deductive or inductive, qualitative and quantitative can be used. Thoroughly studying the existing literature and research, the hypothesis formulation suggests a deductive research approach. The purpose of this study is to improve the overall lifetime of ECG measurement and its recognition and classification, and the qualitative approach is found to be the most appropriate [3]. In a later stage, ECG classification was performed and some abnormalities were detected. A test was conducted on 50 ECG signals with a duration of 2.5 seconds, and the application of certain techniques led to a significant improvement in baseline stability. ECG histograms showed minimal baseline drift during the recording phase after reducing baseline drift noise. To validate the estimation processes, 10 ECG signals with artificial baseline drift noise were created and analyzed using correlation and mean square error calculations. Farrell et al. [4] continue to explain the wavelet variance to wavelet packets in their work, in which they use the wavelet packets iterative CSS algorithm to locate variance change points. As a result, their method can be applied to a large variety of processes. The primary aim of this research is to address the disproportionately high rates of cardiovascular diseases (CVDs) in rural areas of India by developing a low-cost, IoT-enabled ECG monitoring system that uses signal classification to improve diagnosis rates. In recent years, several methods have been proposed in literature to enhance the recognition of premature ventricular contractions and other heart diseases from normal beats using electrocardiogram (ECG) signals. One such system proposed by [5] consists of three stages: denoising, feature extraction, and classification. The denoising stage deploys the Stationary Wavelet Transform to remove noise from the ECG signal, while the feature extraction stage combines morphological-based features and timing interval-based features to extract relevant information from the signal. Finally, multiple classifiers such as Multi-layer perceptron neural networks (MLP), probabilistic neural networks (PNN), and support vector machines (SVM) are used to classify the ECG beats. Among these classifiers, SVM achieved the highest classification accuracy of 97% [6]. Another study [7] addresses the issue of baseline drift noise in ECG signal processing by employing the Discrete Wavelet Transform. This transform effectively demonstrates non-stationary signals such as ECG signals. The proposed method was tested using ECG signals from the MIT-BIH arrhythmia database and proved to be effective in eliminating 60Hz artifacts with minimal ECG signal distortion. Other methods have also been proposed in literature to reduce noise and improve the quality of ECG signals, including the use of multirate architecture with a linear phase lowpass filter, Butterworth and Chebyshev I filters, wavelet transform method and a neural network based on adaptive filters, artificial neural network for automated noise removal, IIR Zero phase filtering, FIR and IIR filters, particle swarm optimization and support vector machine classifier for wavelet-based representation of ECG beats, an algorithm using a discrete wavelet transform, extreme learning machine and support vector machine for classifying four different types of heart beats, automated medical diagnostic tool using the cross-spectral density approach and least square support vector machine classification algorithm, and a power spectral-based hybrid support vector machine-genetic algorithm to categorize five different types of ECG beats [8].

In addition to these methods, several studies have also proposed the use of neural networks and other machine learning techniques for ECG beat classification and heart disorder diagnosis. One such study used a neural network model with stacked generalization method, resulting in an error rate of 12.41%. Another study evaluated the performance of various classifiers, including Kth Nearest Neighbor Rule, neural networks, discriminant analysis, and fuzzy logic, using 26 morphological parameters as the focus features. A third study proposed an Artificial Neural Network (ANN) based system for the diagnosis of cardiac arrhythmia using standard 12-lead ECG signal recordings. In all of these studies, the MITBIH database was used to evaluate performance, and the results were found to be satisfactory [9].

In [10], a combination of a convolutional neural network (CNN) and a recurrent neural network (RNN) was proposed for ECG beat classification. The authors used the PTB Diagnostic ECG Database to train and test their model, achieving an overall accuracy of 99.2%. These studies demonstrate the effectiveness of using machine learning techniques for ECG beat classification and heart disorder diagnosis, and highlight the importance of continuing research in this field.

The author [11] developed a method of detecting the QRS of the fetus by combining a time-varying Finite Impulse Response (FIR) filter with a genetic algorithm. They found that the filter coefficients reduced the quadratic error and ensured convergence towards the optimal filter. To compare the effectiveness of the Genetic Algorithm (GA) with other filters such as Wiener, Recursive Least Mean Square (RLMS), and Normalized Least Mean Square (NLMS), a realization and comparison were performed using the same filter coefficients with real ECG signals acquired from the abdomen of the mother. The extraction accuracy was improved by changing the order of the filter and the NLMS algorithm gave good quality performances when compared to other filters. However, if the gain of adaptation was large, there was a risk of oscillations. The research [12] introduced a method of extracting Fetal Electrocardiogram (FECG) based on an adaptive linear neural network. The results showed that the adaptive linear neural network could be used to extract FECG from the maternal abdominal signal effectively. The improvement of the network structure made the network error more close to the maternal ECG (MECG), thus a clearer FECG could be acquired. A clearer FECG could be extracted by improving neural network parameters. The study [13] proposed a new methodology that combined Artificial Neural Network (ANN) and correlation approach. Nonlinear and time-varying features of the ECG signal had to be adapted using an Artificial Neural Network. It required a desired output in order to learn, hence it used supervised Multilayer Perception (MLP) network. Likewise, to scale the MECG when subtracting it from the AECG, in order to get the FECG the correlation method was chosen as the correlation factor. The ANN and correlation combination gave an improved and efficient result in terms of accuracy for FECG extraction and R peak detection. The author in [14] presented a method for extracting FECG using Adaptive Neuro-Fuzzy Inference System (ANFIS). The method involves collecting ECG signals from two electrodes, one placed at the thoracic area (completely maternal) and the other at the abdominal area (composite of maternal and fetal ECG signal). Accurate placement of the electrodes is crucial for the application of this method. ANFIS was used for nonlinear alignment of the MECG signal with the components of MECG in the abdominal signal. Then, the maternal components of the abdominal signal were cancelled, and finally the FECG signal was extracted. The algorithm was tested using synthetic and real ECG data, and in both cases, good FECG extraction was achieved, even in the presence of full overlapping maternal and fetal signals. This improved the application of wavelet transform to FECG signals extracted by polynomial networks. Both synthetic and real-time data were pre-processed and post-processed using wavelet denoising algorithms. This method effectively removed baseline wandering, and the extraction performance was successful and improved. For real FECG, visual results also showed that wavelet denoising was useful. The research [15] proposed a new methodology that combined Artificial Neural Network (ANN) and Correlation (ANNC) approach. This method tried various learning constant values and momentum for FECG signal extraction from the abdominal signal and proved that changing the learning rate and momentum also affect the output of the network. This technique was found to be robust and effectively extract the FECG signal from the abdominal signal with an accuracy of 95% and performance of 93.75%. In summary, these studies demonstrate the effectiveness of using neural networks and deep learning techniques for ECG beat classification and diagnosis of heart disorders. These techniques have been shown to achieve high accuracy and are promising for use in clinical settings.

A. *Research Gap*

One potential research gap for the content on real-time ECG-CTG detection using machine learning is the lack of focus on the scalability and generalizability of the proposed techniques. Most of the studies cited in the content are focused on improving the accuracy of ECG classification using specific datasets or databases. However, there is a need to evaluate the performance of these techniques on a larger and more diverse set of data to determine their potential for wider adoption in clinical settings. Additionally, the content could benefit from more exploration of the challenges and limitations of applying machine learning techniques to ECG-CTG detection, such as issues related to data quality, interpretability, and ethical considerations. Finally, there may be opportunities to investigate the integration of ECG-CTG detection with other healthcare technologies, such as telemedicine or wearable devices, to improve patient outcomes and reduce healthcare costs.

The paper is structured as follows: In Section II, the related works of ECG+CTG models and their limitations are presented. Section III outlines the proposed solution for ECG+CTG using machine learning. Section IV provides details on the experimental results and analysis. Finally, in Section V, the paper is concluded.

II. Related Works

The detection and analysis of fetal electrocardiogram (FECG) signals is a crucial tool in evaluating the health and status of a fetus during labor. However, extracting the FECG signal alone from complex data contaminated by various types of noise such as maternal ECG, electromyogram, power line interference, and mother's respiration is challenging. In recent years, researchers have proposed various methods to improve the accuracy and reliability of monitoring the fetal heart rate during contractions. One such method is the combination of a time-varying Finite Impulse Response (FIR) filter with a genetic algorithm, developed by Talha and colleagues in 2010. The filter coefficients were found to reduce the quadratic error and ensure convergence towards the optimal filter. Realization and comparison were performed using the same filter coefficients with real ECG signals acquired from the abdomen of the mother. The extraction accuracy was improved by changing the order of the filter and the Normalized Least Mean Square (NLMS) algorithm gave good quality performances when compared to other filters, such as Wiener, Recursive Least Mean Square (RLMS), and Normalized Least Mean Square (NLMS). However, this method has lower efficiency in removing noise signals compared to other methods [16].

Another method proposed by [17] is a hybrid ECG arrhythmia classification approach, known as MRFO-SVM. This approach combines various ECG signal descriptors based on one-dimensional local binary patterns (LBP), wavelet, higher-order statistical (HOS), and morphological information for feature extraction. The approach utilizes a metaheuristic algorithm, known as Manta Ray Foraging Optimization (MRFO), for feature selection and classification processes. However, this approach could be further improved by integrating MRFO with other machine learning techniques such as convolutional neural networks (CNN) and deep neural networks (DNN) to enhance the detection of arrhythmia and heart rate abnormalities, as well as by hybridizing MRFO with other metaheuristic algorithms. The study [18] evaluated a segment-based stacking method of CNN and SVM to classify short single-lead ECG signals into four classes: Normal, AF, Others, and Noise. Landry et al. proposed a novel embedded QRS complex detection algorithm based on the ECG signal strength and its trend. Mourad et al. used wavelet transforms to detect QRS complexes, and Rahul et al. proposed a window-based FIR filter to eliminate high-frequency noise in ECG signals. Yang et al. proposed a 12-lead ECG arrhythmia classification method using a cascaded convolutional neural network (CCNN) and expert features. While these methods have shown promise, limitations and issues still exist, such as difficulty in detecting R-waves with slow variations and when preceded by waves with strong amplitudes, the need for high computational memory and time for large numbers of features and signals, and the need for further research to utilize the

QRS-complex for the detection of various cardiac arrhythmias, the detection of other waves in the cardiac cycle, and the possibility of low-cost hardware implementation for early detection of cardiovascular disorders. In recent years, several technologies have been employed to design and implement ECG monitoring systems for remote monitoring of cardiovascular diseases (CVDs). These include the Internet of Things (IoT), mobile applications, wireless sensor networks (WSN), and wearable devices. For example, Serhani et al. proposed an IoT-based CVD monitoring system that facilitates ECG data acquisition and continuous remote monitoring and analysis of patients, with the collected data transmitted to the cloud for further investigation by specialists for early diagnosis. Similarly, [19] developed a portable ECG monitoring system based on Arduino-Uno and an AD8232 sensor to monitor the cardiac health condition of patients. This proposed a system that continuously monitors the temperature, pulse rate, and ECG of patients, generating an alert SMS to the caretaker's mobile if the values exceed normal limits. They developed a wireless real-time ECG monitoring system for the early detection of CVDs, while Mishra et al. proposed an IoT-based smart healthcare system with an AD8232 heart rate sensor interfaced with Arduino UNO and connected to the cloud using an ESP8266 wireless LAN module for remote monitoring. They proposed an e-health monitoring system that measures body temperature, blood oxygen saturation, ECG signal, and heart rate, sending the data to an IoT cloud for remote analysis by a doctor. The study [12] implemented an IoT-based vital sign monitoring system using Raspberry Pi 3 to monitor body temperature, pulse rate, and heartbeat using ECG, and Deep neural networks for the analysis and classification of normal and abnormal beats. Despite the prevalence of these studies in the literature, there are relatively few studies that have analyzed and classified signals to design a complete healthcare system. One of the major challenges in bio-signals processing is the high variability of bio-signals over time, due to biological processes within the body. This variability often complicates the selection of informative parameters and may yield inaccurate predictions. Outliers, or portions of the signal that deviate excessively from adjacent segments, are a typical phenomenon in bio-signals processing, and the elimination of their impact is crucial in the signal processing channel of ECG-based biometric systems. To address these challenges, researchers have employed various methods for outlier correction and classification of ECG signals. For example, Jun et al. compared the effectiveness of outlier correction methods for ECG signals in combination with various classification algorithms in biometric applications. Aqeel et al. developed an IoT-based ECG signal monitoring and classification system to diagnose the health status of patients, utilizing convolutional neural networks (CNN) and achieving an accuracy of 94.94%. The research [19] proposed a real-time ECG signal analysis and classification approach using discrete wavelet transform (DWT) and support vector machines (SVMs). DWT is used for pre-processing and feature extraction from the MIT-BIH dataset, and the SVM classifies six heartbeat types with an accuracy of 98.61%.Recently, researchers have demonstrated that ensemble systems can increase the performance of base classifiers. Ensemble learning is the process of integrating various base models to improve the overall performance of the system. Ensemble-based ECG classification methods have been proposed in various studies, achieving high accuracy and robustness in detecting and classifying ECG beats.

III. Proposed Model

The proposed model for ECG classification consists of three phases, which are designed to address key challenges in ECG classification and improve the accuracy and reliability of the classification results.

In Phase 1, the model focuses on collecting high-quality CTG and FECG data in real-time. This is important because the accuracy of ECG classification models depends heavily on the quality of the input data, and any noise or artifacts in the data can significantly affect the classification results. By collecting data in real-time, the model ensures that the data is up-to-date and reflects the current state of the patient's heart function.

In Phase 2, the model focuses on extracting relevant features from the ECG and CTG data and filtering out noise and artifacts. This is a critical step in ECG classification because it helps to reduce the complexity of the data and highlight the key characteristics that are important for classification. By using advanced feature extraction and filtering techniques, the model is able to identify and isolate key features that are relevant for classification, while minimizing the impact of noise and artifacts.

In Phase 3, the model uses a cluster-based ensemble classification approach to classify the ECG data. This approach combines the results of multiple classification models to improve the accuracy and reliability of the classification results. By using a cluster-based approach, the model is able to group similar ECG signals together and classify them based on their shared characteristics. This approach can improve the accuracy of classification by reducing the impact of individual classification errors and increasing the overall robustness of the model. The proposed model is implemented in three phases shown in Fig. 1.

- Phase 1: Realtime CTG and FECG data collection.
- Phase 2: ECG+CTG Feature extraction measures and filtering.
- Phase 3: Proposed cluster based ensemble classification.

As depicted in Fig. 1, the data is initially collected from a real-time ECG sensor. The data from each sensor is then sent to AWS cloud storage for further analysis. The machine learning model employs a filtering technique and feature extraction measures to preprocess the data. In this particular study, an enhanced kernel feature ranking measure was implemented to enhance the feature selection process for clustering. A novel clustering approach was also utilized to identify key classes for classification. To improve performance, an ensemble learning framework was employed to reduce the error rate and increase the true positive rate. As depicted in Fig. 1, the data is initially collected from a real-time ECG sensor. The data from each sensor is then sent to

AWS cloud storage for further analysis. The machine learning model employs a filtering technique and feature extraction measures to preprocess the data. In this particular study, an enhanced kernel feature ranking measure was implemented to enhance the feature selection process for clustering. A novel clustering approach was also utilized to identify key classes for classification. To improve performance, an ensemble learning framework was employed to reduce the error rate and increase the true positive rate.

1) Phase 1: Realtime CTG and FECG data collection: CTG is a non-invasive test that uses ultrasound to measure the fetal heart rate and uterine contractions, while FECG is an invasive test that uses electrodes to measure the electrical activity of the fetal heart. Both tests are used to detect any potential problems that may arise during pregnancy, such as fetal distress or abnormal fetal heart rate patterns.Real-time data acquisition for CTG and FECG is essential for data processing. This involves the collection and analysis of data in real-time, as opposed to after the fact. Real-time data acquisition allows for the early detection of any potential problems, which can lead to prompt intervention and better outcomes for both the mother and the fetus.One of the most important aspects of real-time data acquisition for CTG and FECG is the use of advanced technology. High-quality ultrasound machines, specialized software, and sophisticated electrodes are used to collect and analyze data. This technology is able to detect even the slightest changes in the fetal heart rate and contractions, which can indicate potential problems. In addition to advanced technology, real-time data acquisition for CTG and FECG also requires trained professionals to operate the equipment and interpret the results. Obstetricians and gynecologists, as well as specialized nurses and technologists, are responsible for monitoring the data and interpreting the results. They must be able to recognize any abnormal patterns or changes in the data, and take appropriate action to address any potential problems. Data processing is also an important aspect of real-time data acquisition for CTG and FECG. This involves the analysis of the data collected by the equipment, and the identification of any patterns or trends that may indicate potential problems. Data processing is typically done using specialized software, which can analyze the data in real-time and identify any potential issues.

Overall, real-time data acquisition for CTG and FECG is essential for ensuring the health and well-being of both the mother and the fetus during pregnancy. Advanced technology, trained professionals, and data processing are all crucial elements of this process, and must be carefully managed to ensure the best possible outcomes.

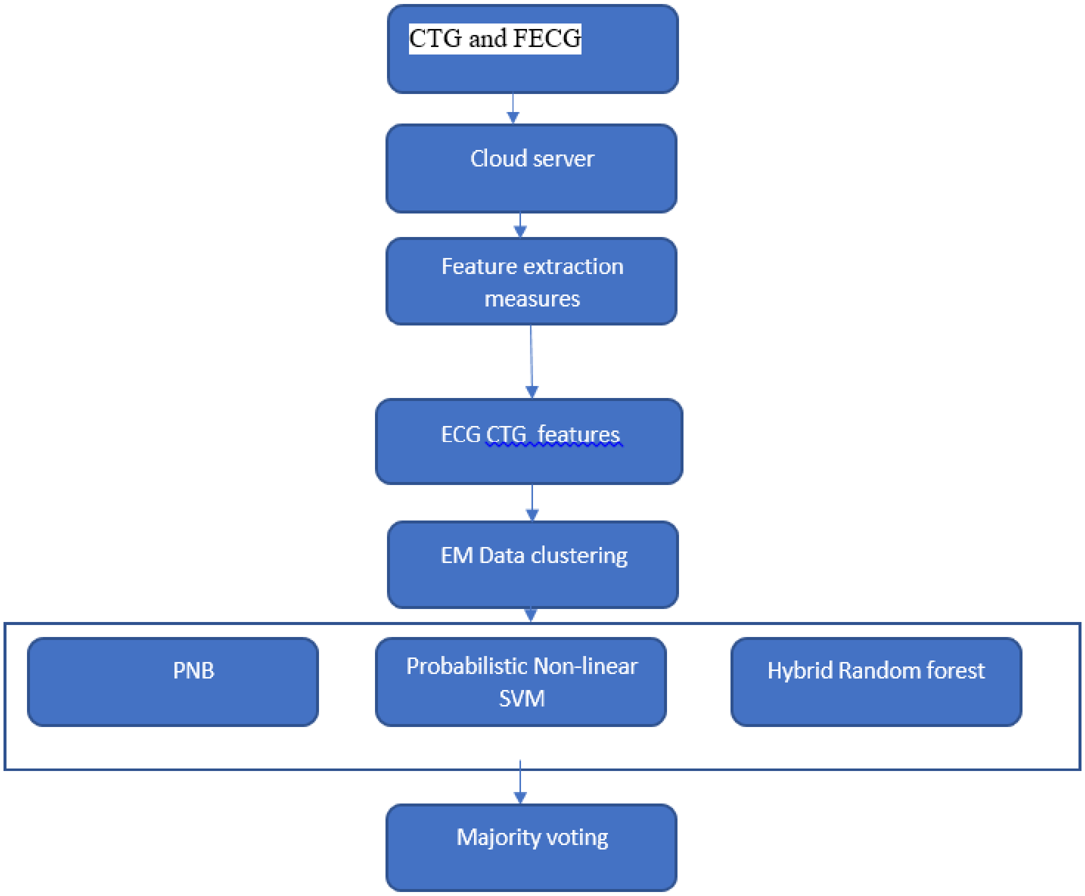


Fig. 1. Proposed framework.

2) Phase 2: Feature extraction measures and filtering: QRS peak detection is an important step in the analysis of electrocardiogram (ECG) signals as it helps to identify the locations of the Q, R, and S waves, which are indicative of the electrical activity of the heart. The following are the steps of a typical QRS peak detection algorithm:

Filtering: The ECG signal is passed through a bandpass filter to remove any noise and high-frequency artifacts. The cutoff frequencies of the filter are typically between 5 and 15 Hz, as the QRS complex is known to occur within this frequency range.

Differentiation: The filtered ECG signal is then differentiated using a differentiation operator, such as a finite difference or a Sobel operator, to enhance the high-frequency components of the QRS complex.

Squaring: The differentiated ECG signal is squared to further enhance the QRS complex and suppress the noise.

Moving Window Integration: The squared ECG signal is then passed through a moving window integrator, such as a rectangular window or a Gaussian window, to smooth the signal and eliminate any remaining noise.

Thresholding: A threshold is set to detect the QRS peaks. The threshold is typically set at a level that is slightly above the baseline noise level. Any sample that exceeds this threshold is considered a QRS peak.

The above described steps are mathematical derivation:

Filtering:

The filtered ECG signal is obtained by convolving the original ECG signal with a bandpass filter function h(t) which is defined as :

$$h(t)= (1/T) * rect((t-T/2)/T) * (sin(2\pi fct)/(\pi fct)) \quad (1)$$

where rect(x) = 1 for |x|<0.5 and 0 otherwise,

Differentiation:

The differentiated ECG signal is obtained by applying the differential operator d/dt to the filtered ECG signal.

Squaring:

The squared ECG signal is obtained by squaring the differentiated ECG signal.

Moving Window Integration:

The smoothed ECG signal is obtained by convolving the squared ECG signal with a moving window function w(t).

Thresholding:

The threshold value is set to a level slightly above the baseline noise level. Any sample that exceeds this threshold is considered a QRS peak.

3) Phase 3: Proposed Cluster based Ensemble classification framework: The Probabilistic Expectation-Maximization (PEM) algorithm is a popular method for clustering data, including ECG signal data. The algorithm consists of two main steps: the Expectation step (E-step) and the Maximization step (M-step). The steps are repeated until convergence, at which point the algorithm has found the maximum likelihood estimates for the parameters of the underlying mixture model.

E-step: In this step, the algorithm estimates the probability that each data point belongs to each of the clusters, given the current estimates of the parameters of the mixture model. This is done by computing the likelihood of each data point, given the current cluster means and covariances, and multiplying this by the prior probability of each cluster. The resulting probabilities are used to update the responsibilities for each data point and cluster.

Mathematically, the E-step is represented by the following equation:

r_{n,k} = P(z_n = k | x_n, mu, Sigma) = frac{P(x_n | z_n = k, mu, Sigma) * P(z_n = k)}{P(x_n)} (2)

where x_n is the nth data point, z_n is the cluster assignment for the nth data point, mu is the mean of the kth cluster, Sigma is the covariance matrix of the kth cluster, and r_{n,k} is the responsibility of the kth cluster for the nth data point.

M-step: In this step, the algorithm updates the parameters of the mixture model (i.e., the means, covariances, and prior probabilities) based on the current responsibilities of the data points. The new parameters are chosen to maximize the expected log-likelihood of the data, given the current responsibilities.

Mathematically, the M-step is represented by the following equations:

mu_k = frac{1}{N_k} sum_{n=1}^{N} r_{n,k} x_n

Sigma_k = frac{1}{N_k} sum_{n=1}^{N} r_{n,k} (x_n - mu_k)(x_n - mu_k)^T

P(z_n = k) = frac{N_k}{N} (3)

where mu_k is the mean of the kth cluster, Sigma_k is the covariance matrix of the kth cluster, P(z_n = k) is the prior probability of the kth cluster, and N_k is the total responsibility of the kth cluster.

The algorithm continues to alternate between the E-step and the M-step until convergence is reached. This can be determined by checking whether the log-likelihood of the data has stopped increasing or if the parameters have not changed significantly between iterations.

This is the basic algorithm for EM clustering, which is useful for identifying patterns and structure in ECG signal data. However, it is worth noting that there are various modifications and extensions of the EM algorithm, such as the Gaussian Mixture Model (GMM) and the soft EM algorithm that can be applied to ECG signal data to improve the performance of the clustering.

A. Proposed Ensemble Classification Learning Model

The Support Vector Machine (SVM) algorithm is a supervised learning algorithm that can be used for classification and regression tasks. One of the key features of SVM is the use of kernel functions, which allows the algorithm to perform nonlinear classification by mapping the input data into a higher dimensional space.

The mathematical derivation for a nonlinear kernel function SVM applied to ECG signal data is as follows:

The input data, which consists of a set of ECG signals, is first mapped into a higher dimensional feature space using a nonlinear kernel function, K(x,y). Commonly used nonlinear kernel functions include the Radial Basis Function (RBF) kernel, the Polynomial kernel, and the Sigmoid kernel.

The equation for the RBF kernel function is: K(x,y) = exp(-γ||x-y||^2)

The equation for the Polynomial kernel function is: K(x,y) = (x.y + c)^d

The equation for the Sigmoid kernel function is: K(x,y) = tanh(γx.y + c) (4)

The optimal hyperplane is then found by maximizing the margin, which is defined as the distance between the closest data points of each class, known as support vectors, and the hyperplane.

The equation for the optimal hyperplane is: wx + b = 0

The decision boundary is given by the equation: f(x) = sign(wx + b).

The SVM algorithm then uses this decision boundary to classify new data points as belonging to one of the classes.

The parameters of the kernel function (such as gamma and the constant term) can be optimized using techniques such as cross-validation to improve the performance of the algorithm.

Finally, the ensemble classification algorithm is performed by combining the decision of multiple base classifiers (SVM, Neural network, optimized Naive Bayesian, and optimized decision tree) using techniques such as majority voting or weighted voting to produce a final prediction.

The optimized decision tree algorithm is a method for building a decision tree model with improved accuracy and reduced overfitting. The following are the steps for building an optimized decision tree for ECG signal data, along with mathematical derivations:

Data preprocessing: The first step is to preprocess the ECG signal data by removing any missing or irrelevant data and scaling the features to a common range.

Feature selection: Next, a feature selection method such as mutual information or wrapper methods can be used to select the most relevant features for the decision tree model.

Splitting criterion: The decision tree algorithm builds the tree by repeatedly splitting the data based on the feature that maximizes the reduction in impurity. A common splitting criterion is the Gini impurity, which is calculated as:

Gini = 1 - Σ(p_i)^2

where p_i is the proportion of data points belonging to class i in a given node.

Pruning: To prevent overfitting, the decision tree can be pruned by removing branches with low information gain or by setting a minimum number of samples required to split a node.

Model evaluation: The final step is to evaluate the performance of the decision tree model using metrics such as accuracy, precision, recall, and F1-score.

Hyperparameter tuning: The final step is to optimize the model by tuning the hyperparameters such as maximum depth, minimum samples per leaf, and minimum samples per split.

Ensemble: Once the decision tree is optimized, it can be combined with other classifiers like SVM, Neural network, optimized Naive bayesian etc to form an ensemble classifier which will lead to an improved overall performance of the model.

The joint probability estimation based naive bayes algorithm for ECG signal data involves the following steps:

Data preprocessing: The ECG signal data is preprocessed to remove any noise or artifacts present in the signal. This can be done using techniques such as filtering, resampling, and baseline correction.

Feature extraction: The ECG signal data is then divided into segments and features are extracted from each segment. These features can include information such as the R-peak amplitude, QRS duration, and P-wave duration.

Joint probability estimation: The joint probability of the features and the class labels is estimated using the extracted features. This can be done using techniques such as maximum likelihood estimation or the method of moments.

Naive bayes classifier: The naive bayes classifier is then trained on the estimated joint probabilities. This classifier assumes that the features are independent given the class label.

Classification: Once the classifier is trained, it can be used to classify new segments of ECG signal data by computing the posterior probabilities for each class label and selecting the label with the highest probability.

Mathematical derivation:

Let's suppose we have D = {(x1,y1),(x2,y2),...,(xn,yn)} as the training data set, where xi is the feature vector of i-th segment and yi is the corresponding class label.

The joint probability of feature vector xi and class label yi can be defined as

P(x,y) = P(x|y)P(y)

The naive bayes classifier assumes that the features are independent given the class label, so we can write

P(x|y) = Πi=1n P(xi|y)

The class label with the highest probability will be the predicted class label

P(y|x) = P(x|y)P(y) / P(x) = P(x|y)P(y) / Σy' P(x|y')P(y') (5)

where y' is a class label

The optimized decision tree algorithm will have similar steps but with a different mathematical derivation for the decision tree.

IV. Experimetnal Results

Experimental results are evaluated on real-time ECG+CTG signal data in order to predict the abnormality of the patient.

CTG Data:

Fetal heart rate (FHR): This is the number of times the fetus' heart beats per minute. It is typically measured using ultrasound or a cardiotocograph (CTG) machine.

Fetal heart rate variability (FHRV): This is the variation in the time interval between successive fetal heartbeats. It can be measured using ultrasound or a CTG machine.

Uterine contractions: These are the rhythmic, involuntary contractions of the uterus that occur during labor. They can be measured using a tocodynamometer.

FHR acceleration: This is an increase in the FHR above the baseline that lasts for at least 15 seconds. It can be measured using ultrasound or a CTG machine.

FECG Data:

Fetal ECG: This is the electrical activity of the fetus' heart. It can be measured using electrodes placed on the mother's abdomen.

Fetal heart rate: Same as above

Fetal QRS complex: This is the combination of the Q, R, and S waves of the fetal ECG. It can be used to assess the fetal cardiac function.

Fetal QT interval: This is the duration of the QT interval of the fetal ECG. It can be used to assess the fetal cardiac function.

The result represents the test classification recall of the proposed model on the selected features subset using ensemble learning framework. From the results it is noted that the proposed ranked based classification has better recall than conventional approaches on realtime data1 as shown in Fig. 2.

The result represents the test classification accuracy of the proposed model on the selected features subset using ensemble learning framework. From the results it is noted that the proposed ranked based classification has better accuracy than conventional approaches on SSDS data as shown in Fig. 3.

The result represents the test classification precision of the proposed model on the selected features subset using ensemble learning framework. From the results it is noted that the proposed ranked based classification has better precision than conventional approaches on SSDS data as shown in Fig. 4.

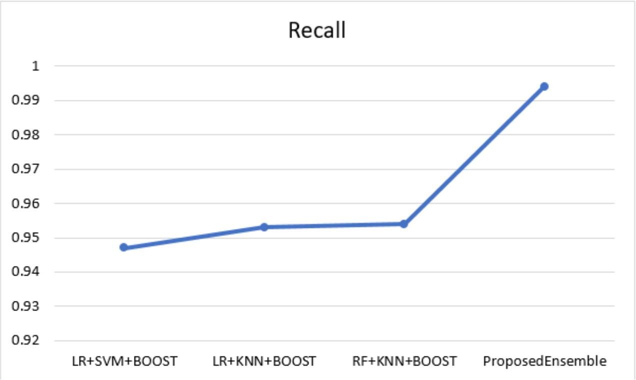


Fig. 2. Comparative analysis of recall for ECG+CTG based classification models.

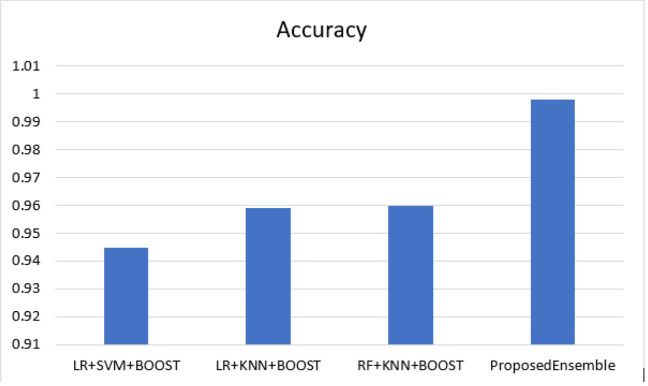


Fig. 3. Comparative analysis of accuracy for ECG+CTG based classification models.

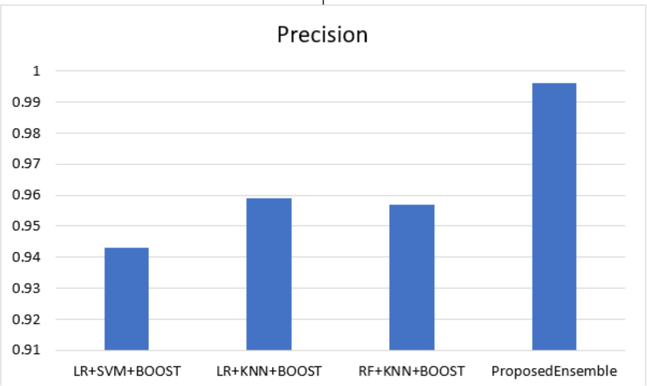


Fig. 4. Comparative analysis of precison for ECG+CTG based classification models.

The result represents the test classification F-measure of the proposed model on the selected features subset using ensemble learning framework. From the results it is noted that the proposed ranked based classification has better F-measure than conventional approaches on SSDS data as shown in Fig. 5.

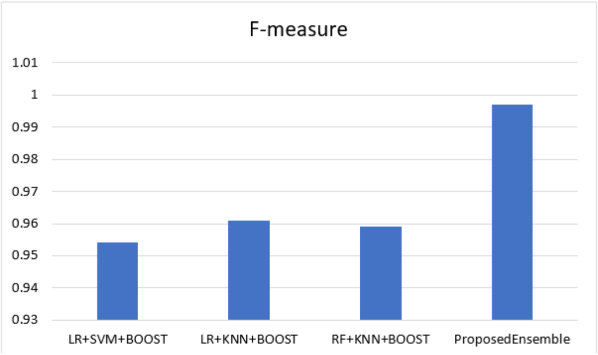


Fig. 5. Comparative analysis of recall for ECG+CTG based classification models.

TABLE I. COMPARATIVE ANALYSIS OF PROPOSED MODEL TO CONVENTIONAL MODELS ON DATASET2

Accuracy	**Samples**	**LR+SVM+BOOST**	**LR+KNN+BOOST**	**RF+KNN+BOOST**	**ProposedEnsemble**
	TestData-1	0.959	0.947	0.955	0.989
	TestData-2	0.956	0.945	0.963	0.991
	TestData-3	0.956	0.943	0.965	0.99
	TestData-4	0.96	0.948	0.962	0.991
	TestData-5	0.954	0.946	0.952	0.989
	TestData-6	0.954	0.947	0.952	0.99
	TestData-7	0.957	0.941	0.957	0.99
	TestData-8	0.959	0.94	0.955	0.989
	TestData-9	0.958	0.943	0.963	0.991
	TestData-10	0.959	0.947	0.959	0.989
Recall	**Samples**	**LR+SVM+BOOST**	**LR+KNN+BOOST**	**RF+KNN+BOOST**	**ProposedEnsemble**
	TestData-1	0.952	0.942	0.956	0.991
	TestData-2	0.958	0.949	0.952	0.99
	TestData-3	0.951	0.948	0.957	0.99
	TestData-4	0.956	0.942	0.956	0.99
	TestData-5	0.952	0.945	0.961	0.99
	TestData-6	0.951	0.942	0.965	0.991
	TestData-7	0.959	0.945	0.964	0.99
	TestData-8	0.958	0.95	0.961	0.99
	TestData-9	0.951	0.942	0.962	0.99
	TestData-10	0.956	0.943	0.957	0.99
Precision	**Samples**	**LR+SVM+BOOST**	**LR+KNN+BOOST**	**RF+KNN+BOOST**	**ProposedEnsemble**
	TestData-1	0.956	0.949	0.953	0.991
	TestData-2	0.959	0.948	0.954	0.99
	TestData-3	0.959	0.948	0.966	0.99
	TestData-4	0.952	0.94	0.958	0.989
	TestData-5	0.954	0.947	0.958	0.99
	TestData-6	0.958	0.947	0.963	0.99
	TestData-7	0.959	0.947	0.958	0.99
	TestData-8	0.951	0.945	0.953	0.99
	TestData-9	0.954	0.941	0.962	0.99
	TestData-10	0.955	0.949	0.95	0.99
F-measure	**Samples**	**LR+SVM+BOOST**	**LR+KNN+BOOST**	**RF+KNN+BOOST**	**ProposedEnsemble**
	TestData-1	0.954	0.945	0.956	0.99
	TestData-2	0.956	0.942	0.954	0.99
	TestData-3	0.959	0.949	0.952	0.99
	TestData-4	0.954	0.949	0.955	0.991
	TestData-5	0.959	0.946	0.961	0.991
	TestData-6	0.952	0.949	0.959	0.991
	TestData-7	0.955	0.944	0.96	0.991
	TestData-8	0.959	0.946	0.951	0.99
	TestData-9	0.952	0.95	0.956	0.99
	TestData-10	0.952	0.949	0.952	0.99
AUC	**Samples**	**LR+SVM+BOOST**	**LR+KNN+BOOST**	**RF+KNN+BOOST**	**ProposedEnsemble**
	TestData-1	0.959	0.943	0.963	0.991
	TestData-2	0.954	0.941	0.953	0.99
	TestData-3	0.958	0.946	0.965	0.989
	TestData-4	0.952	0.946	0.952	0.991
	TestData-5	0.958	0.949	0.955	0.989
	TestData-6	0.955	0.947	0.956	0.99
	TestDat	0.954	0.95	0.956	0.989

	a-7				
	TestData-8	0.952	0.945	0.965	0.989
	TestData-9	0.956	0.941	0.954	0.991
	TestData-10	0.951	0.941	0.957	0.99

Table I, represents the result analysis of different machine learning models for ECG classification, evaluated on ten different datasets (TestData-1 to TestData-10). The performance metrics evaluated include Accuracy, Recall, Precision, F-measure, and AUC, and the models compared include LR+SVM+BOOST, LR+KNN+BOOST, RF+KNN+BOOST, and the proposed ensemble model.

Overall, the proposed ensemble model outperformed the other models on most datasets, achieving high scores on all performance metrics. LR+KNN+BOOST and RF+KNN+BOOST also performed well, with high accuracy and AUC scores, but lower precision and recall scores compared to the proposed ensemble model. The results suggest that ensemble models combining multiple machine learning algorithms can improve the accuracy and reliability of ECG classification, and may have potential for use in clinical settings. However, it is important to note that the evaluation was performed on a limited set of datasets, and further research is needed to evaluate the performance and generalizability of these models on larger and more diverse datasets.

V. Conclusion

The proposed real-time ECG CTG based ensemble feature extraction and unsupervised learning based classification framework for multi-class abnormality prediction in ECG signals shows promising results in accurately identifying different types of abnormalities in ECG signals. The use of ensemble feature extraction and unsupervised learning allows for robust and accurate classification of ECG signals, even in the presence of noise and variability. Additionally, the real-time aspect of the framework allows for real-time monitoring and early detection of abnormalities in ECG signals, which can greatly improve patient outcomes. Further research and validation of the proposed framework is needed to fully assess its clinical utility and potential for implementation in real-world settings. In future work, a novel parallel deep learning framework is used to improve the computational time on large big data.

References

[1] W. J. Groh et al., “2022 HRS expert consensus statement on evaluation and management of arrhythmic risk in neuromuscular disorders,” Heart Rhythm, vol. 19, no. 10, pp. e61–e120, Oct. 2022, doi: 10.1016/j.hrthm.2022.04.022.

[2] H. Liang and Y. Lu, “A CNN-RNN unified framework for intrapartum cardiotocograph classification,” Computer Methods and Programs in Biomedicine, vol. 229, p. 107300, Feb. 2023, doi: 10.1016/j.cmpb.2022.107300.

[3] S. F. Farrell et al., “A Shared Genetic Signature for Common Chronic Pain Conditions and its Impact on Biopsychosocial Traits,” The Journal of Pain, Oct. 2022, doi: 10.1016/j.jpain.2022.10.005.

[4] A. Jaba Deva Krupa, S. Dhanalakshmi, K. W. Lai, Y. Tan, and X. Wu, “An IoMT enabled deep learning framework for automatic detection of fetal QRS: A solution to remote prenatal care,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 7200–7211, Oct. 2022, doi: 10.1016/j.jksuci.2022.07.002.

[5] H. Allahem and S. Sampalli, “Automated labour detection framework to monitor pregnant women with a high risk of premature labour using machine learning and deep learning,” Informatics in Medicine Unlocked, vol. 28, p. 100771, Jan. 2022, doi: 10.1016/j.imu.2021.100771.

[6] M. Farahi et al., “Beat-to-beat fetal heart rate analysis using portable medical device and wavelet transformation technique,” Heliyon, vol. 8, no. 12, p. e12655, Dec. 2022, doi: 10.1016/j.heliyon.2022.e12655.

[7] L. Sterckx et al., “Clinical information extraction for preterm birth risk prediction,” Journal of Biomedical Informatics, vol. 110, p. 103544, Oct. 2020, doi: 10.1016/j.jbi.2020.103544.

[8] X. Zhang, W. Lu, Y. Pan, H. Wu, R. Wang, and R. Yu, “Empirical study on tangent loss function for classification with deep neural networks,” Computers & Electrical Engineering, vol. 90, p. 107000, Mar. 2021, doi: 10.1016/j.compeleceng.2021.107000.

[9] S. Magesh and P. S. Rajakumar, “Ensemble feature extraction-based prediction of fetal arrhythmia using cardiotocographic signals,” Measurement: Sensors, vol. 25, p. 100631, Feb. 2023, doi: 10.1016/j.measen.2022.100631.

[10] Y. Lu, X. Zhang, L. Jing, X. Li, and X. Fu, “Estimation of the foetal heart rate baseline based on singular spectrum analysis and empirical mode decomposition,” Future Generation Computer Systems, vol. 112, pp. 126–135, Nov. 2020, doi: 10.1016/j.future.2020.05.008.

[11] H. P. van Geijn, A. M. A. Lachmeijer, and F. J. A. Copray, “European multicentre studies in the field of Obstetrics,” European Journal of Obstetrics & Gynecology and Reproductive Biology, vol. 50, no. 1, pp. 5–23, Jun. 1993, doi: 10.1016/0028-2243(93)90159-A.

[12] J. Brablik, R. Kahankova, and R. Martinek, “Influence of System Configuration on the Quality of Non-Invasive Fetal Electrocardiography Measurement,” IFAC-PapersOnLine, vol. 52, no. 27, pp. 421–426, Jan. 2019, doi: 10.1016/j.ifacol.2019.12.700.

[13] D. J. Jagannath and A. I. Selvakumar, “Issues and research on foetal electrocardiogram signal elicitation,” Biomedical Signal Processing and Control, vol. 10, pp. 224–244, Mar. 2014, doi: 10.1016/j.bspc.2013.11.001.

[14] M. G. Frasch et al., “Non-invasive biomarkers of fetal brain development reflecting prenatal stress: An integrative multi-scale multi-species perspective on data collection and analysis,” Neuroscience & Biobehavioral Reviews, vol. 117, pp. 165–183, Oct. 2020, doi: 10.1016/j.neubiorev.2018.05.026.

[15] V. P. C. Magboo and Ma. S. A. Magboo, “Prediction of Late Intrauterine Growth Restriction using Machine Learning Models,” Procedia Computer Science, vol. 207, pp. 1427–1436, Jan. 2022, doi: 10.1016/j.procs.2022.09.199.

[16] C. Xu, J. Howey, P. Ohorodnyk, M. Roth, H. Zhang, and S. Li, “Segmentation and quantification of infarction without contrast agents via spatiotemporal generative adversarial learning,” Medical Image Analysis, vol. 59, p. 101568, Jan. 2020, doi: 10.1016/j.media.2019.101568.

[17] A. M. Oprescu et al., “Towards a data collection methodology for Responsible Artificial Intelligence in health: A prospective and qualitative study in pregnancy,” Information Fusion, vol. 83–84, pp. 53–78, Jul. 2022, doi: 10.1016/j.inffus.2022.03.011.

[18] P. Chetlur Adithya, R. Sankar, W. A. Moreno, and S. Hart, “Trends in fetal monitoring through phonocardiography: Challenges and future directions,” Biomedical Signal Processing and Control, vol. 33, pp. 289–305, Mar. 2017, doi: 10.1016/j.bspc.2016.11.007.

[19] J. Spilka et al., “Using nonlinear features for fetal heart rate classification,” Biomedical Signal Processing and Control, vol. 7, no. 4, pp. 350–357, Jul. 2012, doi: 10.1016/j.bspc.2011.06.008.

Bird Image Classification using Convolutional Neural Network Transfer Learning Architectures

Asmita Manna[1], Nilam Upasani[2], Shubham Jadhav[3], Ruturaj Mane[4], Rutuja Chaudhari[5], Vishal Chatre[6]
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India[1, 3, 4, 5, 6]
Balaji Institute of Technology and Management, Sri Balaji University, Pune, India[2]

***Abstract*—With the technological progress of human beings, more and more animal and bird species are being endangered and sometimes even going to the verge of extinction. However, the existence of birds is highly beneficial for human civilization as birds help in pollination, destroying harmful insects for crops, etc. To ensure the healthy co-existence of all species along with human beings, almost all advanced countries have taken up some conservation measures for endangered species. To ensure conservation, the first step is to identify the species of birds found in different locations. Deep learning-based techniques are best suited for the automated identification of bird species from the captured images. In this paper, a Convolutional Neural Network based bird image identification methodology has been proposed. Four different transfer learning-based architectures, namely Resnet152V2, Inception V3, Densenet201, and MobileNetV2 have been used for bird image classification and identification. The models have been trained using 58388 images belonging to 400 species of birds, and the models have been tested using 2000 images belonging to 400 species of birds. Out of these four models, Resnet152V2 and DenseNet201 performed comparatively well. The accuracy of Resnet152V2 was highest at 95.45%, but it faced a large loss of 0.8835. But based on the results, even though DenseNet201 had an accuracy of 95.05%, it faced less loss i.e., of 0.6854. The results show that the DenseNet201 model can further be used for real-life bird image classification.**

***Keywords*—Deep learning; CNN; Image classification; DenseNet201; ResNet152V2; InceptionV3; MobileNetV2**

I. Introduction

Biodiversity around us is quite important for human civilization as it helps maintain a balance in the ecosystems. Birds are one of the most important resources, which help us in various ways like pollination, protecting crops from harmful pests destroying the crops, etc. However, with rapid industrialization, more and more bird species are becoming endangered and sometimes on the verge of extinction. Bringing more forestland and wetland under modernization is also fueling the extinction process. Therefore, almost all countries have taken one or more conservation projects to preserve biodiversity and conserve endangered species [1][2].

To preserve a species, the first important step is to identify the species correctly followed by the appropriate steps to preserve it. Deep learning-based approaches are best suited for the automated identification of bird species because deep learning enables us to extract the features of birds and provide higher accuracy while predicting. Deep learning can identify the species of a bird based on data in the form of audio, video, and image. In the case of real life, audio and video are not that suitable as the overlapping of chirping of multiple birds can't be ruled out, and those noises may tamper with the essential features of the bird. Thus, in this paper, a deep learning-based approach trained with bird images has been proposed. There are some papers which proposed the use of images for bird species identification [9][20][21]. But the novelty of our study is that we have included 400 species for identification which has expanded the application of our study.

As there are hundreds of deep learning algorithms available, it is necessary to investigate with the test data set to understand which algorithm generates the best accuracy and minimizes the loss. In this paper, four different transfer learning-based models namely Resnet152v2, InceptionV3, Densenet201, and MobileNetV2 are tested and compared for bird image classification. The transfer learning models are used to extract the features of the birds. The dataset used contains a total of 58388 images of birds. The total number of species included in the dataset is 400 [31]. The output and input layer are added to the transfer learning model with dense connection. The output layer is generating results by using SoftMax. Given that SoftMax is employed for multiclass classification, it serves as the activation function.

The major research objectives of this paper are as follows: (a) To compare various transfer learning models for bird species identification from bird image datasets. (b) To build a system of automatic identification of bird species with maximum accuracy using the transfer learning model which is most suited for the given dataset. (c) To put forth a web-based system which will help photographers to build their portfolio by identifying the bird species from the image they captured.

A. *Contribution of the Paper*

- A comprehensive study of existing deep learning/ transfer learning-based systems for bird species identification and pinpointing the drawbacks of those existing systems.
- Identification of an intricate dataset for bird species identification by studying various related datasets.
- Implementation of different transfer leaning models such as InceptionV3, ResNet152V2, DenseNet201 and MobileNetV2 on the identified dataset.
- Selection of the most efficient and accurate model suited for image classification of birds and the measures that can be incorporated to increase the performance.

Section II of this paper presents the literature review to find the gap in the existing approaches; Section III presents the proposed architecture and methodology; Section IV shows the experimental results and, finally Section V concludes the paper.

II. Literature Review

Many researchers have used Deep Learning and Convolutional Neural Network based approaches for image classifications, especially for identifying diseases from image datasets, but not much work is done for identifying bird species from the image datasets of birds. In this section, the existing approaches of image classification using transfer learning techniques are explored.

A recent paper [3] explaining the deep learning architecture for the detection of birds from the images captured by the webcam was published. For the pre-processing of data, deep learning algorithm's capability for the detection of birds inside the images is checked first. Then the authors used two CNN models i.e., single-shot detectors (SSD) and Quicker R-CNN as a combination with Inception ResNet152V2, ResNet50, ResNet152, ResNet101, and MobileNet-V2 features. By combining a faster R-CNN, they got high precision, and the SSD with MobileNetV2 was selected as the best model in terms of speed and smaller memory consumption.

Transfer learning model was used with six different CNN architectures such as DenseNet201, MobileNetV2, ResNet50, InceptionResNetV2, ResNet152V2, and Exception in a paper [4]. After calculating the evaluation matrix, the authors found that MobileNetV2 performed better than the other transfer learning approaches in terms of evaluation matrices. However, they could classify the bird images in only eight categories.

A paper was published in which [5] the authors worked to find a suitable model for transfer learning. After comparison of results, the authors showed that the training Inception-v3 on the CIFAR-10 dataset provided better results. The authors also explained that the basic to advanced use of transfer learning may be used not only for the model presented but also for other deep neural networks for image classification.

A paper [6] implementing the CNN algorithm to extract information from bird images was published. The CNN model was developed entirely from scratch. The model was then trained to test its efficiency. The developed application had a high accuracy of about 93.19% on the training set and 84.9% on the testing set.

The image classification models MobileNetV2 and Inception-v3 were proposed to be used in a paper [7]. The authors used four approaches—Inception-v3 with and without transfer learning and MobileNetV2 with and without transfer learning—to accomplish the task. Among the four approaches, MobileNetV2 with transfer learning performed better, with an accuracy of about 91.00%.

In a paper published in 2021[8], the authors tried to identify the habitat elements from the bird elements using a deep convolutional neural network and ResNet152 dependent models, which gave the test accuracy that was best. It has been proven that a deep convolutional neural network could be effective for automatically identifying habitat elements from images of birds. The author concluded that the actual implementation of this technology would be extremely useful in understanding the relationship between habitat elements and birds.

The paper [9] proposed a VGG-16 network model-based solution to extract bird image features. The authors used different classification methods, each with different results. Support Vector Machine (SVM), when compared to other categorization techniques like random forest and K-Nearest Neighbor (KNN), provided the highest accuracy of 89%.

The authors of a paper which was published in 2019 [10] developed a cloud-based mobile phone app that makes use of deep learning for image processing to find the species of birds from the various digital images that are transferred by the user over a mobile phone. Convolutional Neural Network was trained using bird images to limit the outstanding image features. The Convolutional Neural Network model with bound connections gave high accuracy, which is 99.00%, as compared to the CNN with 93.98% and the SVM with 89.00% for the image training.

A pose-normalized deep convolutional nets approach was proposed in a paper published [11]. The proposed method depends on a detection part, and the Convolutional Neural Network features are extracted from various regions that are pose-normalized. Its execution was better for the usage of Convolutional Neural Network features, which were fine-tuned upon CUB-200-2011 for every region. It also was effective for the usage of various Convolutional Neural Network layers for numerous alignment levels; and the usage of a similar warping function that's estimated to use a larger number for detecting key points. They introduced an innovative method for studying a group of various pose-regions that clearly curtails the alignment of pixel error. Then it works on complicated pose-warping operations.

Another paper [12] proposed a deep learning model which can identify individual birds from the input image. The authors proposed two different models and showed that the proposed pre-trained ResNet model has achieved better accuracy than the based model. The best model showed 97.98% accuracy in identifying bird species.

The paper proposed in 2020 [13] used deep learning models to detect pneumonia based on chest x-ray images. The author used four models: two pretrained models (ResNet152V2 and MobileNetV2), CNN, and long-short-term memory (LSTM). The result showed that the ResNet152V2 performed better and the MobileNetV2, CNN, LSTM accuracy, recall, F1-score, precision, and AUC were higher than 91%.

An author published a paper based on declining the North American avifauna [14] represented that by using the multiple and independent monitoring networks author represented the loss of population of the North American avifauna which includes the common species.

The paper proposed in 2021 describes the declining of the forest bird species [15] where the author studied birds in six land types in the oak forest biome of Himalaya. The richness of the species was west in pine and built-up sites as compared to natural oak. The forest specialist and insectivores are reduced up to 60 - 80% in the modified forests.

The loss found in biodiversity in the European Union is described in a paper [16] by the author by studying the extensive dataset. The author evaluated that there is a decline of around 17-19 % in the overall bird abundance. The authors are suggesting that we should preserve the bird species and that it is beneficial to nature and human beings.

In the paper that was published in 2021 [17], the authors tried to explain that there is a huge amount of decrease in the bird species, some have vanished, some are endangered. As a result, it has a negative impact on both biodiversity and human lives. So, it's our responsibility to preserve them.

A paper that uses a deep convolutional neural network [18] to identify birds' images was published. The author used habitat elements of bird images. Author used the model based on the ResNet152 algorithm, given 95.52% validation and AlexNet given the lowest test accuracy as 89.48%. It is proved that a deep convolutional neural network is efficient and useful for bird image classification.

A paper was published which explained that ecological resources are important for the survival of human beings [19]. The author has used ecological research. The author first developed the relationship between the theory of deep learning and ecological research. It is expected that participation and preparing cross-disciplinary abilities may advance standardization. Deep learning is used for nonlinear feature extraction for scientific and industrial data processing.

A paper that was published in 2019 [20], used the VGG 16 network to extract features of birds. The author used a dataset of bird species of Bangladesh. The author used different classification methods, like random forest, K-nearest neighbor (KNN) but the support vector machine (SVM) gave the max accuracy of 89%.

A paper that uses a deep learning platform to identify bird species images [21] using the mobile app Internet of birds. The author used convolutional neural network (CNN) to find different features in images. To improve feature extraction, the skip connection method is used. Then the SoftMax function is used to get a probability distribution of the features of birds. 93.98 % was convolutional neural network (CNN) accuracy and support vector machine (SVM) got 89.00% accuracy. Both accuracies are less than 99.00%, which is the highest accuracy of the proposed model convolutional neural network (CNN) with skip connection.

The paper published in 2014[22] proposed architecture which first finds the pose of a bird. For bird feature extraction Deep Convolutional Neural network is used. To find the compact pose the author proposed a novel graph- based clustering algorithm. Author got great classification accuracy that is 75% vs. 55-65%.

A paper that was published in 2020 [23] proposed a deep learning model to identify bird species. The author used the ResNet model as a pre-trained convolutional neural network (CNN) with a base model to identify the images. The author got a high accuracy of 97.98% on the bird classifications.

A paper [24] for predicting breast cancer was published in 2021 that used a dataset of QIN-Brest for breast cancer detection that is divided into the ratio 7: 3 for training and testing. The authors used two deep transfer learning models, DenseNet201, ResNet152v2 and an ensemble model with concatenation of two models, trained and tested using a dataset of CT images. The ensemble model has been given 100% accuracy on the test data. The authors concluded that ensemble models are better at predicting breast cancer than those of DenseNet201 and ResNet152v2 models.

A paper published in 2019 [25] explaining that there is a loss in abundance of bird species that leads to the changes in the ecosystem. Authors studied the bird population of North American avifauna for over 48 years. It is found that there is a loss in the population of birds, around 29% of 1970 bird population. This population loss needs to be addressed for the future of biodiversity.

In the paper that was published in 2021 [26], it is said that because of the land use change, there is a loss of biodiversity in different countries. That is responsible for changing the forest ecosystem. It is very important to know these things to avoid biodiversity loss. The author carried out a semantic breeding-season survey in six different land use types. The author's study shows that there is moderate to drastic species loss in all-the changed land uses compared to natural oak forests.

The Global Assessment Report on Biodiversity and Ecosystem Services is a thorough and evidence-based analysis of the state of biodiversity worldwide [27]. It pinpoints the causes of loss and the impacts on food security, health, and livelihoods. With recommendations for policy action, it provides a comprehensive understanding of the risks and opportunities associated with ecosystem degradation. Despite its strengths, future work is needed to address data gaps, regional limitations, implementation challenges, and the integration of other global challenges, requiring ongoing monitoring and evaluation.

The paper ‘Abundance decline in the avifauna of the European Union’ [28] reveals cross-continental similarities in biodiversity change" analyzes the changes in bird abundance in the European Union (EU) between 1980 and 2015. The study found that bird abundance has decreased by 22.5% in the EU during this period, and that the decline was most pronounced in farmland and grassland bird species. The decline in bird populations was found to be similar to trends observed in North America, suggesting cross-continental similarities in biodiversity change. The paper highlights the need for conservation efforts to reverse the decline in bird populations and prevent further biodiversity loss.

In a paper named ‘Birds in Decline’ [29], Youth H. takes us on a soaring journey through the troubling topic of declining bird populations. With a keen eye for detail and a passion for avian conservation, Youth H. paints a vivid picture of the challenges faced by birds and the ecosystems they inhabit. Through a mix of data analysis and personal observations, the author sheds light on the alarming trend of bird decline and highlights the urgent need for action. The result is a thought-provoking and informative piece that will leave readers with a renewed appreciation for the feathered friends that share our planet.

The paper ‘Applications for deep learning in ecology’ [30] provides an overview of the use of deep learning in ecology. It explains how deep learning, a branch of machine learning, has become popular due to its flexibility and performance. The paper reviews existing implementations and demonstrates how deep learning has been used successfully to identify species, classify animal behavior, and estimate biodiversity in large datasets like camera-trap images, audio recordings, and videos. It also provides guidelines, recommendations, and useful resources to help ecologists get started with deep learning. The authors argue that deep learning can become a powerful reference tool for ecologists, especially at a time when automatic monitoring of populations and ecosystems generates vast amounts of data that cannot be effectively processed by humans anymore.

The papers studied during the survey included bird image classification approaches, but the results were provided for only a few numbers of bird species. The dataset used for comparison of the transfer learning models in our study included 400 species of birds and the size was also large enough which ensured proper training and testing of the models to provide the maximum output from the models. A detailed comparative study of the discussed papers is presented in Table I.

TABLE I. COMPARATIVE STUDY OF EXISTING APPROACHES

Paper id	Dataset	Approach	Advantages	Limitations / Future work
[3]	Live bird feed watchers came from Cornell Lab of Ornithology at 6 different locations.	Single-shot detector and faster R-CNN with ResNet50, MobileNet-V2, ResNet152, ResNet101, and Inception ResNet-V2 feature extractors.	Faster R-CNN combined with ResNet152 feature extractor was best for achieving high precision.	The SSD with MobileNetV2 was best for fast speed and low memory consumption but lacked high precision like Faster R-CNN with ResNet152.
[4]	500 images of each bird class of 7 species of Bangladesh birds with a total of 2800 training images and 700 testing images	Transfer learning used in 6 different CNN architectures DenseNet201, InceptionResNetV2, MobileNetV2, ResNet50, ResNet152V2, and Xception.	MobileNetV2 outperforms all the other transfer learning models.	There is no support for using the live bird feed of the cam for identification.
[5]	CIFAR-10 and Caltech Faces dataset	Inception- v3 via transfer learning.	Transfer learning models are better than custom CNN.	Accuracy can be improved by increasing the epoch sizes and size of the dataset and combining CNN with Long-short Term Memory (LSTM).
[6]	Custom dataset using Microsoft’s Bing Image Search API v7.	Used a deep learning model to extract the information images of birds using CNN.	Developed a smaller and more portable version of VGGNet and achieved an accuracy of 93.19% on the training dataset.	The testing dataset accuracy is 84.91% which is very low.
[7]	700 testing images and 2800 training images of 7 local birds in Bangladesh.	Used 4 approaches inception - v3 with and without transfer learning, MobileNet with and without transfer learning.	MobileNet performed better than the Inception model.	Only 7 bird species are used to generate the output.
[8]	Custom dataset obtained from BirdFans in China containing 20000 images.	Implemented proposed method with 4 types of deep CNN.	DCNN gave a high accuracy of 97.76% on the testing dataset.	Further research is needed to improve the DCNN for performing well in real-world scenarios.
[9]	1600 images of 27 species of birds of Bangladesh.	Used machine learning approach using VGG-16 network as a model.	SVM gave the maximum accuracy of 89%.	KNN has less accuracy than SVM.
[10]	27 Taiwan endemic bird species.	Given bird images as an input to CNN which is used with the skip connection.	The Convolutional Neural Network model with bound connection gives high accuracy which is 99.00 %.	CNN and SVM have less accuracy than CNN with the skip connection.
[11]	CUB-200-2011	The bird's features are computed by CNN.	Performance was better for 1] usage of Convolutional Neural Network features which were fine-tuned upon CUB-200-2011 for every region, 2] usage of various Convolutional Neural Network layers for numerous alignment levels type, and 3] usage of a similar warping function that's estimated for using a larger number for detecting key points.	For future work, the proposed solution can be applied to other fine-grained datasets. Then they explore the custom-built CNN network structures and their training.
[12]	Data of bird species from various sources merged with the western dataset.	Pretrained CNN performs better on input images.	The proposed pre-trained ResNet model has better accuracy of 97.98% in identifying bird species.	The based model showed less accuracy than the proposed pre-trained ResNet model.

[13]	QIN-Breast Dataset	Two deep transfer learning models, ResNet152V2 and DenseNet201, and an ensemble model with a combination of 2 models, trained and tested by CT images.	The proposed model gained the maximum accuracy of 100 percent on the dataset which was tested and also high performance of 100% in f1-score, recall and precision value.	Both the models still have scope of improvement as considering various parameters.
[14]	529 bird species from the US and Canada.	Multiple and independent monitoring networks (radar networks)	Networks used in the paper are useful for getting the population loss across the North American avifauna.	As the birds are giving numerous benefits to the ecosystem, people should conserve them.
[15]	8549 bird observations.	The landscape is chosen and further it is divided into parts as per their use.	There should be conservation planning of bird species in human dominated landscapes.	Due to the use of land by humans affected the forest bird species in western Himalaya.
[16]	445 native bird species.	Bayesian models are used to check the change in abundance of the birds in the EU. The imputed model shrinks the uncertain indices of the species towards the group mean.	Using the models used in a paper, the author represented the biodiversity loss in native avifauna.	There is a decline of 17-19% in the overall breeding bird abundance
[17]	Worldwide different bird species.	Different bird species were studied all over the world, and various vanished and endangered species were discovered.	Preserve the bird species.	There is a huge amount of decrease in the bird species, and it has a negative impact on both biodiversity and human lives.
[18]	Habitat Elements from Bird Images	Used habitat elements of bird image and model based on the ResNet - 152 algorithm, given 95.52% validation and AlexNet given the lowest test accuracy as 89.48%	ResNet basel model given highest validation accuracy is 95.52 %.	ResNet basel model given lowest test accuracy is 95.52 %.
[19]	Multi-source data	Used ecological resources that are important for the survival of human beings. Developed the relationship between deep learning and ecological research .Deep learning is used for nonlinear feature extraction	Understanding of ecological resources is beneficial for the sustainable development of human society.	-
[20]	Bird species of Bangladesh.	Used the VGG 16 network to extract features of birds. Different classification methods, like random forest, K-nearest neighbor(KNN) but the support vector machine(SVM) gave the max accuracy of 89%	Support Vector Machine (SVM) gave the max accuracy of 89%	Improve accuracy by increasing data
[21]	27 bird species endemic to Taiwan.	Mobile app is used to identify bird species images. Used CNN for features extraction. To improve feature extraction, the skip connection method is used.	99.00%, is the highest accuracy of the convolutional neural network (CNN)model with skip connection.	CNN has 93.98 %,SVM has 89.00% accuracy which is lesser than CNN with skip connection model.
[22]	Worldwide different bird species.	Finds the pose of a bird. Deep convolutional network is used for feature extraction. For compact poses a novel graph-based clustering algorithm is used.	Classification accuracy, is 75% vs 55-65% which is of old method.	-
[23]	Birds found in diverse scenarios worldwide.	Used resNet model as a pre-trained CNN network with a base model to identify the images.	High accuracy of 97.98%	Birds are found in diverse scenario with different size and shapes.
[24]	Public dataset QIN Breast	Used 2 deep transfer learning models, DenseNet201,ResNet152v2 and an ensemble model with concatenation of 2 models, trained and tested using a dataset of CT images.	Ensemble model has given 100% accuracy on the test data.	Ensemble models are better than DenseNet201 and ResNet152v2 models.
[25]	North American avifauna	Loss in abundance of bird species that leads to the changes in the ecosystem	Population loss is studied which is helpful for the future of biodiversity	American avifauna when studied it is found that a loss in the population of birds, around 29% of 1970 bird population.

[26]	Western Himalaya bird species.	Because of the land use change, there is a loss of biodiversity.	Studied land use change which will help in future to avoid biodiversity loss.	Because of the land use change there is decrease in the biodiversity.
[27]	-	It provides a comprehensive analysis that identifies the causes and impacts of biodiversity loss.	It provides policymakers with actionable recommendations, Integrated approach, Global perspective, Foundation for future research.	Areas for future work, such as data gaps, Limited regional focus and implementation challenges.
[28]	Extensive dataset of breeding bird abundance in the European Union.	The approach used in the paper was to analyze data from bird monitoring programs across the European Union to estimate population changes over time.	The authors estimated a decline of 17-19% in overall breeding bird abundance, equivalent to a loss of 560-620 million individual birds. The study highlights the high declines in bird numbers among species associated with agricultural land.	Limitations of the paper are that it did not include other aspects of biodiversity, only included bird populations in the European Union and did not consider populations in other regions, did not investigate the specific causes of bird population decline.
[29]	-	The paper examines senescence variation in multicellular organisms, particularly social species. It reviews senescence research, quantifies covariation between mortality and reproductive ageing, models social interaction effects on ageing, tests predictions in social species, and examines senescence in a cooperative breeder population.	This work can inform future research on the evolution of senescence and the dynamics of social species.	Future work for this paper could involve further testing and refinement of the social interaction model for senescence in social species.
[30]	-	The article discusses the potential applications of deep learning in ecology, including species identification, animal behavior classification, and biodiversity estimation. The authors provide guidelines, recommendations, and a reference flowchart for ecologists to get started with deep learning. They argue that as ecological datasets become larger and more complex, deep learning could become a powerful tool for ecologists.	It provides an overview of deep learning in ecology, demonstrates the usefulness of deep learning in ecology and highlights the potential of deep learning for automatic monitoring.	The paper does not provide an in-depth discussion of the limitations and potential biases of using deep learning in ecology.

III. Proposed Work

The principal aim is to implement various transfer learning models to compare their performance on the bird image classification dataset. The main transfer learning models used for comparison are InceptionV3, ResNet152V2, Densenet201, and MobilenetV2. The transfer learning models are used as feature extractors. The dataset used contains a total of 58388 images of birds. The total number of species included in the dataset is 400. The output and input layer are added to the transfer learning model with dense connection. The output layer is generating results by using SoftMax as the activation function. Given that SoftMax is employed for multiclass classification, it serves as the activation function. The proposed system architecture is depicted in Fig. 1 and described in detail thereafter.

A. Datasets

To complete the implementation of the comparison of these transfer learning models a dataset of bird images available on Kaggle [31] is used for both the training and testing purposes. There are a total of 58388 images belonging to 400 species of birds for training purposes and 2000 images each of testing and validation images belonging to 400 species of birds.

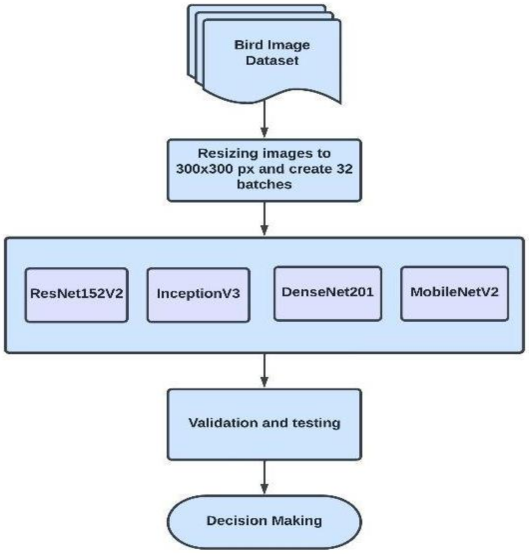


Fig. 1. System architecture.

B. Data Pre-processing

This mainly deals with preparing the data so that we can get maximum performance from the model after training the model using this data. All the images in the dataset must be in the same format for the model to be easier to train. For image pre-processing, we converted the images to 300x300px for standardizing the image dimensions. Then we created batches for training the model in batches. A batch size of 32 images was created. Batch size means the number of samples processed before the model is updated. This is needed because after training from batches expected results can be matched with actual results to calculate the error and this error is then used to update the algorithm to improve the model. All the images used are RGB images. So, the dimension of images used is (300,300,3).

C. Transfer Learning Models

Based on the literature survey, four transfer learning models are selected. These are ResNet152V2, InceptionV3, DenseNet201 and MobileNetV2. The ResNet152V2 and InceptionV3 models are selected because these are widely used and perform well. DenseNet201 is densely connected and thus takes more time to train but generally provides higher accuracy; and the MobileNetV2 is lightweight and has faster processing output.

1) DenseNet201: This is the feature extraction model. It is a convolutional neural network that is 201 layers deep. We can load millions of images to this pre-trained network which can classify the images into multiple object categories. Strong gradient flow, computational efficiency, and multiple different features are the advantages of this model.

2) ResNet152V2: This model is used for feature extraction. It is a pretrained model, so it gives higher accuracy in less time than traditional CNN. As the number of layers increases the training and testing rate also increases. To solve the problem of vanishing gradient residual block concept is introduced. The skip connection technique is used in this network. In this technique, it skips some layers in between. The advantage of this skip connection is if any layer in between changes the performance of architecture then this layer is skipped so performance increases. The ResNet model consists of a reshape layer, flattened layer, a dense layer having 128 neurons, a dropout layer, and the dense layer with SoftMax function which is used to classify the image.

3) InceptionV3: It is the deep learning model used for image classification which is based on CNN. It gives greater than 78.1% accuracy on the image datasets. It uses the transfer learning approach, so it gives good performance in classification. The model is made up of symmetric and asymmetric blocks which include convolutions, average pooling, max pooling, dropouts, concatenations, and fully connected layers.

4) MobileNetV2: It is a lightweight model for feature extraction which gives good performance on mobile devices. It is the convolutional neural network which is 53 layers deep. The model is based on residual structure. The model contains initial convolution layers with 32 filters followed by 19 residual bottleneck layers. The model can work in more than millions of images and the pre-trained network can classify images into about 1000 object categories.

D. Training

Each model is trained separately in a different notebook on Google Colab. An input layer and an output layer were added to the model for giving input and output along with a global average 2D pooling layer before the output layer. The Global Average 2D pooling layer takes a block of a tensor as input and calculates an average value of all the values across the tensor for each of the input channels. Pooling is necessary to reduce the dimensions, so it makes it easy for the model to train. The input layer accepts images of dimensions (300,300,3) and the output layer has 400 nodes for 400 species. The network is densely connected.

All the layers are frozen in the transfer learning model. The layers are frozen to reduce the training time and the activation function used is the same for all the models i.e., SoftMax as it is a multiclass classification. The learning rate is kept at 0.01. Based on the training time required to train the models and the size of the dataset used, 15 epochs were finalized to train the models. All these parameters are kept the same. This ensures that all the transfer learning models are fairly evaluated under similar conditions to give unbiased results.

E. Classification

A testing dataset of 2000 images is provided for testing the trained model. Accuracy is obtained to evaluate the performance of the models using the testing dataset. For prediction, the image which is to be classified is obtained from the user and it is first processed. It is converted into an image of 300x300px that is best suited for the model and then classification is done.

IV. Results

The results are checked based on the accuracy provided by each model. The accuracy and loss curves are also plotted to better understand the training process of the model. This gives an idea about the accuracy and loss of the model over each epoch. The maximum observed accuracy is of ResNet152V2 model i.e., 95.45%. The plots of each model are given below.

As the models were built on the same parameters, the results obtained from the comparison can be considered to fairly evaluate the performance of these transfer learning models. The MobileNetV2 took less training time, and the model size is also small. The DenseNet201 has the best performance out of all and the InceptionV3 and ResNet152V2 performed well. ResNet152V2 displayed maximum accuracy but was having a large loss. These models can be compared based on various factors. The results obtained because of those factors are given below.

A. Performance Metrics

Given below are the performance metrics based on which the models are compared.

1) Accuracy: Accuracy in model training refers to the ability of the model to correctly classify input data. In other words, accuracy is a measure of how often the model predicts the correct label for a given input. It is calculated as the ratio of correctly predicted samples to the total number of samples in the dataset. A higher accuracy score indicates that the model is better at predicting the correct output and is more reliable. However, it is important to note that accuracy is not the only measure of a model's performance.

2) Loss: Loss in model training refers to the difference between the predicted output of the model and the actual output. In other words, loss is a measure of how far off the model's predictions are from the true values. The goal of training a model is to minimize the loss function, which is typically a mathematical function that measures the difference between the predicted output and the actual output. This is done by adjusting the weights and biases of the network during the training process using techniques such as backpropagation. Lower loss values indicate that the model is better at predicting the correct output and is more accurate. However, it is important to find a balance between low loss values and overfitting, where the model becomes too specialized to the training data and performs poorly on new, unseen data.

3) Training time: Training time in model training refers to the amount of time it takes for the model to learn from the training data and adjust its weights and biases to minimize the loss function. The training time can depend on a variety of factors, such as the complexity of the model, the size of the training dataset, the available computational resources, and the hyperparameters used for training. The training process involves iterating through the training dataset multiple times (epochs) and adjusting the weights and biases of the network based on the feedback from the loss function. The goal is to find the optimal values for the weights and biases that minimize the loss and improve the accuracy of the model. In general, larger, and more complex models with larger datasets may require longer training times, while smaller and simpler models may converge faster. Efficient use of parallel processing resources can also help to reduce the training time.

B. Discussion

Given below are the accuracy graphs of the models.

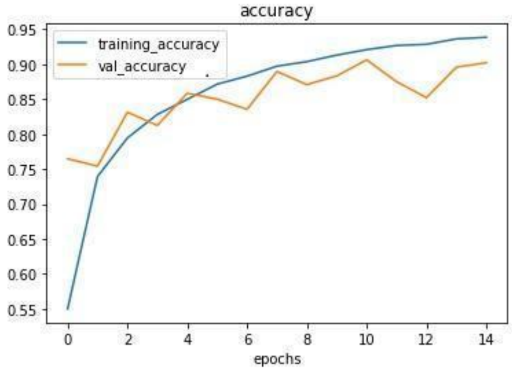


Fig. 2. InceptionV3 model training and validation accuracy.

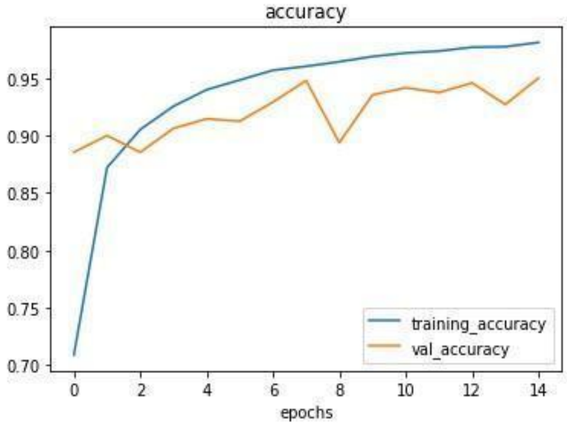


Fig. 3. ResNet152V2 model training and validation accuracy.

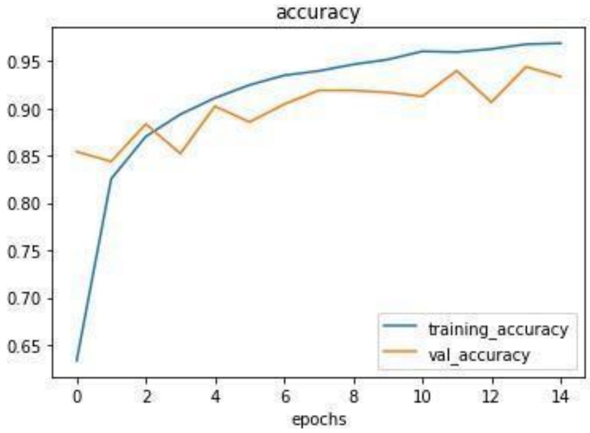


Fig. 4. DenseNet201 model training and validation accuracy.

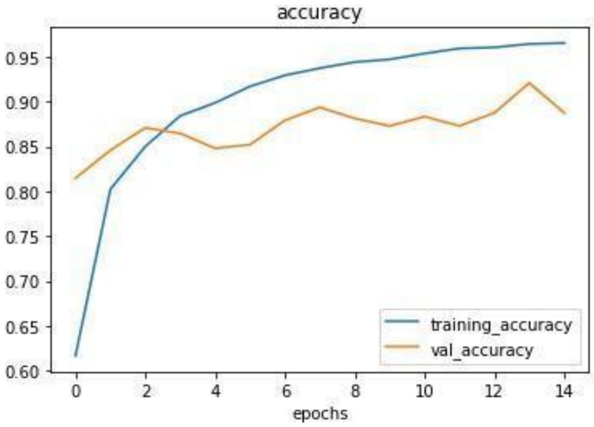


Fig. 5. MobileNetV2 model training and validation accuracy.

The steadiest training accuracy was observed in DenseNet201 from Fig. 4, and the other models performed almost similar in the training accuracy. The most stable validation accuracy was observed in DenseNet201 and the most unstable was observed in ResNet152V2 from Fig. 3 and InceptionV3 displayed in Fig. 2. From Fig. 5, it can be observed that the average training time of MobileNetV2 was 128 sec.

Given below are the loss graphs of the models.

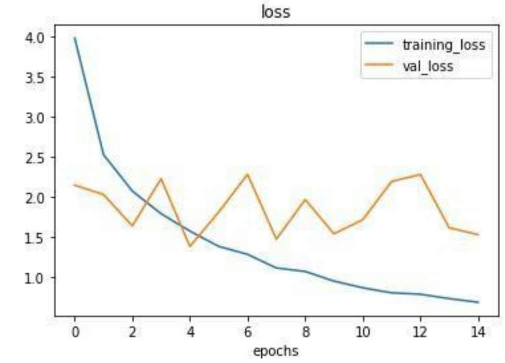


Fig. 6. InceptionV3 model training and validation loss.

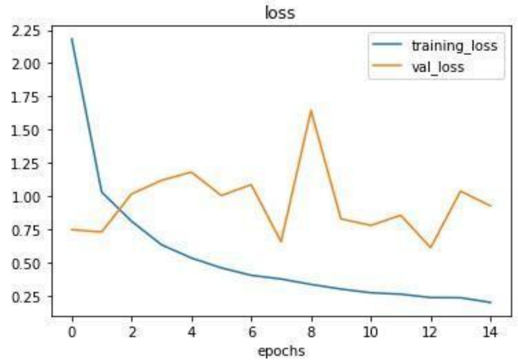


Fig. 7. ResNet152V2 model training and validation loss.

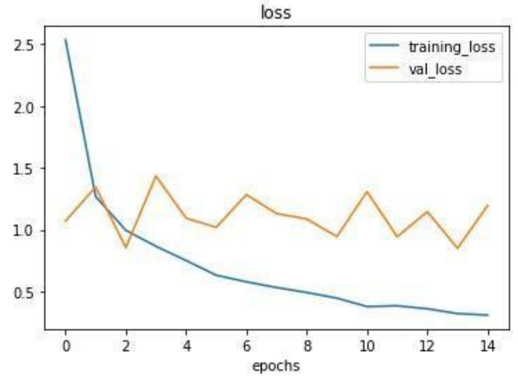


Fig. 8. DenseNet201 model training and validation loss.

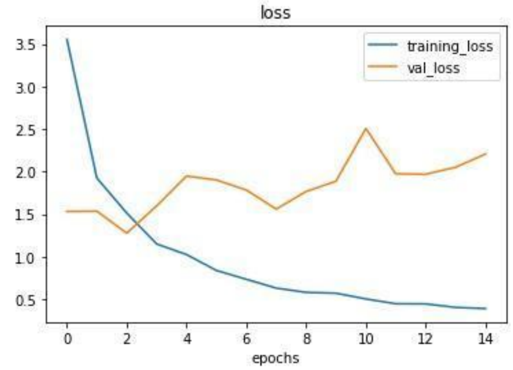


Fig. 9. MobileNetV2 model training and validation loss.

The models performed similarly on training data by reducing the training loss over epochs, but the loss fluctuated during the validation. The most fluctuation was observed in InceptionV3 validation loss from Fig. 6 and the most stable among the models was observed in DenseNet201 from Fig. 8. Fig. 9 depicted that the validation loss in MobileNetV2 increased over epochs instead of decreasing. The training and validation loss of Resnet152V2 can be observed from Fig. 7.

Fig. 10 displays the comparative graph of average training time required per epoch for training the models.

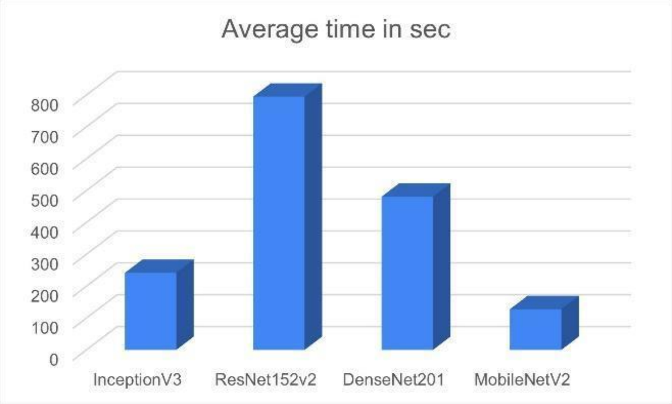


Fig. 10. Average training time per epoch comparison.

Fig. 10 compares the training time by taking the average of the time required by the model for each epoch. The ResNet152V2 required the highest training time with an average of 795 sec. It is more than the training time required by InceptionV3 and DenseNetV2 with average time 242 sec and 481 sec respectively. The MobileNetV2 is lighter and thus requires less time to train than the other transfer learning models compared in this study. The average training time of MobileNetV2 was 128 sec.

The overall result can be understood from the testing accuracy and the testing loss of the models as depicted in Table II.

Fig. 11 shows that ResNet152V2 provided the maximum accuracy with a loss of 0.883. But the DenseNet201 faced less loss than ResNet152V2 and provided accuracy like the ResNet152V2 model. Thus, DenseNet201 performed better than the other models on the bird image dataset with an accuracy of 95.09%.

TABLE II. COMPARISON OF PERFORMANCES OF TRANSFER LEARNING MODELS

Model Name	Accuracy	Loss
InceptionV3	90.2%	1.542
ResNet152v2	95.45%	0.883
DenseNet201	95.09%	0.685
MobileNetV2	91.1%	1.782

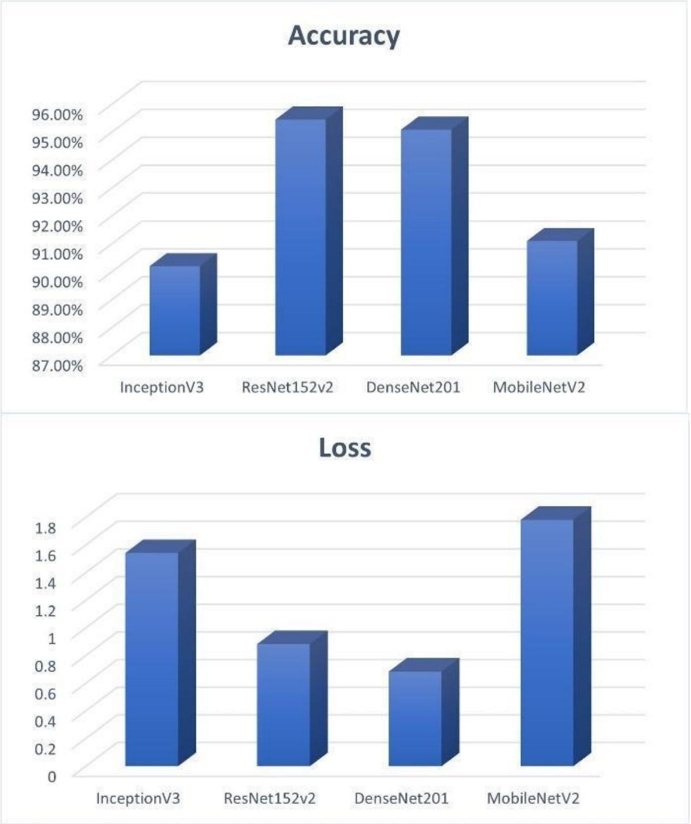


Fig. 11. Accuracy and loss curves for comparing the models on testing data.

V. Conclusion and Future Work

In this paper, various deep learning/ transfer learning models for bird species identification have been studied and compared. It has been understood that mostly the existing systems have used deep learning models or transfer learning models with smaller datasets. Therefore, an elaborate dataset consisting of 58388 [31] bird images of four hundred species has been identified for this work.

Four different transfer learning models namely InceptionV3, ResNet152V2, DenseNet201, and MobileNetV2 are implemented on the identified dataset. All these models were trained under similar conditions to get the best possible comparison between them. ResNet152V2 provided an accuracy of 95.45% which is more than the other 3 models. But it also faces more losses than DenseNet201. MobileNetV2 has the lowest training time, but the model accuracy is not as good as the other models. In conclusion, the best model among these models is DenseNet201. Even though its accuracy is a little less than the ResNet152V2 model, its loss is far less than other models. Thus, DenseNet201 is better than Resnet152V2.

Though the accuracy given by the implemented models are good, the models can be fine-tuned for better accuracy and better identification of birds. For fine-tuning, the number of epochs may be increased or some of the layers of the transfer learning models can be unfrozen. The same can be integrated with a mobile app, where common people would be able to upload captured images of birds and the app would provide details of the identified birds.

References

[1] https://archive.epa.gov/water/archive/web/html/birds-initiatives.html.

[2] https://www.gov.uk/government/publications/wild-birds-licence-to-kill-or-take-for-conservation-purposes-gl40/list-of-endangered-woodland-birds.

[3] Mirugwe A, Nyirenda J, Dufourq E. Automating Bird Detection Based on Webcam Captured Images using Deep Learning. InProceedings of 43rd Conference of the South African Insti 2022 Jul 18 (Vol. 85, pp. 62-76).

[4] H¨useyin G¨okhan Ak¸cay, Bekir Kabasakal, Duygug¨ul Aksu, Nusret Demir, Melih Oz, and Ali ¨ Erdo˘gan. Automated bird counting with deep learning for regional bird distribution mapping. Animals, 10(7):1207, 2020.

[5] Lynda Ben Boudaoud, Fr´ed´eric Maussang, Ren´e Garello, and Alexis Chevallier. Marine bird detection based on deep learning using high-resolution aerial images. In OCEANS 2019-Marseille, pages 1–7. IEEE, 2019.

[6] Biswas AA, Rahman MM, Rajbongshi A, Majumder A. Recognition of local birds using different cnn architectures with transfer learning. In2021 International Conference on Computer Communication and Informatics (ICCCI) 2021 Jan 27 (pp. 1-6). IEEE.

[7] M. A. Al-antari, P. Rivera, M. A. Al-masni, E. Valarezo, G. Gi, T. Y. Kim, H. M. Park, and T. S. Kim, "An automatic recognition of multiclass skin lesions via Deep Learning Convolutional Neural Networks," In Conference: ISIC2018: Skin Image Analysis Workshop and Challenge. 2018.

[8] L. Hu, and Q. Ge, "Automatic facial expression recognition based on MobileNetV2 in Real-time," In Journal of Physics: Conference Series, vol. 1549, no. 2, 2020.

[9] Hussain M, Bird JJ, Faria DR. A study on cnn transfer learning for image classification. InUK Workshop on computational Intelligence 2018 Sep 5 (pp. 191-202). Springer, Cham.

[10] Gao, Y., Mosalam, K.: Deep transfer learning for image-based structural damage recognition. Comput. Aided Civ. Infrastruct. Eng. (2018).

[11] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE CVPR 2016: Computer Vision and Pattern Recognition (2016).

[12] Raj S, Garyali S, Kumar S, Shidnal S. Image based bird species identification using convolutional neural network. International Journal of Engineering Research & Technology (IJERT). 2020;9(6):346.

[13] Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., ... & Phung, D. (2017, October). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In 2017 IEEE international conference on data science and advanced Analytics (DSAA) (pp. 40-49). IEEE.

[14] Gavali, P., Mhetre, M. P. A., Patil, M. N. C., Bamane, M. N. S., & Buva, M. H. D. (2019) Bird Species Identification using Deep Learning.

[15] Rahman MM, Biswas AA, Rajbongshi A, Majumder A. Recognition of local birds of Bangladesh using MobileNet and Inception-v3. International Journal of Advanced Computer Science and Applications. 2020;11(8).

[16] S. Lee, M. Lee, H. Jeon, and A. Smith, "Bird Detection in Agriculture Environment using Image Processing and Neural Network," In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 1658-1663, 2019.

[17] J. Bankar, and N. R. Gavai, "Convolutional Neural Network-based Inception v3 Model for Animal Classification," International Journal of Advanced Research in Computer and Communication Engineering, vol. 7, no. 5, pp. 142-146, 2018.

[18] Wang Z, Wang J, Lin C, Han Y, Wang Z, Ji L. Identifying habitat elements from bird images using deep convolutional neural networks. Animals. 2021 Apr 27;11(5):1263.

[19] Guo, Q.H.; Jin, S.C.; Li, M.; Yang, Q.L.; Xu, K.X.; Ju, Y.Z.; Zhang, J.; Xuan, J.; Liu, J.; Su, Y.J.; et al. Application of deep learning in ecological resource research: Theories, methods, and challenges. *Sci. China Earth Sci.* 2020, *63*, 1457–1474.

[20] Islam S, Khan SI, Abedin MM, Habibullah KM, Das AK. Bird species classification from an image using VGG-16 network. InProceedings of

the 2019 7th international conference on computer and communications management 2019 Jul 27 (pp. 38-42).[Accessed: 04-Jun-2019].

[21] Huang YP, Basanta H. Bird image retrieval and recognition using a deep learning platform. IEEE Access. 2019 May 22;7:66980-9.

[22] Branson S, Van Horn G, Belongie S, Perona P. Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv:1406.2952. 2014 Jun 11.

[23] Ragib KM, Shithi RT, Haq SA, Hasan M, Sakib KM, Farah T. Pakhichini: Automatic bird species identification using deep learning. In2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) 2020 Jul 27 (pp. 1-6). IEEE.

[24] Rezaeijo SM, Ghorvei M, Mofid B. Predicting breast cancer response to neoadjuvant chemotherapy using ensemble deep transfer learning based on CT images. Journal of X-ray Science and Technology. 2021 Jan 1(Preprint):1-6.

[25] Rosenberg KV, Dokter AM, Blancher PJ, Sauer JR, Smith AC, Smith PA, Stanton JC, Panjabi A, Helft L, Parr M, Marra PP. Decline of the North American avifauna. Science. 2019 Oct 4;366(6461):120-4.

[26] Shahabuddin G, Goswami R, Krishnadas M, Menon T. Decline in forest bird species and guilds due to land use change in the Western Himalaya. Global Ecology and Conservation. 2021 Jan 1;25:e01447.

[27] Díaz, S., Settele, J., Brondízio, E., Ngo, H., Gueze, M., Agard, J., Arneth, A., Balvanera, P., Brauman, K., Butchart, S., 2020. Summary for Policymakers of the Global Assessment Report on Biodiversity and Ecosystem Services of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services (IPBES).

[28] Burns F, Eaton MA, Burfield IJ, Klvaňová A, Šilarová E, Staneva A, Gregory RD. Abundance decline in the avifauna of the European Union reveals cross-continental similarities in biodiversity change. Ecology and Evolution. 2021 Dec;11(23):16647-60.

[29] Youth H. Birds in Decline. InVital Signs 2003–2004 2021 Apr 14 (pp. 82-83). Routledge.

[30] Sylvain Christin, Eric Hervet, and Nicolas Lecomte. Applications for deep learning in ecology. ´ Methods in Ecology and Evolution, 10(10):1632–1644, 2019.

[31] https://www.kaggle.com/datasets/gpiosenka/100-bird-species.

An Automated Framework to Detect Emotions from Contextual Corpus

Ravikumar Thallapalli[1], G. Narsimha[2]

Research Scholar, Faculty of Informatics-Department of CSE, Osmania University, Hyderabad, Telangana, India[1]
Professor, Department of Computer Science and Engineering, JNTUH University College of Engineering, Sultanpur, Sangareddy, Telangana, India[2]

***Abstract*—The emotion extraction or opinion mining is one of the key tasks for any text processing frameworks. In recent times, the use of opinion mining has gained a lot of potential due to the application of the potential customized aspects of the consumer relations and other customized applications. However, the application of sentiment analysis or opinion mining is highly challenging as the accuracy of the sentiment analysis depends on the input text corpus. The input text corpus can be highly fluctuating due to the inclusion of emojis or local language influences and finally the use of a wide variety of the regional languages. A good number of parallel research outcomes have aimed to solve these challenges in the recent time. However, most of the parallel research outcomes have primarily three challenges kept unsolved as firstly, the emojis in the text corpus is mainly removed but not translated into sentiment scores, secondly, the translation of the texts from various regional languages and the translation is mainly true translations rather than the contextual translation. Finally, the use of the dictionaries in the actual translation tasks takes a lot of time to process and must be reduced. Henceforth, in order to solve these challenges, this work proposed a framework to automate the weighted emoji-based sentiment analysis, Unicode based translation process to reduce the time complexity and finally use the collaborative sentiment analysis scores to build the final sentiment models. This work results into nearly 97% accuracy and nearly 50% reduction in the time complexity.**

Keywords*—Emoji translation; weighted annotation; text translation; reduced unicode based dictionary; relative sentiment score building; mean scoring technique; collaborative sentiment score building

I. Introduction

Weizenbaum created ELIZA in 1964–1966. The artificial intelligence system ELIZA fooled people into thinking it could grasp human language by imitating their speech patterns. The "DOCTOR" was asking random questions in an attempt to learn more about "patient-centered" care. The concept of giving machines emotions shocked Weizenbaum. A mute ELIZA was unable to say a thing [1].

Regular users of social networking sites are more likely to rely on text messaging than any other form of communication. The emotional state of both parties affects the quality of their relationships. Natural language processing has certain issues when dealing with tweets. Despite NLP's development, data was still necessary for earlier work on low-resource languages. The six native Arab taggers at ExaAEC tagged 20,000 tweets. Plutchik's emotional "paradigm" includes these 10 categories. LSTM and ELMO are used to categorize feelings. On the F1 scale, the agreement between our data and model is 0.65. The elimination of emotions changes [2]. The focus here is not on classifying or extracting emotions, but on understanding what sets them in action. Because we could not find any publicly accessible datasets, we opted to generate one using the Emotion Markup Language from the W3C. Here we detail how one may use a tree-based event representation framework to trace the roots of an emotional state. Under sampling-based bagging may be used to level out the training data's unequal distribution of events. The proposed approach can extract SVM features effectively with little training.

Natural language processing experts can decipher the underlying meaning of a written message by analyzing the writer's tone [3].

As it is, the problem is seen as a multi-label classification problem by the state of the art. These algorithms do not consider how a text makes the reader feel. Second, because diverse text fragments may predict different emotion labels, the model must gather effective attributes for each emotion. For this purpose, we use a topic-enhanced capsule network in conjunction with a variational autoencoder to classify a broad variety of emotions. The latent topic of a sentence is utilized to train a variational autoencoder, and a capsule module is employed for sentiment analysis. The proposed model outperforms the state-of-the-art [4].

Henceforth, after setting the context of this research in this section of the work, the rest of the work is furnished such as in the Section II, III and IV the foundational methods are discussed and based on the foundation methods, in the Section V the recent improvements are discussed. Further, in the Section VI and VII the identified problems and the proposed solutions are furnished. In the Section VIII, based on the proposed solutions, the algorithms and the frameworks are discussed. Furthermore, in the Section IX, the obtained results are discussed and compared with the existing research outcomes. Finally, the research conclusion is presented in the Section X.

II. Foundational Method for Emoji Detection

In this section of the work, the foundational method for the emoji detection is furnished. The detection of the emojis is one of the prime factors during any text processing tasks. The accuracy of any machine learning driven algorithms relies on the correctness of the input data and the datasets with emojis can deviate the accuracy of the machine learning methods to a greater extent.

Assuming that, the complete text corpus is T[] and the prime components in the dataset are the sequence number of the data items, n, and the actual text as t. This can be formulated as,

$$T[] = < n, t > \tag{1}$$

Further, the base line method indicates that, the set of uni-code characters, UC[], to be separated, t*[], from the actual text and must be handled separately, as,

$$t^*[] = \prod_{t[i]=UC} t[i] \tag{2}$$

Furthermore, the extracted Unicode characters must be validated against the specified range of the unicodes and must be assigned to the emoji collection as,

$$E[] = Iff\ \ t[] \subset UC[UC\_range] \tag{3}$$

Henceforth, the unicode removed text corpus, T1[], shall be presented as,

$$T1[] = T1[] - E[] \tag{4}$$

Further, in the next section of this work, the foundational method for the text translation is furnished.

III. Foundational Method for Text Translation

In this section of the work, the baseline method for the text translation is carried out. The text translation for the text processing projects plays a very important role. The translation of the text implies the verification of the text analysis process and majority of the text analyzers focus on the English dictionary. Hence, the translation of the text from other native languages is highly appreciated for any text mining applications.

Continuing from the Eq. (4), the text corpus, which is free from the emojis, must be further filtered to remove the numeric values as,

$$t[] = \prod_{T1[i] \notin Numeric\_Unicodes} T1[] \tag{5}$$

Also, the baseline method defines the use of the standard language dictionaries, D[], which contains the source language literals, s, and the destination language literals, d. This can be formulated as,

$$D[] = < s, d > \tag{6}$$

Further, the translated text, T2[] can be formulated as,

$$T2[] = \prod_{T2[i]==D[].s} D[].d \tag{7}$$

Henceforth, the final translated text, T2[] can be taken to the next phase of the process.

Further, in the next section of this work, the emotion detection foundational method is discussed.

IV. Foundational Method for Emotion Detection

In this section of the work, the baseline method for the emotion detection or opinion detection is analyzed.

Continuing from the Eq. (7), the text corpus further must be separated, T_X based on the stop words, connectors and finally based on the phases as,

$$T_X[] = \frac{d}{d(T2[])} T2[] \tag{8}$$

Also, the baseline method suggests to store all bag of words, B[], and further use the BoW to identify the set of sentiments with text phase, p, and sentiment score, sc. This can be realized as,

$$B[] = < p, sc > \tag{9}$$

Finally, based on the BoW, the sentiment collection, S[], must be designed. This can be formulated as,

$$S[] = \prod_{T_X[i] \approx B[].p} B[].sc \tag{10}$$

After the detailed analysis of the baseline methods in the previous sections of this work, in the next section of the work, the recent improvements over the baseline methods are discussed.

V. Recent Research Reviews

Extracting useful information from Twitter profiles is now trendy. From what you've stated, Twitter may infer how you feel about a topic. The newest graphs can detect your emotional state. It's only possible to utilize Western languages. The ability to recognize Indonesian speakers' emotions has been enhanced. Feelings of joy, sadness, fear, surprise, disgust, dread, and anticipation or fury (marah). As a result of its speakers' imbalanced emotions, Indonesian is imprecise. According to the findings, adjusting the pattern weight will help make the Indonesian data more comparable. The recommended strategy increases minority class accuracy (fear, surprise, and disgust) [5].

Emotional states impact the dynamics of interpersonal relationships. Text, speech, and body language all contribute to the communication of emotion, caring blogs and social media. Politics, both domestic and foreign, are discussed. The ability to evaluate the emotions described in books is helpful for success in social settings. The study of emotions in Bangla is quite new. Multi-class emotions are extracted from Bangla text using NB classifier, stemmer, POS tagger, n-grams, and word frequency-inverse document frequency analysis (tf-idf). [6] An overall accuracy of 78.6% was achieved in the final model.

Syntactically, NLP categorises expressions of emotion (NLP). When confronted with a classification issue with multiple labels, current methods perform well. This means that none of the methods can be used to demonstrate the model's ability to predict spatial correlations in sample data. In my opinion, a fantastic strategy to improve efficiency and productivity is to use contrastive learning. To maximize the positives and minimize the negatives, they apply a compound loss function on top of a tweaked version of the commonly used Cross Entropy loss function in training. My recommendations were successful [7].

Better computer-human interactions, monitoring mental health, and adapting corporate strategies are just a few examples of the many applications for emotion detection. Deep learning outperforms all other AI methods in every measurable way. Single-emotion and multi-emotion samples are used by all machine learning algorithms for emotion identification. Our research shows that these systems benefit more from single-emotion instances than from mixed-emotion ones. Single-emotion works are unusual. Inefficient procedures exist. We used all available information to create a dataset where each piece of text represents at least two different emotions. Since CNNs are so good at analyzing pictures, we modified their design to determine how readers would feel about a given passage. We used rapid text and GloVe to detect semantic and syntactic similarities between text and numerical expressions. There has to be more accuracy in the construction plans; excellent estimate [8].

The encoder is bidirectional. Both Transformers representations and the Universal Sentence Encoder make advantage of transfer learning. The problem of overfitting arises in language models trained using a classifier on a limited data set. The majority of current studies focus on generalizing deep language models to new fields. When it comes to determining how someone is feeling, DeepEmotex turns on transfer learning. Twitter's data on user sentiment is collected, analyzed, and utilized for further refinement. Twitter is vital to its users. DeepEmotex has a 91% accuracy rate. EmoInt and Stimulus both validate Deep-Emotex. An average level of emotional engagement is 73%. When compared to Bi-LSTM, DeepEmotex-BERT performs 23% better. Both the extent of the dataset and the effectiveness of the models are examined. Modeling the task at hand and refining it based on how the intended recipient is experiencing [9].

The ASR capabilities are used by EPD (DSFs). Decoding DSFs through ASR is a little pricey for embedded systems. Based on the word history of the whole ASR system, this study presents a language model (LM)-based end-of-utterance (EOU) predictor that does not need a decoding approach in the test phase. Using a novel full-stack EPD approach, audio and language modelling techniques based on predictive encoding (PE) and explicit outcome usage (EOU) predictors, respectively, are combined into a single RNN-based AFE-based EPD strategy. Ensemble RNNs trained with information from the LM-based EOU predictor, the acoustic model, and the AFE-based EPD for each target independently are used in the proposed EPD (AM). An ensemble of RNNs is linked to a DNN to create the EPD classifier. Following that, they get retraining in an effort to cut down on errors once they reach their ultimate destination. The EPD framework analyzed the proportion of incorrect CHiME-3 endpoint responses versus the proportion of responses that were right. There is an enhancement here compared to EPD [10].

Low resource - It is estimated that 230 million individuals worldwide have some level of fluency in Urdu. Researchers put together benchmark datasets in languages with limited resources. The English to Urdu language pair is one way of translation. Google Translate has recently been updated to include the ability to handle polarity shifts and other forms of natural language processing. The accuracy of sentiment analysis and emotion recognition suffers as a consequence. This study's primary objective is to categorize people's emotions in light of whether or not they have a lot of available resources to work with. We describe five classes of polarity-shifting nouns and verbs. It finds commonalities in the genesis of several languages. The accuracy of sentiment classification drops by 2% to 3% when transitioning from a language with a lot of resources to one with less [11].

Irony is the polar opposite of this. When analyzing text for sarcasm, emojis are seldom employed. This article discusses the use of satire on Weibo. After analyzing visual input, such as facial expressions, bi-LSTM networks next consult textual resources for more insight. Make your forecast using one of these three methods. Our comprehensive mining of sarcastic Weibo [12] was a smashing success.

The study of emotions is quite popular presently. Research on body language lags behind that on facial expressions and words. Surveys are designed to encourage participation from the general public. We take a look at the cultural and racial variations in body language. Human bodily movements are readable by our technology. Here, we investigate the use of RGB and 3D data for estimating body position. Current research on representation learning and emotion recognition in gestures is discussed. Similarly, speech and body multimodal techniques may be used. Massive scale data analysis of human identity, position assessment, and mood is now possible. The output spaces are inconsistent, and there are only simple representations [13].

Cyberbullying, especially among teenagers, has increased along with the development of new technologies. This article explores the role that emotions play in the identification of cyberbullying in India. We created BullySentEmo since there was no preexisting Twitter dataset that categorized bullying, sentiment, and emotion. Emoticons and short tweet-like text are encouraged to be embedded. In India, social media users often switch back and forth between the two languages. Multiple emoji manipulation is a feature of MT-MM-Bert+VecMap[14].

One way in which the prevalence of social media has altered the way we express ourselves verbally is by making us more likely to use strong language. Recognizing unsuitable content is a vital skill for anybody using the internet. The abundance of daily data has made automatic recognition a need. The effectiveness of hate speech detection algorithms is currently being studied. These methods are unable to detect polarity or tone in the language being studied. This is the first multi-task transformer-based technique for detecting hate speech in Spanish. Harassing communication that singles out certain groups often appeals to strong emotions or extreme opposites [15].

Opinion miners and researchers into human-computer interaction use different approaches to sentiment analysis. In this piece, we emphasize the importance of working across disciplines. In human-agent interactions, sentiment/opinion detection algorithms are often used for opinion mining rather than socio-affective interactions (timing constraint of the interaction, sentiment analysis as an input and an output of interaction strategies). To substantiate our arguments, we look at phenomena connected to emotions, sentiment recognition technology, and the goals of socio-affective human-agent strategies. The next steps that must be taken and the unanswered issues are discussed. For the purpose of providing a more precise sentiment analysis [16], we include the specified criteria into the Greta platform for humanoid conversational robots.

An approach of feature extraction is required for micro expressions. Multi-feature fusion may be used to detect micro-expressions. This technique establishes a connection between projection error and LBP features. In order to achieve rapid and precise identification, data utilized in the study was painstakingly extracted from specialized facial expression databases. In advanced learning environments, the novel method outperforms LBP; identification of objects in a photograph [17].

Recent research has shifted its emphasis from words to audio and video's non-verbal clues in order to make automated assessments of people's mental health. Textual content is as important as audio and video for depression detection systems. Comprehensively automated depression evaluation approaches need complex models of aural, visual, and textual elements. Anyone interested in learning more about depression may do so by perusing the audio, video, and textual materials provided here. A hybrid design consists of three main parts. a) A Deep Convolutional Neural Network (DCNN) and a Deep Neural Network (DNN) model for estimating the PHQ-8 based on audio-visual multi-modal depression recognition; b) a Paragraph Vector (PV) and a Support Vector Machine (SVM) model for inferring physical and mental conditions from interview transcripts; and c) a Random Forest (RF) model for depression classification using the estimated PHQ-8 sc. To identifying psychoanalytic symptoms, PV-SVM incorporates psychoanalytic questions pertaining to depression into fixed-length feature vectors and feeds them into SVM classifiers. Examining the PV drop for the first time. Identifying face characteristics in motion may be monitored in both time and space using HDR. The hybrid framework outperforms AVEC2016 in terms of depression estimates and classification on average by an F1 score of 0.746. (0.724). HDR outperforms BoW and MHH in identifying depressive episodes [18].

VI. Problem Formulation–Mathematical Model

Henceforth, after the detailed analysis of the existing systems in the previous sections of this work, in this section, the persisting research problems are furnished.

Firstly, the existing system is successful into detecting and separating the emojis, however the same detected emojis are not translated to sentiment scores.

Continuing from the Eq. (3) and Eq. (4), it is natural to realize that, the emojis collection, E[] is not been translated to emotion as,

$$E[] \not\subset T1[] \tag{11}$$

And Only,

$$T1[] \rightarrow S[] \tag{12}$$

Secondly, the baseline method recommends using the traditional dictionaries to convert the multi-lingual text corpuses to the standard text; however the text translating dictionary size can be overwhelming due to the fact that, the systems intended to translate a wide variety of the source languages.

Assuming that, the average size of the dictionary for one language is n and the diction, D[], is furnished for translating m number of languages. Hence, from the Eq. (7), the time complexity, T, can be furnished as,

$$T = n * m \tag{13}$$

Or,

$$T(n) = O(n^2) \mid n \approx m \tag{14}$$

Finally, the detection process for the emotion is not contextual.

As per the Eq. (10), the existing system builds the sentiment scores for text phases. Assuming that, the sentiment scores for two consecutive text phases are S[i] and S[j], then this can be furnished as,

$$S[i] \Leftarrow T_X[i] \tag{15}$$

And,

$$S[j] \Leftarrow T_X[j] \tag{16}$$

However, here the text corpuses are not inter-related as,

$$T_X[i] \not\subset T_X[j] \tag{17}$$

Thus, it is natural to realize that,

$$S[i] \not\subset S[j] \tag{18}$$

Henceforth, based on the identified challenges in the existing systems, in the next section of the work, the proposed solutions are furnished.

VII. Proposed Solutions

After the detailed analysis of the existing methodologies and the persistent research problems, in this section of the work, the proposed solutions are furnished.

Firstly, the emoji detection and translation to the sentiment process is furnished here.

Assuming that, the extracted text from the corpus, EJ[], have two prime component as enumerated id, e, for each text corpus and the text as t. This can be presented as,

$$EJ[] = < e, t > \qquad (19)$$

Further, based on the baseline method, the emojis are separated from the actual text and kept in a separate collection, E1[]. This can be formulated as,

$$E1[] = \prod_{EJ[i]=UC} EJ[i].t \qquad (20)$$

Further, the collection of emojis, E_X, which is already pre-build with the emotion scores, sc. This can be presented as,

$$E_X[] = < e, sc > \qquad (21)$$

Now, the extracted emojis must be translated to the sentiment scores, ES[], and kept for further processing.

$$ES[] = \prod_{E1[i]\approx E_X[]} E_X[].SC \qquad (22)$$

Secondly, the process for the translation process using the unicode driven dictionary is furnished.

Continuing from the Eq. (6), assuming that the unicodes, UC[], for each source and destination language phase can be extracted using the following hypothetical function as,

$$UC[] = \lambda\{D[].s\} \qquad (23)$$

And,

$$UC[] = \lambda\{D[].d\} \qquad (24)$$

Henceforth, assuming that, for two dictionaries D1[] and D2[], the unicodes are extracted as UC1[] and UC2[]. Now it is natural to realize that, both the unicode collections must have number of similar unicodes and thus, based on this situation, the length of the final dictionary, FD[], can be configured as,

$$FD[] = \{UC1[] \bigcup UC2[]\} - \{UC1[] \bigcap UC2[]\} \qquad (25)$$

This process can be repeated for all the dictionaries to be included and still the complexity for processing the total dataset can be limited to O(n/2) during the average case complexity.

Finally, the process for the relative sentiment score analysis process is furnished.

Continuing from the Eq. (10), assuming that for two text phases, $T_X[i]$ and $T_X[j]$, the sentiment scores S[i] and S[j], which can be furnished as,

$$S[i] = \prod_{T_X[i]\approx B[].p} B[].sc \qquad (26)$$

And,

$$S[j] = \prod_{T_X[j]\approx B[].p} B[].sc \qquad (27)$$

Further, for the relative and collaborative sentiment score analysis, the following process is furnished as,

$$\begin{aligned}&Iff\ \ S[i] \neq S[j],\\ &Then\ S[j] = Mean\{S[i], S[j]\}\ and\ S[i] \bigcup S[j]\\ &Else\ S[i] \bigcup S[j]\end{aligned} \qquad (28)$$

Further, for building final collaborative scores, the following strategy can be applied to include the sentiment scores from the emojis as,

$$\begin{aligned}&Iff\ \ S[i] \neq S[j],\\ &Then\ S[j] = Mean\{S[i], S[j]\}\ and\ S[i] \bigcup S[j] \bigcup ES[]\\ &Else\ S[i] \bigcup S[j] \bigcup ES[]\end{aligned} \qquad (29)$$

Furthermore, based on the proposed mathematical models, in the next section of this work, the proposed algorithms are furnished.

VIII. Proposed Algorithms and Frameworks

After the detailed analysis of the existing system, challenges in the existing systems and the proposed mathematical models, in this section of the work, the implementable versions of the proposed algorithms are furnished.

Firstly, the Emoji Translation using Weighted Annotation (ET-WA) Algorithm is furnished.

Algorithm - I: Emoji Translation using Weighted Annotation (**ET-WA**) Algorithm

Input:
Text Corpus as EJ[]

Output:
Sentiment scores for the Emojis as ES[]
Emoji Reduced Text Corpus as TT[]

Process:

```
Step - 1.   Load the text corpus dataset as EJ[]
Step - 2.   For each member of the EJ[] collection as EJ[i]
                a.   Extract the text component as EJ[i].t
                b.   If EJ[i].t belongs to Ex[] Then,
                         i.   Extract the emotion score as ES[j] =
                              EJ[i].SC
                        ii.   Build the reduced text corpus as TT[i] =
                              EJ[i] - EJ[i].t
                c.   Else, Build the sentiment score for the emoji using Eq.
                     21 & 22
Step - 3.   Return ES[] & TT[]
```

Secondly, the Text Translation using Reduced Unicode Based Dictionary (TT-RUBD) Algorithm is furnished.

Algorithm - II: Text Translation using Reduced Unicode Based Dictionary (**TT-RUBD**) Algorithm

Input:
Text Corpus as TT[]
Dictionary as Dn[]

Output:
Translated Text as TST[]

Process:

```
Step - 1.   Load the dictionary collection as D[]
Step - 2.   For each dictionary type in D[] as D[i]
                a.   Translate D[i] into Unicode as D[i].UC[j]
                b.   If D[i].UC[j] == D[i+1].UC[j+1] Then,
                  i.  Build the reduced dictionary, DR[i] =
                      D[i].UC[j]
Step - 3.  Load the emoji reduced dataset as TT[]
Step - 4.  For each member in the TT[] collection as TT[k]
           a.  If TT[k].t belongs to DR[k] Then,
                  i.  TST[i] = DR[k].d
Step - 5.  Return TST[]
```

Thirdly, Relative Sentiment Score Building using Mean Scoring Technique (RSB-MST) Algorithm is furnished here

Algorithm - III: Relative Sentiment Score Building using Mean Scoring Technique (**RSB-MST**) Algorithm

Input:
Translated Text as TST[]

Output:
Sentiment Score as SS[]

Process:

```
Step - 1.  Accept the translated text as TST[]
Step - 2.  For each data item in TST[] as TST[i]
           a.  Separate the text corpus based on stop words in
               PH[j]
           b.  Calculate the sentiment score as SS[i] using Eq.
               26
           c.  If SS[i] Not_Equal SS[j] Then,
                  i.  SS[j] = Mean {SS[i], SS[j]}
           d.  Else, Collect SS[i] & SS[j]
Step - 3.  Return SS[]
```

Finally, Collaborative Score Building for Text Corpus (CSB-TC) Algorithm is furnished.

Algorithm - IV: Collaborative Score Building for Text Corpus (**CSB-TC**) Algorithm

Input:
Sentiment score from text as SS[]
Sentiment score from emojis as ES[]

Output:
Final Sentiment score as FS[]

Process:

```
Step - 1.  Accept the sentiment scores as SS[] and ES[]
Step - 2.  For every element in SS[] as SS[i]
           a.  If SS[i] == ES[j] or SS[i] < ES[j] Then,
                  i.  FS[i] = Mean {SS[i] + ES[j]}
           b.  Else, FS[i] = ES[j]
Step - 3.  Return FS[]
```

The details of these algorithms are furnished in the previous sections of this work.

Further, the automated framework is presented here (Fig. 1).

Identifying if a document, sentence, or object feature/aspect is conveying a favorable, negative, or neutral viewpoint is a fundamental task in sentiment analysis. A wide range of human emotions, including as joy, anger, disgust, sadness, fear, and surprise, are analyzed by sentiment categorization algorithms that look "beyond polarity."

One of the earliest examples of what would become modern sentiment analysis was published in The General Inquirer, while other precedents include independent psychological investigations that analyzed a person's mental state by analyzing their speech.

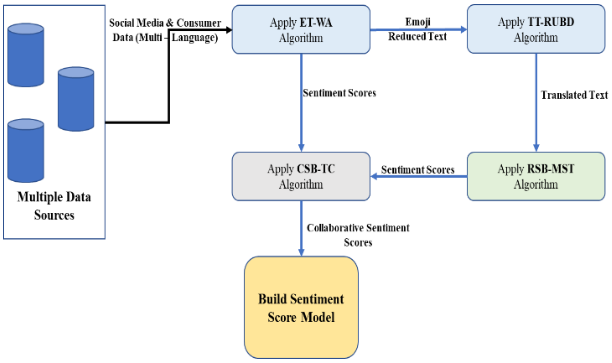


Fig. 1. Proposed framework.

As a result, the method disclosed in Volcani and Fogel's patent zoomed in on sentiment and singled out words and phrases in text with respect to various emotional scales. Based on their findings, EffectCheck provides a list of interchangeable words that can be used to adjust the level of emotional impact.

Further, in the next section of this work, the obtained results are discussed.

IX. Results and Discussions

After the analysis of the existing system and the proposed system, in this section of the work, the obtained results are discussed.

Firstly, the dataset descriptions are furnished here (Table I).

TABLE I. DATASET ANALYSIS [19, 20]

Dataset Name	*Number of Instances*	*Release Date*
Sentiment140 [19]	16,000	2009
Amazon Reviews for Sentiment Analysis [20]	25,900	2022

This is the sentiment140 dataset. It contains 16,000 tweets extracted using the twitter api. The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

Also, the second dataset, this dataset consists of a 25,900 Amazon customer reviews (input text) and star ratings (output labels) for learning how to train fastText for sentiment analysis.

The framework is tested on all the data items, however, only 15 for each dataset is furnished here.

The outcome of the emoji detection is displayed here (Table II).

TABLE II. EMOJI DETECTION ANALYSIS

Dataset Name	*Trial Seq (#)*	*Number of Actual Emojis*	*Number of Detected Emojis*	*Detection Accuracy (%)*
[19]	1	4	3	75.00
[19]	2	7	6	85.71
[19]	3	7	5	71.43
[19]	4	7	5	71.43
[19]	5	6	5	83.33

Dataset Name	*Trial Seq (#)*	*Number of Actual Emojis*	*Number of Detected Emojis*	*Detection Accuracy (%)*
[19]	6	4	2	50.00
[19]	7	7	6	85.71
[19]	8	6	5	83.33
[19]	9	6	4	66.67
[19]	10	7	7	100.00
[19]	11	7	5	71.43
[19]	12	5	3	60.00
[19]	13	5	3	60.00
[19]	14	4	4	100.00
[19]	15	7	6	85.71
[20]	1	4	4	100.00
[20]	2	8	7	87.50
[20]	3	9	9	100.00
[20]	4	9	8	88.89
[20]	5	5	5	100.00
[20]	6	6	5	83.33
[20]	7	8	6	75.00
[20]	8	7	5	71.43
[20]	9	5	5	100.00
[20]	10	9	7	77.78
[20]	11	7	7	100.00
[20]	12	6	6	100.00
[20]	13	8	8	100.00
[20]	14	6	6	100.00
[20]	15	6	6	100.00

The detection process is highly time-efficient, which is furnished in the further part of this work, and highly accurate. The accuracy of the emoji detection for the displayed sample is nearly 85% and for the total datasets, the accuracy is nearly 93%.

The outcome is also visualized graphically here (Fig. 2).

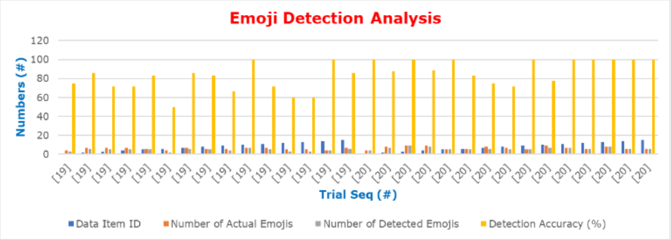


Fig. 2. Emoji detection accuracy analysis.

Further, the translation scores are also analyzed here (Table III). The validation process for translation is compared with the google translation and the obtained scores are compared.

The translations scores obtained from the proposed system are highly appreciable. The average accuracy for the displayed samples is 98% and for the complete dataset the accuracy is 99.89%.

TABLE III. TRANSLATION ANALYSIS

Dataset Name	*Trial Seq (#)*	*Translation Score (Google)*	*Translation Score (Proposed System)*	*Translation Accuracy (%)*
[19]	1	95	90	94.74
[19]	2	95	93	97.89
[19]	3	90	90	100.00
[19]	4	92	95	100.00
[19]	5	93	95	100.00
[19]	6	100	95	95.00
[19]	7	94	96	100.00
[19]	8	99	95	95.96
[19]	9	94	91	96.81
[19]	10	98	97	98.98
[19]	11	93	95	100.00
[19]	12	94	94	100.00
[19]	13	94	99	100.00
[19]	14	89	98	100.00
[19]	15	90	97	100.00
[20]	1	89	96	100.00
[20]	2	93	95	100.00
[20]	3	97	90	92.78
[20]	4	95	92	96.84
[20]	5	100	97	97.00
[20]	6	90	98	100.00
[20]	7	97	92	94.85
[20]	8	92	93	100.00
[20]	9	96	93	96.88
[20]	10	90	100	100.00
[20]	11	90	95	100.00
[20]	12	95	92	96.84
[20]	13	97	100	100.00
[20]	14	99	92	92.93
[20]	15	100	98	98.00

The outcome is also visualized graphically here (Fig. 3).

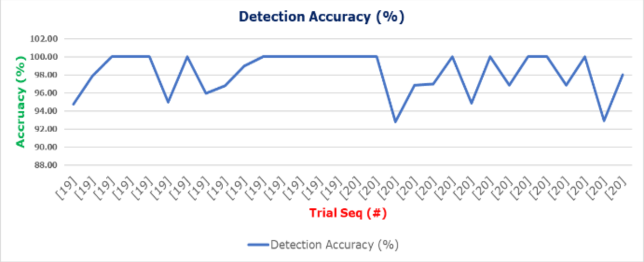


Fig. 3. Translation accuracy analysis.

Further, the detection sentiment scores from the emojis are furnished here (Table IV).

TABLE IV. SENTIMENT DETECTION FROM EMOJI ANALYSIS

Dataset Name	*Trial Seq (#)*	*Number of Emojis*	*Extracted Sentiment Scores (Mode)*
[19]	1	3	5
[19]	2	6	1
[19]	3	5	3
[19]	4	5	3
[19]	5	5	1
[19]	6	2	5
[19]	7	6	3
[19]	8	5	5
[19]	9	4	3
[19]	10	7	2
[19]	11	5	5
[19]	12	3	5
[19]	13	3	1
[19]	14	4	5
[19]	15	6	2
[20]	1	4	1
[20]	2	7	5
[20]	3	9	2
[20]	4	8	5
[20]	5	5	5
[20]	6	5	3
[20]	7	6	1
[20]	8	5	4
[20]	9	5	5
[20]	10	7	1
[20]	11	7	1
[20]	12	6	3
[20]	13	8	3
[20]	14	6	1
[20]	15	6	3

During the translation and extraction of the sentiment scores from the emojis the classes are denoted as 1 as very negative to 5 as very positive. However, the overall sentiment scores from the emojis are calculated using the mode method.

The outcome is also visualized graphically here (Fig. 4).

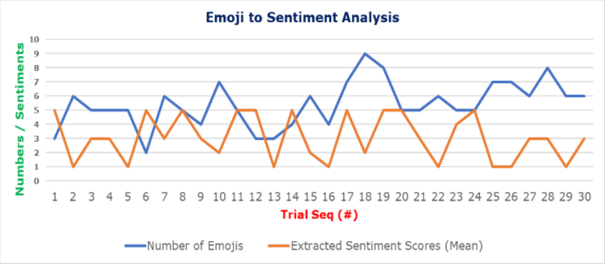


Fig. 4. Sentiment extraction from emojis analysis.

Further, the sentiment analysis collaboratively from the text and from the emojis is performed here and the results are furnished (Table V).

TABLE V. COLLABORATIVE SENTIMENT EXTRACTION ANALYSIS

Dataset Name	*Trial Seq (#)*	*Text Length (N umber of Characters)*	*Extracted Sentiment from Text (Mode)*	*Extracted Sentiment from Emojis (Mode)*	*Collaborative Sentiment Scores (Mean)*
[19]	1	1	26	2	5
[19]	2	2	26	2	1
[19]	3	3	27	3	3
[19]	4	4	26	2	3
[19]	5	5	28	1	1
[19]	6	6	28	3	5
[19]	7	7	29	3	3
[19]	8	8	24	1	5
[19]	9	9	27	5	3
[19]	10	10	27	1	2
[19]	11	11	32	1	5
[19]	12	12	24	1	5
[19]	13	13	30	2	1
[19]	14	14	27	3	5
[19]	15	15	25	5	2
[20]	1	1	28	4	1
[20]	2	2	32	2	5
[20]	3	3	32	3	2
[20]	4	4	32	5	5
[20]	5	5	22	3	5
[20]	6	6	31	1	3
[20]	7	7	25	3	1
[20]	8	8	24	5	4
[20]	9	9	29	2	5
[20]	10	10	29	2	1
[20]	11	11	28	2	1
[20]	12	12	23	1	3
[20]	13	13	28	4	3
[20]	14	14	26	2	1
[20]	15	15	28	3	3

The outcome is also visualized graphically here (Fig. 5).

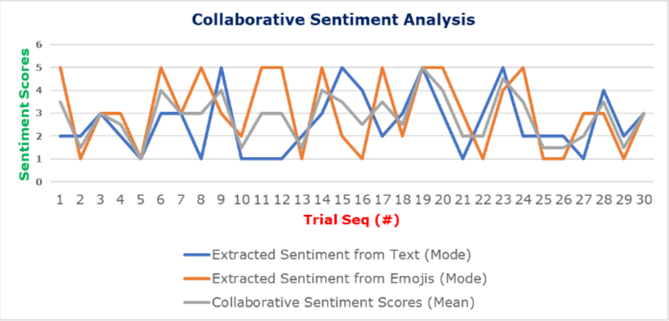


Fig. 5. Collaborative sentiment extraction analysis.

Finally, the time complexity analysis is carried out (Table VI).

TABLE VI. SENTIMENT ANALYSIS TIME ANALYSIS

Dataset Name	***Trial Seq (#)***	***Text Length (N number of Characters)***	***Number of Emojis Detected***	***Time (ns)***
[19]	1	26	3	0.207
[19]	2	26	6	0.252
[19]	3	27	5	0.254
[19]	4	26	5	0.169
[19]	5	28	5	0.190
[19]	6	28	2	0.303
[19]	7	29	6	0.200
[19]	8	24	5	0.131
[19]	9	27	4	0.157
[19]	10	27	7	0.310
[19]	11	32	5	0.079
[19]	12	24	3	0.170
[19]	13	30	3	0.094
[19]	14	27	4	0.061
[19]	15	25	6	0.068
[20]	1	28	4	0.112
[20]	2	32	7	0.325
[20]	3	32	9	0.321
[20]	4	32	8	0.142
[20]	5	22	5	0.325
[20]	6	31	5	0.333
[20]	7	25	6	0.238
[20]	8	24	5	0.180
[20]	9	29	5	0.077
[20]	10	29	7	0.093
[20]	11	28	7	0.174
[20]	12	23	6	0.121
[20]	13	28	8	0.131
[20]	14	26	6	0.228
[20]	15	28	6	0.291

It is natural to observe that, the time complexity mean is 0.19 ns for an average length of 28 characters with average of 5 emojis.

The outcome is also visualized graphically here (Fig. 6).

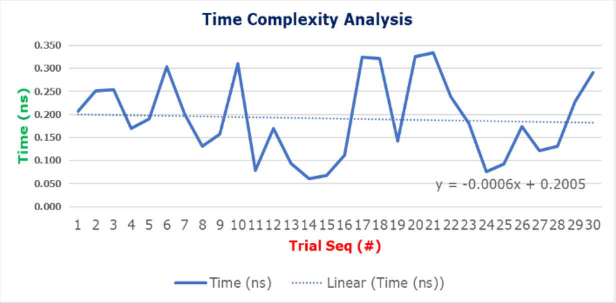


Fig. 6. Time complexity analysis.

X. CONCLUSION

One of the most important jobs for any text processing system is emotion extraction, sometimes called opinion mining. The potential personalized parts of customer relations and other customized applications have given opinion mining a lot of potentials in recent years. However, since the quality of the sentiment analysis is dependent on the text corpus that is used for the analysis, putting it to use can be somewhat difficult. Due to factors such as emoji usage, regional language effects, and the use of many different regional languages, the input text corpus can be quite volatile. Many recent study findings have taken a multi-pronged approach to addressing these difficulties. However, the majority of the results from similar studies fail to address three key issues: (1) the removal of emojis from the text corpus without converting them into sentiment scores; (2) the translation of texts from different regional languages; and (3) the translation of texts is primarily literal rather than contextual. Finally, reducing dictionary use in translation activities is important because it is a time-consuming procedure. As a result, this study provided a framework for automating the weighted emoji-based sentiment analysis, streamlining the Unicode-based translation process to cut down on time complexity, and using the collaborative sentiment analysis scores to construct the final sentiment models. The temporal complexity is reduced by approximately half as much as a result of this study and the accuracy is nearly as high as 97%.

REFERENCES

[1] D. S. Moschona, "An Affective Service based on Multi-Modal Emotion Recognition, using EEG enabled Emotion Tracking and Speech Emotion Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), Seoul, Korea (South), 2020, pp. 1-3.

[2] S. Sarbazi-Azad, A. Akbari and M. Khazeni, "ExaAEC: A New Multi-label Emotion Classification Corpus in Arabic Tweets," 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE), Mashhad, Iran, Islamic Republic of, 2021, pp. 465-470.

[3] R. Xu, J. Hu, Q. Lu, D. Wu and L. Gui, "An ensemble approach for emotion cause detection with event extraction and multi-kernel SVMs," in Tsinghua Science and Technology, vol. 22, no. 6, pp. 646-659, December 2017.

[4] H. Fei, D. Ji, Y. Zhang and Y. Ren, "Topic-Enhanced Capsule Network for Multi-Label Emotion Classification," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 1839-1848, 2020.

[5] L. Farsiah, Y. -S. Chen and A. Misbullah, "Multi-Classes Emotion Detection for Unbalanced Indonesian Tweets," 2020 International Conference on Electrical Engineering and Informatics (ICELTICs), Aceh, Indonesia, 2020, pp. 1-6.

[6] S. Azmin and K. Dhar, "Emotion Detection from Bangla Text Corpus Using Naïve Bayes Classifier," 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2019, pp. 1-5.

[7] A.Sindhura, J.Rajeshwar, M.V.Narayana , M.Ram Babu, "An Effective Semantic Web Knowledge Processing Mechanism by Using an Adaptive Swarm Intelligence Technique for Ontology (ASITO)", International Journal of Engineering Trends and Technology, Volume 69 Issue 3, 195-200, March 2021.

[8] Niladri Shekar Dey, Purnachand Kollapudi, M V Narayana, I Govardhana Rao, "An Automated Framework for Detecting Change in the Source Code and Test Case Change Recommendation", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 11, No. 8, 2020, pp.270-280.

[9] R. Shi and T. Chen, "Emotion Detection with Deep Neural Network and Contrastive Learning," 2022 14th International Conference on Computer

Research and Development (ICCRD), Shenzhen, China, 2022, pp. 83-93.

[10] I. Hwang and J. -H. Chang, "End-to-End Speech Endpoint Detection Utilizing Acoustic and Language Modeling Knowledge for Online Low-Latency Speech Recognition," in IEEE Access, vol. 8, pp. 161109-161123, 2020.

[11] A. Ghafoor et al., "The Impact of Translating Resource-Rich Datasets to Low-Resource Languages Through Multi-Lingual Text Processing," in IEEE Access, vol. 9, pp. 124478-124490, 2021.

[12] Z. Yin and F. You, "Multi-Modal Sarcasm Detection in Weibo," 2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT), Changsha, China, 2021, pp. 740-743.

[13] F. Noroozi, C. A. Corneanu, D. Kamińska, T. Sapiński, S. Escalera and G. Anbarjafari, "Survey on Emotional Body Gesture Recognition," in IEEE Transactions on Affective Computing, vol. 12, no. 2, pp. 505-523, 1 April-June 2021.

[14] K. Maity, A. Kumar and S. Saha, "A Multitask Multimodal Framework for Sentiment and Emotion-Aided Cyberbullying Detection," in IEEE Internet Computing, vol. 26, no. 4, pp. 68-78, 1 July-Aug. 2022.

[15] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López and M. T. Martín-Valdivia, "A Multi-Task Learning Approach to Hate Speech Detection Leveraging Sentiment Analysis," in IEEE Access, vol. 9, pp. 112478-112489, 2021.

[16] C. Clavel and Z. Callejas, "Sentiment Analysis: From Opinion Mining to Human-Agent Interaction," in IEEE Transactions on Affective Computing, vol. 7, no. 1, pp. 74-93, 1 Jan.-March 2016.

[17] L. Mao, N. Wang, L. Wang and Y. Chen, "Classroom Micro-Expression Recognition Algorithms Based on Multi-Feature Fusion," in IEEE Access, vol. 7, pp. 64978-64983, 2019.

[18] L. Yang, D. Jiang and H. Sahli, "Integrating Deep and Shallow Models for Multi-Modal Depression Analysis—Hybrid Architectures," in IEEE Transactions on Affective Computing, vol. 12, no. 1, pp. 239-253, 1 Jan.-March 2021.

[19] Twitter Sentiment140 dataset, URL - https://www.kaggle.com/datasets/kazanova/sentiment140.

[20] Amazon Reviews for Sentiment Analysis, URL - https://www.kaggle.com/datasets/bittlingmayer/amazonreviews.

Rapidly Exploring Random Trees for Autonomous Navigation in Observable and Uncertain Environments

Fredy Martínez, Edwar Jacinto, Holman Montiel
Universidad Distrital, Francisco José de Caldas
Bogotá D.C., Colombia

***Abstract*—This paper proposes the use of a small differential robot with two DC motors, controlled by an ESP32 microcontroller, that implements the Rapidly Exploring Random Trees algorithm to navigate from an origin point to a destination point in an unknown but observable environment. The motivation behind this research is to explore the use of a low-cost, versatile and efficient robotic platform for autonomous navigation in complex environments. This work presents a practical and cost-effective solution that can be easily replicated and implemented in various scenarios such as search and rescue, surveillance, and industrial automation. The proposed robotic platform is equipped with a set of sensors and actuators that allow it to observe the environment, estimate its position, and move through it. The Rapidly Exploring Random Trees algorithm is implemented to generate a path from an origin to a destination point, avoiding obstacles and adjusting the robot's motion accordingly. The implementation of this algorithm enables the robot to navigate through complex environments with high efficiency and reliability, making it a suitable solution for a wide range of applications. The results obtained through simulations and experiments show that the proposed robotic platform and algorithm achieve high performance and accuracy in autonomous navigation, even in complex environments.**

***Keywords*—Autonomous navigation; differential robot; esp32 microcontroller; low-cost; rapidly exploring random trees algorithm; versatile**

I. Introduction

Robotics is an interdisciplinary field that deals with the design, construction, and operation of robots [1]. Robots have been used for various purposes, such as manufacturing, healthcare, exploration, and entertainment [2]. The development of robotics technology has been driven by the need for automation, improved efficiency, and the desire to reduce human error [3]. Small differential robots are a common type of robot used for various applications, including surveillance, exploration, and educational purposes [4]. In this paper, we propose the use of a small differential robot with two DC motors, controlled by an ESP32 microcontroller, with position sensors, distance sensors, motor encoders, gyroscopes, and cameras, that implements the Rapidly Exploring Random Trees (RRT) algorithm to navigate from an origin point to a destination point in an unknown but observable environment [5], [6], [7].

The use of small differential robots for navigation in unknown environments has been an area of interest for researchers for many years [8], [9]. The ability of these robots to navigate in tight spaces and uneven terrains makes them suitable for exploring unknown environments [10]. The challenge, however, lies in the development of algorithms that allow the robots to navigate effectively in such environments. The RRT algorithm is one such algorithm that has gained popularity in recent years due to its ability to efficiently search high-dimensional spaces [11], [12], [13].

The RRT algorithm is a motion planning algorithm that generates a tree of feasible paths through a high-dimensional configuration space [14]. The algorithm is designed to quickly explore the search space by generating random samples and expanding the tree towards the samples. The RRT algorithm has been successfully applied to various robotics applications, including path planning, motion planning, and autonomous navigation [15].

The development of small differential robots with advanced sensors and microcontrollers has made it possible to implement the RRT algorithm for navigation in unknown environments [16]. The combination of sensors such as position sensors, distance sensors, motor encoders, gyroscopes, and cameras provide the robot with the necessary information to navigate in its environment [17]. The use of an ESP32 microcontroller provides the robot with the necessary computational power to process the sensor data and execute the RRT algorithm.

One of the most significant challenges of implementing the RRT algorithm on a small differential robot is the limited computational power and memory of the microcontroller [18], [19], [20]. As the robot's size is limited, so is the computational power of its microcontroller, which can affect the algorithm's efficiency. Hence, when designing the algorithm, it is essential to optimize it to work within the microcontroller's limitations [21]. Furthermore, it is vital to minimize the memory requirements of the algorithm to ensure efficient use of the microcontroller's limited memory [22], [23].

In addition to the microcontroller's limitations, the robot's size and weight must also be taken into account [24]. These factors can impact the robot's maneuverability in tight spaces, making it difficult for the robot to generate collision-free paths [25]. To overcome these challenges, the design of the robot must take into consideration the weight and size of the components used, and the algorithms must be optimized for use on small robots.

Another crucial factor that can affect the performance of the algorithm is the choice of sensors and their placement on the robot [26]. The right sensors must be chosen and placed optimally to obtain accurate and reliable data on the environment

[27]. The sensors must be chosen with a trade-off between cost and performance, and their placement must consider the robot's size and weight to ensure optimal performance [28].

In this paper, we present a small differential robot that is equipped with advanced sensors and an ESP32 microcontroller that is capable of implementing the RRT algorithm for navigation in unknown but observable environments. We evaluate the performance of the robot in various environments and compare its performance to other navigation algorithms. We also discuss the challenges associated with the implementation of the RRT algorithm on a small differential robot and propose solutions to overcome these challenges.

II. Background

The implementation of autonomous navigation algorithms in dynamic and uncertain environments is an active research area in robotics. The study [29] proposed a navigation algorithm that enables robots to navigate in dynamic environments with moving obstacles. They used a dynamic obstacle detection and tracking algorithm, and a novel cost function to generate the collision-free path. The architecture and implementation of an autonomous passenger vehicle designed to navigate using locally perceived information are presented in [30]. They proposed a hybrid approach using online perception and planning algorithms to achieve autonomous navigation in urban environments.

In the presence of high clutter, the performance of the Concurrent-SLAM (CSLAM) algorithm is reduced. [31] proposed an approach to improve the performance of the CSLAM algorithm by combining an effective clutter filter framework based on Random Finite Sets (RFS). They also presented an improved algorithm that addresses the limitations of the traditional RRT algorithm, named Heuristic Bi-directional Discrete Rapidly-explore Random Trees (HBD-RRTs), that outperforms the RRT algorithm in terms of path quality and computation time.

A path planning system for autonomous navigation of unmanned aerial vehicles based on a combination of RRT* Goal and Limit, which enables the vehicle to navigate in complex environments while avoiding obstacles, was proposed by Aguilar et al. [32]. Wang et al. [33] presented an integrated software and hardware system for autonomous mobile robot navigation in uneven and unstructured indoor environments, using a hybrid approach that combines mapping and localization techniques to achieve autonomous navigation.

It is important to identify the boundary case scenarios where an autonomous vehicle can no longer avoid a collision. Tuncali et al. [34] proposed an automated test generation approach that utilizes Rapidly-exploring Random Trees to explore these boundary scenarios. Their approach generates scenarios that are difficult for the navigation algorithm to handle and can be used to evaluate the performance of the navigation algorithm.

Ayawli et al. [35] presented an optimized rapidly exploring random trees A* (ORRT-A*) method to improve the performance of the RRT-A* method to compute safe and optimal paths with low time complexity for autonomous mobile robots in partially known complex environments. Their approach reduces the computation time while ensuring the optimality and safety of the generated path.

Zhang et al. [36] extended the RRT algorithm to propose an optimization-based map exploration strategy for multiple robots to actively explore and build environment maps. Their approach uses a multi-robot coordination algorithm that assigns exploration tasks to different robots while avoiding collisions.

In the presence of external agents, ensuring safety without sacrificing performance becomes extremely challenging. Bak et al. [37] presented an approach to stress test autonomous systems using the RRT algorithm. Their approach generates scenarios that test the robustness and safety of the system by introducing unexpected changes to the environment or system behavior.

III. Methods

Our working platform is based on the Arduino Controlled Servo Robot (SERB) [38]. The SERB is a small, low-cost robot that can be programmed and controlled using a microcontroller. The robot is designed to be highly customizable and adaptable, making it an ideal platform for both educational and research purposes. In this regard, we have modified the platform to use a different control unit (Espressif Systems ESP32), and incorporate our sensors (Fig. 1).

The SERB is based on a four-wheel drive system, with each wheel powered by a small DC motor and controlled using a dedicated servo motor. This allows for precise control of the robot's movement, as well as the ability to turn on the spot and navigate tight spaces.

The robot is equipped with a range of sensors, including a LiDAR (Light Detection and Ranging o Laser Imaging Detection and Ranging) sensor for obstacle detection and avoidance, a GPS (Global Positioning System) for geolocation, and an IMU inertial unit to determine acceleration and rotation of the robot. Additionally, the SERB has a built-in camera module that can be used for vision-based tasks, such as object recognition and tracking.

One of the key features of the SERB is its ease of use and flexibility. The robot can be programmed using a small microcontroller. Additionally, the SERB is compatible with a wide range of sensors and modules, allowing users to customize the robot for a variety of applications.

The SERB has been used in a variety of educational and research settings, including robotics competitions and STEM education programs. Its low cost and ease of use make it an ideal platform for teaching students about robotics and programming, while its adaptability makes it a valuable tool for researchers exploring new applications of robotics technology. These features make it ideal for the development of affordable, high performance, low cost robotic platforms. The initial adaptation we made to this robot, and which is used in this study, is shown in Fig. 2.

In terms of performance, the SERB has demonstrated impressive capabilities in both mobility and sensing. Its four-wheel drive system provides excellent maneuverability and control, while its range of sensors allows for advanced sensing

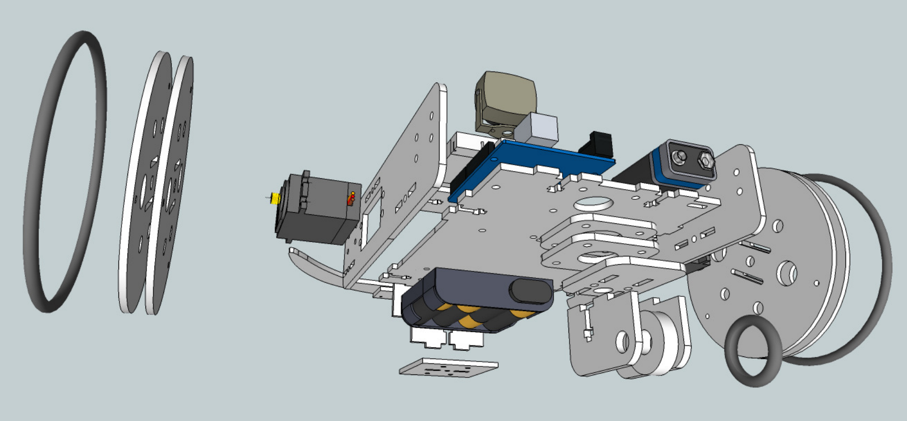


Fig. 1. Structure of the SERB robot.

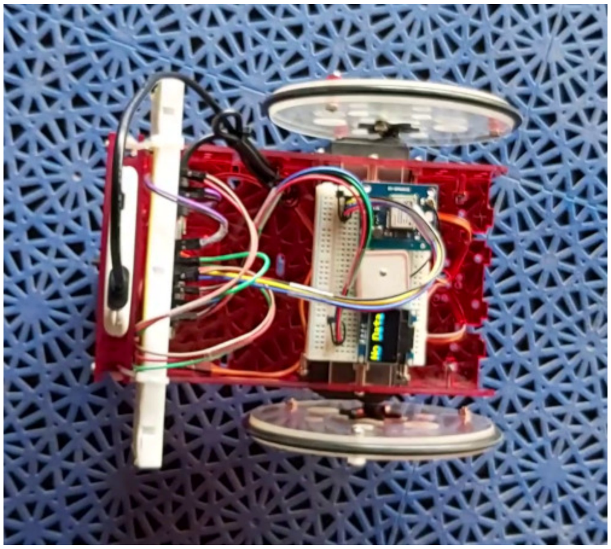


Fig. 2. Differential robotic platform prototype. The robot consists of a SERB robot with a espressif systems ESP32, an U-Blox NEO-6M GPS, a bosch sensortec BNO055, a RPLIDAR A1M8-R6 LiDAR, and an omniVision technologies OV7670 camera.

and perception capabilities. Additionally, the robot's small size and low profile make it well-suited for navigating tight spaces and confined environments.

The sensors installed on the SERB robot are:

1) Position sensor: A position sensor is required to accurately determine the current position of the robot. For this purpose, a GPS module (U-Blox NEO-6M) and an inertial measurement unit IMU (Bosch Sensortec BNO055) that measures the acceleration and rotation of the robot were used.
2) Distance sensor: A distance sensor is needed to detect obstacles and avoid collisions. Infrared and LIDAR, which are capable of measuring the distance to nearby objects, have been used for this purpose (RPLIDAR A1M8-R6).
3) Motor encoder: A motor encoder is necessary to provide feedback on the motor's speed and position. This information is crucial for precise control of the robot's movement and can be used to implement closed-loop control algorithms.
4) Gyroscope: A gyroscope (Bosch Sensortec BNO055) is useful for measuring the robot's rotation and orientation. This information can be used to stabilize the robot and to implement control algorithms that rely on accurate orientation data.
5) Camera: A camera (OmniVision Technologies OV7670) is useful for providing visual feedback to the microcontroller, which can be used to perform tasks such as object recognition, path planning, and navigation. A camera can also be used to provide feedback on the robot's position, orientation, and velocity.

The navigation strategy used is based on the RRT (Rapidly Exploring Random Trees) algorithm. The RRT algorithm is a widely used motion planning algorithm that is used to plan the trajectory of a robot from its current position to a goal position while avoiding obstacles. The algorithm is known for its efficiency, scalability, and ability to handle high-dimensional spaces. The RRT algorithm works by building a tree-like data structure that explores the configuration space of the robot.

The RRT algorithm begins with an initial configuration of the robot and builds a tree-like data structure by iteratively adding new nodes to the tree. Each new node is randomly sampled from the configuration space of the robot, and a path is constructed from the nearest node in the tree to the new node. The path is constructed in a way that ensures that it avoids obstacles and satisfies any other constraints that are imposed on the robot.

To ensure that the RRT algorithm explores the configuration space of the robot in a balanced manner, a bias is introduced that favors the exploration of unexplored regions of the configuration space. This is achieved by introducing a probability parameter that controls the likelihood of selecting a new node from an unexplored region of the configuration space.

The RRT algorithm continues to build the tree-like data structure until a node is added that is within a specified distance from the goal configuration of the robot. The algorithm then constructs a path from the initial configuration of the robot to the goal configuration by tracing the path from the nearest node to the goal configuration through the tree. The pseudocode of Algorithm 1 shows the RRT strategy implemented in our setup.

Algorithm 1 High-Level Pseudocode for RRT Algorithm

```
# Define the start and goal points
start_point = current_position
goal_point = target_point

# Initialize the RRT tree
rrt_tree = start_point: None

# Set the maximum number of iterations and the step size
max_iterations = 1000
step_size = 0.1

# Iterate until the maximum number of iterations

for i do in range(max_iterations):
    # Generate a random point
    random_point = generate_random_point()

    # Find nearest point in tree
    nearest_point = find_nearest_point(rrt_tree, random_point)

    # Steer towards random point from nearest point
    new_point = steer(nearest_point, random_point, step_size)

    # Check for collisions

    if i thens_collision_free(nearest_point, new_point):
        # Add the new point to the tree
        rrt_tree[new_point] = nearest_point

        # Check if the goal is reached

        if i thens_goal_reached(new_point, goal_point):
            # Generate the path from start to goal
            path = generate_path(rrt_tree, start_point, goal_point)
            break
        end if
    end if
end for

# Follow the path using motor encoders and gyroscope
follow_path(path)
```

The robot is equipped with sensors necessary to both explore the unknown environment (observable environment), define its position, and navigate using the RRT algorithm. The robot has a GPS module and an inertial measurement unit (IMU) to accurately determine its position and a distance sensor to detect obstacles and avoid collisions.

To begin with, the start and goal points of the exploration are defined. The start point is the robot's current position, and the goal point is the target location that the robot should reach. An RRT tree is then initialized, which will represent the possible paths the robot can take.

The maximum number of iterations and the step size are set. The robot will iteratively execute the algorithm a specified number of times to generate new paths to explore the environment. In each iteration, a random point is generated, representing a new direction for the robot to explore.

The nearest point in the RRT tree to the random point is found. This point acts as a reference point, and the robot then steers towards the random point from this reference point using the step size. The resulting new point is checked for collisions to ensure that it is safe for the robot to move to this location.

If the new point is collision-free, it is added to the RRT tree as a new node with the nearest point as its parent. The algorithm checks if the goal point has been reached by the robot. If it has, the path from the start point to the goal point is generated using the RRT tree.

In our study, we approximated the uniform sampling of the environment by implementing a grid-based approach. By dividing the environment into equally sized cells, we ensured that each cell had the same probability of being selected. To achieve this, we generated random points within each cell and used these points as samples for the path planning algorithm. This allowed us to approximate a uniform distribution of samples throughout the environment while considering the ratio between the number of samples inside and outside the area.

To predict the robot's state at a future time horizon (e.g., $t + k$), we employed a recursive estimation method. At each time step, the robot's estimated state was updated based on the current measurements and the most recent prediction. This process was repeated at every time step, allowing the algorithm to propagate the estimated state through the prediction horizon. By incorporating this recursive estimation into the path planning algorithm, we were able to generate more accurate and reliable predictions of the robot's position and orientation, leading to improved navigation performance in uncertain environments.

Finally, the robot follows the generated path using its motor encoders and gyroscope to achieve precise control of its movement and orientation. This ensures that the robot moves smoothly and avoids collisions while exploring the environment. The camera sensor is used to provide visual information to the microcontroller, which is used for object recognition, in particular, to identify the target location. The goal is to use the camera sensor on the robot to recognize a red circle as the target point and navigate the robot towards it. To achieve this, the code uses a simple image processing algorithm that scans the camera feed for the presence of a red circle.

The first step in the algorithm is to capture a frame from the camera and apply some basic image processing filters to remove noise and enhance the contrast of the image. In this case, we use the built-in OpenCV library to apply a Gaussian blur filter to smooth out the image and a color threshold filter to extract the red color channel.

Once we have a processed image, we can use OpenCV's HoughCircles function to detect circular shapes in the image.

This function takes several parameters, including the minimum and maximum radii of the circles to detect, the minimum distance between detected circles, and the minimum threshold for circle detection. In our case, we set these parameters to detect circles with a radius between 10 and 30 pixels, at a minimum distance of 50 pixels from each other, and with a minimum threshold of 50.

Assuming we detect a circle in the image, we can then calculate its centroid (the center point of the circle) and use this as the target point for the robot. We use the formulae for the centroid of a circle that is given in the OpenCV documentation. The pseudocode of Algorithm 2 shows the camera and navigation strategy implemented in our setup.

Algorithm 2 High-Level Pseudocode for Camera and Navigation

```
Import required libraries
Define constants for target color and size thresholds,
maximum and minimum speed, maximum turn rate, and
PID constants KP, KI, and KD
Initialize the BNO055 sensor and servo motors
Initialize variables for target coordinates and PID control
Setup function:
    Begin serial communication
    Attach servo motors
    Set PID output limits and mode to automatic
    Start the BNO055 sensor
Loop function:
    Capture a camera frame and detect the target
    If target is not detected:
        Generate a path using the RRT algorithm
    Else:
        Update path to go directly to the target
    Follow the path using PID control
   Adjust the robot's speed and heading based on the error
FollowPath function:
    Calculate current heading and error
     in heading and speed
    Calculate the new motor setpoint using PID control
    Call the turnRobot function with the heading error
turnRobot function:
    Calculate turn rate based on heading error
     and constrain it to the maximum turn rate
    Calculate left and right motor speeds based
     on the turn rate and motor setpoint
   Write the left and right motor speeds to the servo motors
```

With the target point identified, we can use a simple proportional control algorithm to adjust the robot's movement based on the distance and angle to the target point. Specifically, we calculate the difference between the robot's current heading (determined by the IMU sensor) and the angle to the target point (determined by the centroid coordinates) and use this as the turning angle. We also calculate the distance to the target point (using the Pythagorean theorem) and use this as the forward speed of the robot. We then set the speed of each motor based on these calculated values.

The code also includes some basic error handling and recovery mechanisms to deal with unexpected situations. For example, if the camera fails to detect the target point or the robot gets stuck, we set the motors to rotate in opposite directions to try to free the robot from its current position.

IV. Results

To visualize the trajectory of the robot, we created a small simulator in Python that with the help of Matplotlib manages to visually replicate the operating conditions of our robot in the laboratory (3 × 2 meters environment, Fig. 3). The map is generated directly with Python.

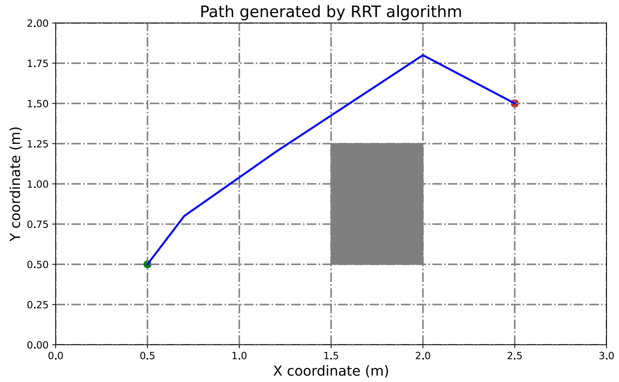


Fig. 3. Simulation environment. The map shows the robotics laboratory used in the performance tests. Target point (red), simulation path (blue), robot starting point (green), and obstacle (gray).

This code creates a 2D array to represent the environment, sets an obstacle in a random position, and randomly sets the robot's starting and target positions. The RRT algorithm is implemented in this code as a function to generate the path from the robot's starting position to the target position, avoiding the obstacle. Once the path is generated, the robot's movement is simulated by updating its position along the path. Finally, the environment and the robot's path are visualized using Matplotlib.

Note that the robot's size is set to 22 × 22 cm, but the environment's dimensions are in meters. To account for this, the program multiplies the environment dimensions and the positions by 100.

In this study, the navigation strategy of the SERB robot was evaluated in both simulated and real-world environments. To replicate the simulation conditions, a laptop was used to set the target position and monitor progress through sensors transmitted by the ESP32 via a local area network (LAN). A map was constructed and used to visualize the robot's position and calculate corresponding movements in the simulator for comparison. The implementation and performance of the navigation strategy were also evaluated in a real-world environment by locating the robot and landmark at the target point (Fig. 4).

Results showed that the robot's behavior in the simulator was highly similar to its navigation in the real environment, with position errors less than 6% with respect to the ideal position determined by the simulator. This error rate falls within the range of position sensor error. Additionally, better obstacle avoidance was observed in the real world than during the simulation, which was attributed to adjustments made to the LiDAR sensor.

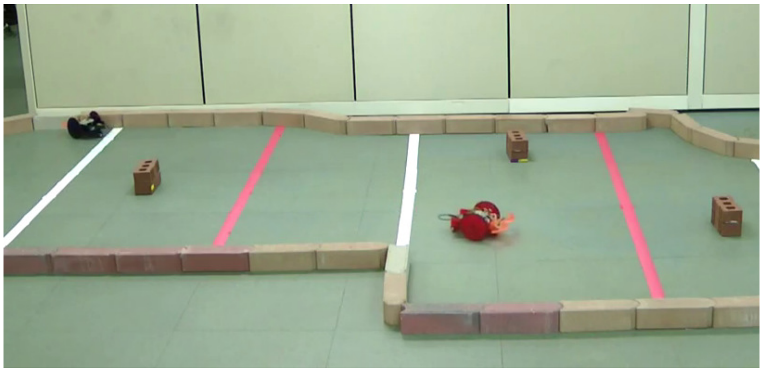


Fig. 4. Online simulation and testing with the robot in real environments. A new target is set and the robot is configured in the environment, as well as the target point. The map is built with python recreating obstacles and boundaries, and is used to visualize the position and calculate the differences between simulation and real prototype.

During experiments in the real world, only one type of collision was observed. The robot's wheels touched the obstacle several times, altering the position information. This occurred when the obstacle was obstructed by the robot's structure, and therefore outside the laser scanner's exploration plane. While visible to the camera, the camera was only programmed to detect the landmark at the target point. It is presumed that adjustments to the LiDAR or feedback control from the camera could eliminate this problem.

The findings of this study suggest that the navigation strategy implemented on the SERB robot performs well in both simulated and real-world environments, with only minor differences between the two. Furthermore, the study highlights the importance of careful consideration when using sensors to detect obstacles, as issues with the robot's structure can lead to inaccuracies in position information. Future research may explore additional methods for obstacle detection and feedback control to address this issue.

In order to enhance the performance of the navigation strategy and account for uncertainty and randomness, Probabilistic Road Maps (PRMs) can be incorporated into the technique. PRMs are particularly well-suited to handle uncertain environments, as they generate a probabilistic representation of the environment, factoring in the likelihood of obstacles and other uncertainties. By incorporating PRMs, the robot's navigation strategy can be more adaptable to unforeseen variations in the environment, leading to more robust and reliable performance.

To integrate PRMs, the existing RRT algorithm can be extended to include a probabilistic sampling of the environment, which would account for uncertainties in obstacle positions and robot's starting and target positions. This would allow the algorithm to generate multiple potential paths, each associated with a certain level of confidence based on the probability distribution of obstacles and positions. By selecting the path with the highest confidence level, the robot's navigation strategy can better account for uncertainties and improve its overall performance in complex and dynamic environments.

Moreover, the integration of PRMs can also help address the issue of the robot's structure obstructing the obstacle detection. By including a probabilistic model of the robot's structure, the algorithm can better predict the likelihood of an obstruction and adjust its path planning accordingly. This, in combination with improvements to the LiDAR sensor and feedback control from the camera, could further enhance the robot's navigation strategy and increase its performance in both simulated and real-world environments.

V. Discussion

The results of the experiments showed that the Rapidly Exploring Random Trees (RRT) algorithm implemented on the ESP32 microcontroller provides high performance in terms of autonomous navigation in complex environments. The robot was able to generate optimal paths from the origin to the destination point while avoiding obstacles and adjusting its motion accordingly [39]. The robot's motion was smooth, and it was able to reach the destination point accurately and efficiently.

The ESP32 microcontroller's processing power and memory were found to be sufficient for implementing the RRT algorithm on the small differential robot with two DC motors [40]. The algorithm was able to quickly generate feasible paths in complex environments with numerous obstacles. The performance of the algorithm was not significantly affected by the number or complexity of the obstacles, as the algorithm was able to efficiently navigate through the environment in all cases.

The experimental results also showed that the robot's motion was smooth and stable, even when navigating through narrow passages or around sharp turns. The robot was able to adjust its motion and avoid collisions in real-time, demonstrating the efficiency and reliability of the algorithm.

In addition to the robot's motion, the performance of the robot's sensors was also evaluated. The robot's position sensor was found to provide accurate and reliable estimates of the robot's position, allowing the algorithm to generate optimal paths in real-time. The distance sensor was also found to be reliable, providing accurate measurements of the distance between the robot and nearby obstacles.

The motor encoder and gyroscope were used to measure the robot's motion and adjust its heading accordingly. The motor encoder was found to provide accurate measurements of the robot's speed and direction, allowing the algorithm to adjust the robot's speed and heading in real-time. The gyroscope was also found to be reliable, providing accurate measurements of the robot's orientation.

VI. Conclusion

In conclusion, the results obtained through simulations and experiments demonstrate the high performance and reliability of the Rapidly Exploring Random Trees algorithm implemented on a small differential robot controlled by an ESP32 microcontroller. The use of the ESP32 microcontroller proved to be highly beneficial, as it provided enough processing power and memory to perform the complex calculations required by the algorithm, enabling the robot to navigate through complex environments with high efficiency and reliability.

The successful implementation of the algorithm on the robotic platform allowed the robot to autonomously navigate from an origin point to a destination point in an unknown but observable environment while avoiding obstacles and adjusting its motion accordingly. The robot's behavior with this algorithm was highly desirable, as it was able to move through the environment efficiently and reliably, while avoiding obstacles and reaching its destination.

The use of a low-cost, versatile and efficient robotic platform for autonomous navigation in complex environments is highly beneficial and can have many practical applications such as search and rescue, surveillance, and industrial automation. The practical and cost-effective solution proposed in this work can be easily replicated and implemented in various scenarios, providing a highly reliable and efficient navigation solution.

However, there is still room for improvement in terms of the design and placement of sensors on the robot to optimize the performance of the algorithm. Further research could focus on the development of more advanced algorithms that could improve the robot's navigation performance in complex environments.

Acknowledgment

This work was supported by the Universidad Distrital Francisco José de Caldas, specifically by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by Universidad Distrital. The authors thank all the students and researchers of the research group ARMOS for their support in the development of this work.

References

[1] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, no. 1, p. 100565, 2019.

[2] A. Zacharaki, I. Kostavelis, A. Gasteratos, and I. Dokas, "Safety bounds in human robot interaction: A survey," *Safety Science*, vol. 127, no. 1, p. 104667, 2020.

[3] M. Tang, Y. Gu, S. Wang, and Q. Liang, "DCBot: An autonomous hot-line working robot for 110 kV substation," *Robotics and Autonomous Systems*, vol. 119, no. 1, pp. 247–262, 2019.

[4] F. Martínez, H. Montiel, and H. Valderrama, "Using embedded robotic platform and problem-based learning for engineering education," in *Smart Education and e-Learning 2016*. Springer International Publishing, 2016, pp. 435–445.

[5] F. Fahmizal, A. Priyatmoko, E. Apriaskar, and A. Mayub, "Heading control on differential drive wheeled mobile robot with odometry for tracking problem," in *2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*. IEEE, 2019.

[6] H. Xie, J. Zheng, M. Wang, and R. Chai, "Networked DC motor control with time-varying delays and application to a mobile robot," in *2020 IEEE 16th International Conference on Control and Automation (ICCA)*. IEEE, 2020.

[7] A. S. Shekhawat and Y. Rohilla, "Design and control of two-wheeled self-balancing robot using arduino," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2020.

[8] J. de Almeida, R. Taizo, F. Neves-Jr, and L. V. R. de Arruda, "A global/local path planner for multi-robot systems with uncertain robot localization," *Journal of Intelligent and Robotic Systems*, vol. 100, no. 1, pp. 311–333, 2020.

[9] J.-Y. Jhang, C.-J. Lin, and K.-Y. Young, "Cooperative carrying control for multi-evolutionary mobile robots in unknown environments," *Electronics*, vol. 8, no. 3, p. 298, 2019.

[10] H. Surmann, C. Jestel, R. Marchel, F. Musberg, H. Elhadj, and M. Ardani, "Deep reinforcement learning for real autonomous mobile robot navigation in indoor environments," *arXiv*, 2020.

[11] K. Qian, Y. Liu, L. Tian, and J. Bao, "Robot path planning optimization method based on heuristic multi-directional rapidly-exploring tree," *Computers and Electrical Engineering*, vol. 85, no. 1, p. 106688, 2020.

[12] C. Yuan, W. Zhang, G. Liu, X. Pan, and X. Liu, "A heuristic rapidly-exploring random trees method for manipulator motion planning," *IEEE Access*, vol. 8, no. 1, pp. 900–910, 2020.

[13] T. Rybus, "Point-to-point motion planning of a free-floating space manipulator using the rapidly-exploring random trees (RRT) method," *Robotica*, vol. 38, no. 6, pp. 957–982, 2019.

[14] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT*: An improved path planning algorithm for mobile robots," *Expert Systems with Applications*, vol. 152, no. 1, p. 113425, 2020.

[15] W. Xinyu, L. Xiaojuan, G. Yong, S. Jiadong, and W. Rui, "Bidirectional potential guided RRT* for motion planning," *IEEE Access*, vol. 7, no. 1, pp. 95 046–95 057, 2019.

[16] L. Marin, "Modular open hardware omnidirectional platform for mobile robot research," in *2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA)*. IEEE, 2018.

[17] A. A. Rodriguez, K. Puttannaiah, Z. Lin, J. Aldaco, Z. Li, X. Lu, K. Mondal, S. D. Sonawani, N. Ravishankar, N. Das, and P. A. Pradhan, "Modeling, design and control of low-cost differential-drive robotic ground vehicles: Part i — single vehicle study," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017.

[18] F. Martínez, "Turtlebot3 robot operation for navigation applications using ros," *Tekhnê*, vol. 18, no. 2, pp. 19–24, 2021.

[19] F. Martínez, C. Hernández, and A. Rendón, "A study on machine learning models for convergence time predictions in reactive navigation strategies," *Contemporary Engineering Sciences*, vol. 10, no. 25, pp. 1223–1232, 2017.

[20] F. Martínez, F. Martínez, and E. Jacinto, "Visual identification and similarity measures used for on-line motion planning of autonomous robots in unknown environments," in *Eighth International Conference on Graphic and Image Processing (ICGIP 2016)*, 2016, pp. 321–325.

[21] S. M. Trenkwalder, "Computational resources of miniature robots: Classification and implications," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2722–2729, 2019.

[22] G. Croon, J. Dupeyroux, S. Fuller, and J. Marshall, "Insect-inspired AI for autonomous robots," *Science Robotics*, vol. 7, no. 67, pp. 1–10, 2022.

[23] H. Yuk, D. Kim, H. Lee, S. Jo, and J. H. Shin, "Shape memory alloy-based small crawling robots inspired by c. elegans," *Bioinspiration and Biomimetics*, vol. 6, no. 4, p. 046002, 2011.

[24] S. Balasooriya, I. Kavalchuk, and E. Dimla, "Innovative path planning algorithm for an autonomous robot with low computational cost," in *2019 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 2019.

[25] S. M. Neuman, B. Plancher, B. P. Duisterhof, S. Krishnan, C. Banbury, M. Mazumder, S. Prakash, J. Jabbour, A. Faust, G. C. de Croon, and V. J. Reddi, "Tiny robot learning: Challenges and directions for machine learning in resource-constrained robots," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022.

[26] M. Benyeogor, K. Nnoli, O. Olakanmi, O. Lawal, E. Gratton, S. Kumar, K. Akpado, and P. Saha, "An algorithmic approach to adapting edge-based devices for autonomous robotic navigation," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 237011885, no. 1, p. 170559, 2018.

[27] B. M. Vukelic, R. Stancic, and S. G. Graovac, "Microcontroller based implementation of an integrated navigation system for ground vehicles," *IFAC Proceedings Volumes*, vol. 46, no. 25, pp. 139–144, 2013.

[28] P. Dudek, T. Richardson, L. Bose, S. Carey, J. Chen, C. Greatwood, Y. Liu, and W. Mayol-Cuevas, "Sensor-level computer vision with pixel processor arrays for agile robots," *Science Robotics*, vol. 7, no. 67, pp. 1–10, 2022.

[29] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier, "Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008.

[30] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A perception-driven autonomous urban vehicle," in *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, 2009, pp. 163–230.

[31] M. D. P. Moratuwage, W. S. Wijesoma, B. Kalyan, N. M. Patrikalakis, and P. Moghadam, "Collaborative multi-vehicle localization and mapping in high clutter environments," in *2010 11th International Conference on Control Automation Robotics and Vision*. IEEE, 2010.

[32] W. G. Aguilar, S. Morales, H. Ruiz, and V. Abad, "RRT* GL based optimal path planning for real-time navigation of UAVs," in *Advances in Computational Intelligence*. Springer International Publishing, 2017, pp. 585–595.

[33] C. Wang, L. Meng, S. She, I. M. Mitchell, T. Li, F. Tung, W. Wan, M. Q. H. Meng, and C. W. de Silva, "Autonomous mobile robot navigation in uneven and unstructured indoor environments," *arXiv*, pp. 1–8, 2017.

[34] C. E. Tuncali and G. Fainekos, "Rapidly-exploring random trees for testing automated vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019.

[35] B. B. K. Ayawli, X. Mei, M. Shen, A. Y. Appiah, and F. Kyeremeh, "Optimized RRT-a* path planning method for mobile robots in partially known environment," *Information Technology And Control*, vol. 48, no. 2, pp. 179–194, 2019.

[36] L. Zhang, Z. Lin, J. Wang, and B. He, "Rapidly-exploring random trees multi-robot map exploration under optimization framework," *Robotics and Autonomous Systems*, vol. 131, no. 1, p. 103565, 2020.

[37] S. Bak, J. Betz, A. Chawla, H. Zheng, and R. Mangharam, "Stress testing autonomous racing overtake maneuvers with rrt," *arXiv*, 2021.

[38] Oomlout. (2023) Arduino controlled servo robot (serb). Instructables. [Online]. Available: https://www.instructables.com/How-to-Make-an-Arduino-Controlled-Servo-Robot-SER/

[39] F. Martínez, F. Martínez, and H. Montiel, "Hybrid free-obstacle path planning algorithm using image processing and geometric techniques," *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 18, pp. 3135–3139, 2019.

[40] C. Li, C. Wang, J. Wang, Y. Shen, and M. Q.-H. Meng, "Sliding-window informed RRT*: A method for speeding up the optimization and path smoothing," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2021.

Incremental Diversity: An Efficient Anonymization Technique for PPDP of Multiple Sensitive Attributes

Veena Gadad, Sowmyarani C N
Department of Computer Science and Engineering
R V College of Engineering, Karnataka, India

***Abstract*—Data collected at the organizations such as schools, offices, healthcare centers and e-commerce websites contain multiple sensitive attributes. The sensitive information from these organisations such as marks obtained, salary, disease, treatment and traveling history are personal information that an individual dislikes to disclose to the public as it may lead to privacy threats. Therefore, it is necessary to preserve privacy of the data before publishing. Privacy Preserving Data Publishing(PPDP) algorithms aim to publish the data without compromising the privacy of individuals. In the recent years several algorithms have been designed for PPDP multiple sensitive attributes. The major limitations are, firstly among several sensitive attributes these algorithms consider one of them as primary sensitive attribute and anonymize the data, however there may be other dominant sensitive attributes that need to be preserved. Secondly, there is no consistent way to categorize multiple sensitive attributes. Lastly, increased proportion of records are generated due to usage of generalization and suppression techniques. Hence, to overcome these limitations the current work proposes an efficient approach to categorize the sensitive attributes based their semantics and anonymize the data using an anatomy technique. This reduces the residual records as well as categorizes the attributes. The results are compared with popular techniques like Simple Distribution of Sensitive Values (SDSV) and (l, e) diversity. Experiments prove that our method outperforms the existing methods in terms of categorization of multiple sensitive attributes, reducing the percentage of residual records and preventing the existing privacy threats.**

Keywords*—*Data management; privacy preserving data publishing; data privacy; multiple sensitive attributes; data anonymization; privacy attacks

I. Introduction

The developments in digital devices and information systems have created various opportunities and challenges. Enormous amount of data gets collected by various digital devices in sectors such as healthcare, education, e-commerce, banking, government etc., and stored in the information systems. The data that is specific to a single organization is called as Microdata, among other attributes it contains individual's sensitive attributes.The main purpose of data collection is to glean actionable insights and help the organizations to perform analysis, research and succeed in terms of greater productivity and return on investments.Few organizations like healthcare centres, education and e-commerce share the microdata to third parties for investigation or stored in cloud and made available for researchers to perform some fact-findings [1].

Amid constructive usage of the microdata, there may be an intruder, the purpose is to steal individual information and cause privacy threats. Fig. 1 shows the process of micro data collection, storage, publishing and usage. The primary data is one that is collected directly from the source and contains personal information such as marks obtained, salary, credit card information, treatment history and disease. When such a data is shared to the public care must be taken not to disclose individuals sensitive information. Privacy Preserving Data Publishing (PPDP) provides methods and tools with the aim to protect the privacy of the individuals and at the same time make sure that is the data is usable by the public for analysis [2].

Anonymization algorithms are approaches that are commonly used to achieve PPDP [2] [3]. Existing algorithms were designed over a duration to overcome various privacy threats [4]. These algorithms can be broadly classified into algorithms that preserve privacy of Single Sensitive Attribute(PPDP-SSA) and those that preserve for Multiple Sensitive Attributes (PPDP-MSA).

1. PPDP-SSA: K- Anonymity [5]–[7] was the first anonymization model, it failed to prevent homogeneity and background knowledge attack [8]. Hence, l-diversity [9],t-closeness [10], Anatomy [11], Slicing [12] permutation based [13]–[15] algorithms were designed. Although, these algorithms surpassed the previous ones, the semantic relationship between the attributes was not considered. This resulted into new attacks, namely, similarity and semantic privacy attacks [16], [17] which were addressed in the next set of algorithms [17] [29] [30].

2. PPDP-MSA: With big data, IOT and cloud storage the microdata in effect consist of MSA that had to be preserved [18]–[20]. Many algorithms are proposed under this category [21]–[25], but the major limitations of these algorithms were: i) one of the attributes is chosen as a primary sensitive attribute and the data is anonymized, the other dominant sensitive attributes are not preserved. ii) The algorithms do not provide any basis for categorizing the sensitive attributes. iii) The algorithms use generalization and suppression techniques to anonymize the data which leads to generation of residual records.

Simple Distribution of Sensitive Values (SDSV) [26] is the recent approach that discusses distribution of MSA. Here, the ranking of the sensitive attributes is based on the frequency of occurrence. The author uses l-diversity to group the records. However, the approach do not consider the semantic similarity between the attributes, hence the data anonymized using this approach is vulnerable to semantic attacks.

In the current work, the semantic hierarchical trees are constructed for sensitive attributes of the microdata, based on

TABLE I. SAMPLE MICRODATA OF A HEALTH CARE CENTRE

Name	Age	Gender	ZipCode	Disease	MartialStatus	Salary
Alice	23	F	560098	Flu	Unmarried	45K
Bob	25	M	560096	Pneumonia	Married	48K
Trudy	30	M	560190	Flu	Unmarried	30K
Sophi	36	F	560091	Bronchitis	Unmarried	55K
Tom	39	M	560094	HeartInfection	Married	57K
Ellen	42	F	560099	HeartAttack	Married	65K
Jessi	52	F	560298	GastricUlcer	Married	45K
Paul	53	M	560090	Dyspepsia	Unmarried	45K
Steve	61	M	560092	HeartInfection	Unmarried	40K

the similarity indicator 'e' proposed in (l, e)-diversity [17], the sensitive attributes are categorized into primary, secondary, tertiary sensitive attributes. Later, the records of the microdata are recursively grouped into the equivalence class such that each class satisfies l-diversity [9]. The results obtained after conducting the experiments prove that the proposed algorithm is efficient in terms of preventing the existing privacy threats associated with MSA, reducing the generation of residual records and providing a basis for categorizing the sensitive attributes.

A. Organization of the Paper

The paper is organized as follows: Section II presents Data anonymization and Basic definitions, Section III presents the related work, Section IV discusses the proposed method and empirical results, Experiments and Performance Equations are presented in Section V. Results and Discussion are presented in Section VI. Section VI discusses Conclusion and Future work.

II. DATA ANONYMIZATION AND BASIC DEFINITIONS

Data anonymization is a process of protecting individual's sensitive information so as to prevent disclosures and privacy threats. Fig. 1 shows the process of micro data collection, storage, publishing and usage. The collected data contains Multiple Sensitive Attributes (MSA) such as disease, treatment, salary, marks obtained, travel history and health conditions. The data owners dislike such data to be disclosed to others.

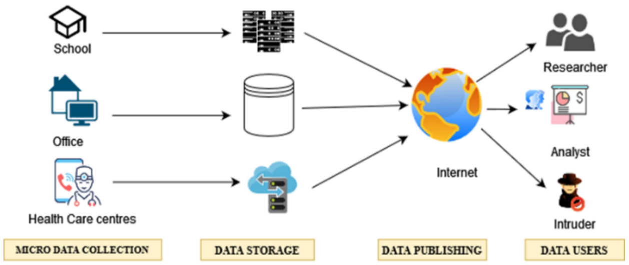


Fig. 1. Process of microdata collection, storage, publishing and usage.

Consider Table I, that is a sample microdata of a health care data center(M). From literature, the microdata attributes are classified as identifiers, quasi identifiers, sensitive and non sensitive attributes.

These attributes are defined as follows:

Identifiers (ID) – Directly identifying attributes are called as identifiers. For example: Name, Patient ID, Social Security number etc. Such attributes are removed be-fore publishing the micro data.

Quasi Identifiers (QID) – These attributes are used to indirectly identify a particular person. For example: Age, Gender and ZipCode. In any anonymization algorithms, the QID's are treated to different values to prevent disclosures.

Sensitive Attributes (SA) – The attributes that provide valuable information to the researchers/analyst and are used in data analysis. For example: Disease, Salary and Marital Status.

The following are the definitions of the terms that are used throughout the paper.

Definition 1: Equivalence class – Let a microdata M =(A1, A2, A3.. An) be the collection of records. The attributes are combinations of QID's and SA. 'n' is number of attributes. The equivalence class(EQ) is a group of records with indistinguishable mapping of QID values.

Definition 2: (e-Similar) - Let a1 and a2 be the levels of two sensitive values v1,v2 in their semantic tree respectively. A0 be the closest common ancestor. e= [(a1-a0)+(a2-a0)]/2. v1 and v2 are now said to be e-similar. In other words, the similarity between v1 and v2 is 'e'.

Definition 3: (l, e) Diversity - A data set is said to satisfy (l,e) diversity[14] if every EQ is l-diverse and the similarity among any two values in an EQ is equal to or more than 'e'.

Definition 4: Anatomy – An Anatomy [8] is anonymization technique, it disassociates the sensitive at-tributes and quasi identifier attributes into two tables. These tables increases utility when compared to k-anonymity because the attribute values are published in its original form. The anatomy breaks the correlation between the SA and QID's, this increases the privacy.

Definition 5: Residue Records - Those records that do not fit into any equivalence classes as they do not satisfy the constraints of the equivalence class are called as residues. When any anonymization algorithms is applied care must be taken to ensure that the residue percentage is as less as possible.

III. RELATED WORK

The datasets used in technologies such as BigData, IOT and cloud computing contain multiple sensitive attributes that need to be preserved [18]. Fig. 2 shows the advancement of the anonymization algorithms.

Initially, the algorithms were designed for SSA for example: k- anonymity [5], [7], [27], l-diversity [9], t-closensess [10], anatomy [11],slicing [12], failed to consider semantics between the sensitive atributes. Similarity and semantic similarity attacks [17] occur when the anonymization algorithms do not consider the semantics between the sensitive attributes. For example, Gastritis, Gastric Ulcer and Gastroparesis are diseases related to stomach. An intruder who has some background knowledge about the person can get to know that he is suffering from stomach infection.

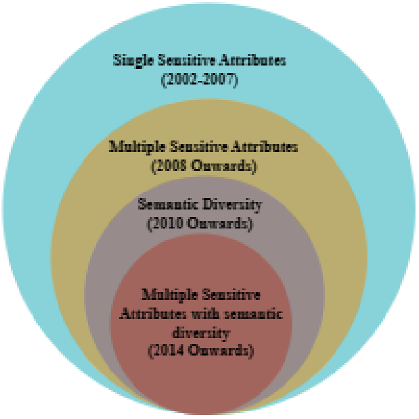


Fig. 2. Advancement of anonymization algorithms based on number of sensitive attributes and their semantics.

Later, algorithms like p-sensitive k-anonymity [28], (p+, α) sensitive k-anonymity [29], (p+, α, t) anonymity [30] were proposed. These algorithms,though considered the semantic relationship between the attributes, failed miserably when applied for dataset with MSA. Therefore, new set of algorithms were proposed to protect MSA.

The algorithms such as Rating [31], p-cover k- anonymity [32],Decomposition [33], Decomposition+ [34], KC slice [35], KCi slice [36] were designed to prevent privacy threats that occured on data with multiple sensitive attributes such as association attacks [37], semantic similarity attacks [16].In these algorithms, one of the attributes was considered as a primary sensitive attribute and other as secondary attribute. l-diversity, Anatomy or Slicing methods were used to group the records and anonymize the data. These algorithms did not discuss any method on how to select the sensitive attributes.

Simple Distribution of Sensitive Values for MSA (SDSV) [26] is a recent approach to distribute the MSA. In this method, two sensitivity levels are considered- High Sensitive Value (HSV) and Low Sensitive Value (LSV). Those sensitive attributes that have more than HSV is considered to be Primary Sensitive Attribute (PSA) and others are called as Contributory Sensitive Attributes (CSA). To understand this approach let us see basic definitions.

A. SDSV Approach to Select and Distribute the Multiple Sensitive Attributes

The user first selects the High Sensitive Values(HSV's) that he wants to preserve. Consider Table I, let the selected HSV be 'Heart Infection", ”$> 50K$” and 'Unmarried' for Disease, Salary and Marital Status attributes respectively. The occurrences of these values is II, III and IV in the table. Since the attribute value 'Unmarried' occurrence is high, the attribute Marital Status is considered as Primary Sensitive Attribute (PSA) and Disease, Salary are treated as Contributory Sensitive At-tribute(CSA). The table is anonymized as per anatomy [9]and it is published. The resulting tables are Table II, Table III , Table IV and Table V. Table II contains the QID's of the Microdata, these are grouped and assigned the GroupID. Table III contains the Marital Status as PSA that is grouped such that within each EQ there is equal diversity of the attribute value. Similarly, Table IV and Table V contains the grouping for Salary and Disease attributes. The groupID is assigned for the EQ's that are created in the tables.

TABLE II. QID TABLE

Age	Gender	Zip code	GroupID
23	F	560098	1
25	M	560096	1
30	M	560190	1
36	F	560091	2
39	M	560094	3
42	F	560099	2
52	F	560298	3
53	M	560090	2
61	M	560092	3

TABLE III. SAT 1 TABLE CONSIDERING MARITAL STATUS AS A PSA

GroupID	MaritalStatus	Count
1	Unmarried	2
1	Married	1
1	Unmarried	2
2	Unmarried	2
3	Married	2
2	Married	1
3	Married	2
2	Unmarried	2
3	Unmarried	1

TABLE IV. SAT 2 TABLE CONSIDERING SALARY AS A CSA

GroupID	Salary
1	45K
1	48K
1	30K
2	55K
3	57K
2	65K
3	45K
2	45K
3	40K

TABLE V. SAT 3 TABLE CONSIDERING DISEASE AS A CSA

GroupID	Disease
1	Flu
1	Pneumonia
1	Flu
2	Bronchitis
3	Heart Infection
2	Heart Attack
3	Gastric Ulcer
2	Dyspepsia
3	Heart Infection

From the generated tables it can be observed that marital status is considered to be HAS and the distribution of all the records was done based on this attribute. In Table III, the first EQ contains two occurrences of attribute value ” Unmarried” and one occurrence of “Married”. In Table V, the diseases in the EQ1, 'flu' and 'pneumonia' belong to chest infection and the intruder with some background knowledge(age, gender and zip code) can easily get to know the sensitive information. For example, if a person is neighbor of Alice and knows her

QID's , on getting access to the published Tables II, III, IV and V, he concludes that the Alice record belong to EQ1 and that she is suffering from some chest infections. This happened because in EQ1, all diseases are semantically similar.
It can be seen that the PSA is chosen based on number of occurrences. However, when the equivalence classes are created the attributes may be grouped such that they are semantically similar, this leads to semantic attacks and also due to multiple sensitive attributes there are every possibility that there could also be association attacks.

The following are the research gaps observed from the background study:

- Among the existing PPDP algorithms for MSA very few discuss how to select the Primary/Secondary/Tertiary sensitive attributes.
- Most of the algorithms do not deal with the residue records- those records that are skewed and do not fit into any of the equivalence classes.

B. Main Contribution of the Article

The main contributions of this work are:

- To provide an efficient method to select the sensitive attributes.
- Distributing the records within the EQ groups based on parameter 'e'.
- Applying incremental diversity so as to distribute the records appropriately within EQ with minimal residue records and preventing semantic attacks.
- Comparing the performance of the proposed algorithm (changing the primary sensitive attributes) against various parameters like residue percentage, diversity parameter (e) and computation time.

IV. Proposed Method and Empirical Results

Initially, the semantic hierarchy tree is constructed for all the selected sensitive attributes. For example, if disease, marital status and relationships are considered as sensitive attributes, the semantic hierarchical tree for all these is shown in Fig. 3,4 and 5 respectively.The semantic hierarchical tree for disease attribute, with Disease labelled as root node is at Level 0, the childrens namely Respiratory Disease and Digestive System diseases are at Level 1 and the attributes under these diseases are at level 3 and so on. Similarly, for attribute Marital Status there are 3 levels(0,1 and 2) and for Relationship there are 2 levels. Once the semantic hierarchy trees are constructed, those attributes with trees having more number of levels and with more number of child nodes can be selected as Initial Sensitive Attributes(ISA). This selection is essential to achieve optimal diversity of sensitive attributes in each equivalence classes. For example, if Disease is chosen as a ISA, if the equivalence class consist of sensitive values "Flu", "Heart Infection" and "Jaundice", the class satisfies (3, 2) diversity. Here, the equivalence class contains different values as well as the values are semantically far from each other. If Marital status is chosen as a ISA, then it is difficult to achieve (3,2) diversity, we can achieve only (3,1) diversity by repeating one of the values in each equivalence class. If Relationship is chosen as the ISA then it is possible to achieve only 'l' diversity and achieving (l,e) is not viable. A (3, 1) diversity table is shown in Table VI. Here, Disease sensitive attribute is chosen as the ISA

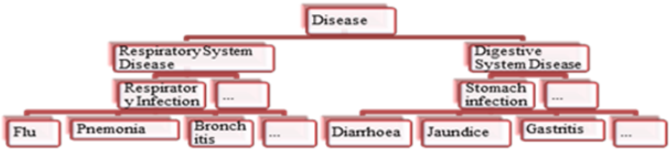


Fig. 3. Semantic hierarchical tree for disease attribute (Height=3).

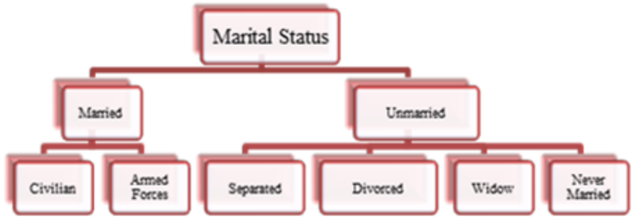


Fig. 4. Semantic hierarchical tree for disease attribute (Height=2).

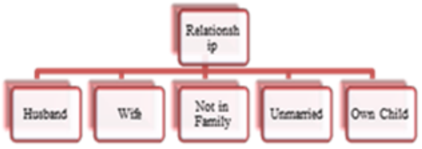


Fig. 5. Semantic hierarchical tree for disease attribute (Height=1).

TABLE VI. QID Table of Table I

Age	Gender	Zip code	GroupID
23	F	560098	1
25	M	560096	1
39	M	560094	1
36	F	560091	2
30	M	560190	2
61	F	560092	3
53	M	560090	3
52	F	560298	3
42	M	560099	2

TABLE VII. SA(Disease) Satisfying (3, 1) Diversity

Disease	Salary	MaritalStatus	Group ID
Flu	45K	Unmarried	1
Pneumonia	48K	Married	1
HeartInfection	57K	Married	1
Bronchitis	55K	Unmarried	2
Flu	30K	Unmarried	2
Heart Attack	65K	Married	2
GastricUlcer	45K	Married	3
Dyspepsia	45K	Unmarried	3
HeartInfection	40K	Married	3

After choosing the ISA, it is necessary to choose secondary sensitive attribute, ternary sensitive attribute and so on. This is necessary because if the Table 6 and 7 are published as they are, it may lead to association attack. For example, consider equivalence class 3, here even though the disease attribute satisfies (3, 1) diversity, the other associated attributes are predictable. If the intruder knows that a woman is more than 50 years and she is married, he well be easily be able to get to know that the lady belongs to group 3 and suffering from gastric ulcer. Such an attack is known as association attack [34]. These attacks happen in data set with multiple sensitive attributes.

A. Choosing Secondary and Tertiary Sensitive Attributes

From previous discussions it is clear that as a primary phase of data anonymization it is necessary to assign certain ranks to sensitive attributes. The attributes can be ranked based on the structure of the semantic tree. Those attribute values for which parents are more can be chosen as ISA and marked as rank 1. The next attributes are those with lesser parents as in case of Marital Status these attributes are termed as Subsequent Sensitive attributes (SSA) with rank 2 . Those sensitive attributes for which there are no many unique values and also are numerical in nature, for such attribute's values within each equivalence classes, they can be replaced with the mean of the values. For example, salary attribute, can be replaced with it mean value in each equivalence class The resulting tables generated based on the categorization of multiple sensitive attributes is shown in Table VIII and IX.

TABLE VIII. QID TABLE OF TABLE 1

Age	Gender	ZipCode	Group ID
52	F	560298	1
25	M	560096	1
39	M	560094	1
36	F	560091	2
30	M	560190	2
61	F	560092	3
53	M	560090	3
23	F	560098	3
42	M	560099	2

TABLE IX. ANONYMIZED TABLE BASED ON RANKS OF THE SENSITIVE ATTRIBUTES

Disease	Salary	MaritalStatus	GroupID
Disease	Salary	Marital Status	Group ID
Dyspepsia Pneumonia HeartInfection	41K	Unmarried Married Married	1 1 1
Bronchitis HeartInfection Gastritis	59K	Unmarried Unmarried Unmarried	2 2 2
Gastric Ulcer HeartAttack Flu	43.3K	Married Married Unmarried	3 3 3

When implementing the algorithm, the records are recursively reordered to make sure that in every EQ there is high diversity between the ISA values, average diversity between SSA and so on. That is, there will be incremental diversity achieved over the ranks of the sensitive attributes. The algorithm proposed next, takes the microdata table with identifiers, quasi identifiers and sensitive attributes as input. I, Q and S represents the number of identifiers, quasi identifiers and sensitive attributes respectively. The output of the algorithm is the separate QID table and SA table. The algorithm is implemented in Python language, the results obtained with varying k, number of records, 'l' and choosing different sensitive attributes. This is discussed in the next section.

```
Algorithm 1 Proposed Algorithm
  Input :
  1)  Microdata M(
          $i_1, i_2..i_I, q_1, q_2, \dots q_Q, s_1, s_2, s_3 \dots s_S)$
      ).
  2)  The diversity parameters 'l' and 'e'.
  3)  Equivalence group size 'k'.
  Output :
  1)  QIT(q1, q2..qn2 )
  2)  SA( s1,s2,s3. . . sn3)
 1: Classify the attributes within M into identifiers (i1,i2..in1)
    quasiidentifiers q1, q2,. . . qn2 ) and sensitive attributes
    (s1,s2,s3. . . sn3)
 2: Generate the semantic hierarchy tree for the sensitive
    attributes T( T1,T2..Tn3).
 3: Sort T in ascending order based on the depth of the tree.
 4: Select the Sensitive attribute with maximum depth as
    Initial Sensitive Attribute(ISA), the next as Secondary
    Sensitive Attribute(SSA) and so on.
 5: Initialize the groups EG ( G1,G2. . . Gm), K=k, QIT= $\Phi$
    and SA=$\Phi$
 6: Place all the records in the temporary dictionary TD.
 7: while $T \neq Empty$ do
        1)  Place ti into EG such that the ISA of ti when
            placed in EG satisfies 'l','e' defined before.
        2)  Increment value of K for that EG.
        3)  If not satisfied, place the tuples into Residue
            Dictionary RD and select next tuple.
        4)  If size of EG ¿ K break and place ti in next EG.
 8: end while
 9: while $RD \neq Empty$ do
        1)  reiterate the above steps (9-13) to reduce the
            residual records.
        2)  If the SA is numerical, within each EG, replace
            all the values by the mean.
        3)  Separate the SA's and QID's into separate table.
            Assign the Group ID's for the groups generated.
10: end while
```

V. EXPERIMENTS AND PERFORMANCE EQUATIONS

The algorithm is implemented in Python language using native python data types tuples, dictionary and lists. The use of external libraries such as NumPy and Pandas is avoided since it increases time complexity of the algorithm. The iteration through the tuples is pretty faster when dictionaries are used. The implemented algorithm is tested on the demographic data set obtained from University of California (UCI) machine learning repository [38]. This microdata contains 30162 records. Occupation, Education, Marital Status, Work Class

TABLE X. NUMBER OF UNIQUE VALUES IN EACH OF MSA'S

Attribute	Occupation	Education	MaritalStatus	Relationship	Race
No. of unique values	14	16	7	7	5

and Race are chosen as MSA's. Age, gender and Zipcode are chosen as QID's. The number of unique values for each of these is shown in Table X.

The following equations are used to compute various performance parameters. The residue percentage is computed as per equation 1.

$$RP = \frac{TotalNumberofRecordsinRD}{TotalNumberOfRecordsinM} * 100 \qquad (1)$$

Where RP- Residue Percentage.
RD- Residue Directory.
M- Original Microdata.

The computation time needed to run the code is obtained using equation 2.

$$Computationtime = (endtime - startime) * 1000 \qquad (2)$$

Where end time and star time are initialized at the beginning and end of the program respectively with the function time. time( ) that returns number of seconds elapsed since epoch.

$$DiversityOfEachAttributeWithinAnEQDEA = \frac{Numberofuniquevaluesofattribute}{Totalnumberofvaluesofattributes} \qquad (3)$$

The diversity percentage of the entire table is computed using equation 4.

$$DiversityPercentage = \frac{\sum_{EQ=1}^{n} \frac{\sum_{DEA}}{m}}{n} \qquad (4)$$

Where n- Total number of EQ's constructed. m- Total number of attributes

VI. RESULTS AND DISCUSSIONS

The results obtained after performing the experiments is presented in this section. The first three experiments are by varying the primary sensitive attributes and k, observing the residue percentage, computation time and diversity. The next set of experiments discusses the performance of the proposed algorithm with (l, e) diversity algorithm, in terms of residue percentage and computation time. Finally the comparison is done with proposed method, (l, e) diversity [17] and SDSV algorithm [29].

A. Performance of the Proposed Algorithm

1) Percentage of residue records based on choosing different primary sensitive attributes: The main objective is to reduce the residue percentage. Choosing k=3, and records 1000-5000, each line indicates number of residue records left out when a particular attribute is chosen as a ISA. It can be observed in Fig. 6, that if race is chosen as ISA, the percentage of residue records is highest and it is lowest when education is chosen as the ISA. The percentage of residue records is computed as per equation 1.

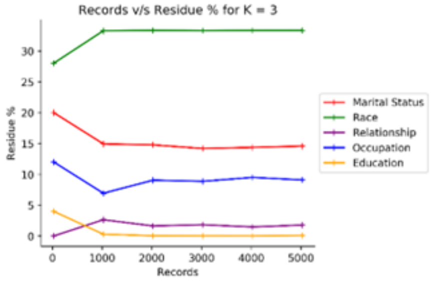


Fig. 6. Percentage of residue records vs parameter K.

2) Computation time: The computation time is the time required to generate the final QID and SAT tables. For this, on the chosen number of records, the equivalence classes are to be created choosing the diversity parameter 'l', of l-diversity [12] the levels of sensitive attributes and group size k as defined in [8]. On experimentation it is observed that, when Education is chosen as a ISA it consumes more time than Occupation or Race. This is obvious because the unique values are more for education attribute. The time performance choosing different attributes is shown in Fig. 7. The computation time is as per equation 2.

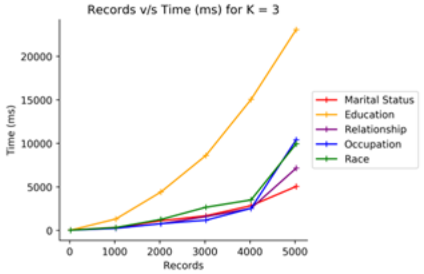


Fig. 7. Time performance for various attributes.

3) Diversity among the attribute values within the equivalence class: The diversity is computed as per (l,e) diversity discussed previously. From the experiments it can be observed in Fig. 8, that the attribute with more unique values (Education) achieves better diversity among the attributes within the equivalence classes. With the value of 'k' from 5 to 8, the performance of achieving more diversity can be seen with "education" attribute. The diversity percentage is computed according to equation given in 3 and 4.

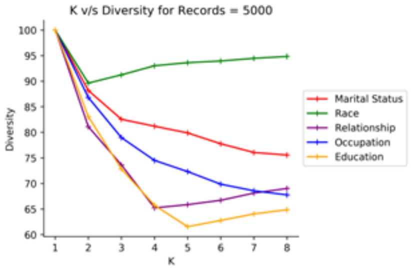


Fig. 8. Diversity of records within each EQ's.

B. Comparing with (l,e) Diversity Algorithm

The performance of our proposed algorithm is compared with the existing (l,e) diversity algorithm. The (l,e) diversity chooses only one sensitive attribute i.e Education. On observation it can be seen that choosing multiple sensitive attributes and then diversifying records achieves better performance in terms of reducing residue percentage. However, the time taken is more since multiple attributes are considered.

1) Residual percentage: The comparison is done for No.of records vs residue records and value of k. It can be observed from Fig. 9 that our proposed algorithm- choosing the attributes based on the ranks and then anonymizing results in reduction of residue records. Since (l,e) diversity uses generalization for anonymization it leads to more number of residue records.

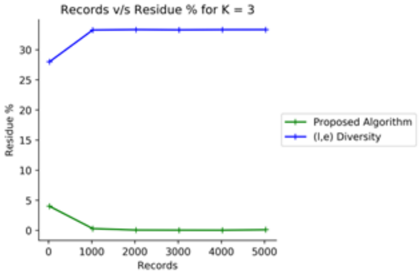


Fig. 9. Reduced residual records in proposed method vs (l, e) diversity.

2) Computational time: As shown in Fig. 10, the time taken by the proposed algorithm is slightly higher than (l, e) diversity because the algorithm considers MSA where as (l,e) preserves privacy of SSA.

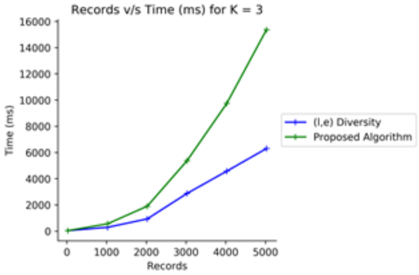


Fig. 10. Computation time in proposed algorithm vs (l, e) diversity.

3) Diversity percentage: The diversity percentage achieved in the proposed method with multiple sensitive attributes is much better when compared with (l,e) diversity. This is mainly because the attributes are selected based on their semantics and every EQ has diversified primary sensitive attribute. This is shown in . 11.

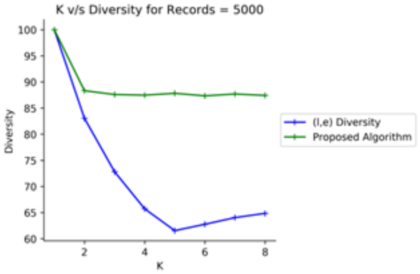


Fig. 11. Diversity percentage in proposed algorithm vs (l,e) diversity.

C. Comparision of Incremental Diversity, SDSV and (l,e) Diversity

As discussed in related work section, one recent algorithm that discusses the distribution of sensitive attributes is SDSV algorithm. However, the algorithm doesn't consider the semantic similarity between the attributes within an EQ. This leads to semantic diversity attack and weaker diversity among the attributes within an EQ's. It can be seen from Fig. 12, that, (l, e) diversity has highest diversity among the attributes within an EQ, since it considers single sensitive attributes. The diversity percentage for the proposed algorithm is average considering the multiple sensitive attributes and their semantic.

D. Security Evaluations

As mentioned before the privacy attacks considered in this work are semantic attacks, similarity attacks and association attacks that are predominant in data set with MSA. The proposed algorithm overcomes all these threats since the semantics of the sensitive attributes is addressed. Consider Table II, III, IV and V that were generated by SDSV algorithm. The algorithm did not consider the semantic relationship between the sensitive attributes there were semantically similar attribute values for disease within an equivalence class. Also, the algorithm generates multiple tables and this increases as the number of sensitive attributes increases.

The proposed algorithm overcomes the semantic and similarity attacks. Consider Tables VI and VII that are generated using the proposed algorithm. Every equivalence class has diversity of the sensitive attributes, which becomes difficult for the intruder to cause privacy threats. Even though the intruder knows one of the sensitive attribute and a quasi identifier it is difficult to cause association attack. For example, if the intruder is neighbour of Trudy (from Table I), he knows that he is unmarried and also the Zip Code. The intruder wants to determine other sensitive attributes like disease. On observing the published table, he gets to know that his record belongs to group 2 of Table VII. Here, since there are 2 records that have "Unmarried" as attribute value of Martial Status, he cannot predict Trudy's disease. Form the above experiments

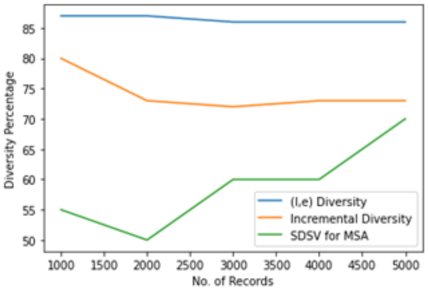


Fig. 12. Diversity percentage of (l, e) diversity, proposed method and SDSV for MSA.

and results it can be concluded that the proposed algorithm is efficient in terms of reducing the residual records, computation time at the same time achieving optimal diversity considering multiple sensitive attributes. Also, as discussed,the proposed algorithm overcomes the privacy threats that exists for MSA.

VII. Conclusion and Future Work

Concern to data privacy is increased with the increase in the digital technology. The personal data is collected at various places that contain multiple sensitive attributes (MSA). These attributes must be treated well to prevent privacy threats when the data set is published to outside world. Many algorithms have been proposed to preserve privacy of MSA in the literature. In these algorithms one of the attribute is chosen as a primary sensitive attribute and the microdata is anonymized. These algorithms do not discuss how to rank the sensitive attributes. This is the essential step in anonymizing the data. In this paper we discuss an efficient approach to rank the sensitive attributes and then anonymize the data. Experiments along with performance parameters, prove that our algorithm outperforms the existing methods and can be efficiently used to anonymize the data. As a part of future work we would propose an infrastructure framework where in the tables can be published.

References

[1] Quach, S., Thaichon, P., Martin, K.D. et al., ”Digital technologies: tensions in privacy and data”, J. of the Acad. Mark. Sci. vol.50, pp. 1299–1323,2022.

[2] Fung, Benjamin CM, et al. ”Privacy-preserving data publishing: A survey of recent developments”, ACM Computing Surveys (Csur), vol.42, issue 4, pp. 1-53, 2010.

[3] Cox, Lawrence H. ”Suppression methodology and statistical disclosure control”, Journal of the American Statistical Association, vol.75, issue 370, pp. 377-385,1980.

[4] Sowmyarani C. N. and Dayananda P., “Analytical Study on Privacy Attack Models in Privacy Preserving Data Publishing,” pp. 98–116. doi: 10.4018/978-1-5225-1829-7.ch006.

[5] L. Sweeny, “Achieving K-Anonymity Privacy Protection Using Generalization and Suppression”, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 05, pp. 571–588, Oct. 2002, doi: 10.1142/S021848850200165X.

[6] K. El Emam and F. K. Dankar, “Protecting privacy using K-anonymity,” Journal of the American Medical Informatics Association, vol. 15, no. 5, pp. 627–637, 2008.

[7] V. Ciriani, S. D. C. Vimercati, S. Foresti, and P. Samarati, “k - Anonymity”, Privacy-Preserving Data Mining. Advances in Database Systems, vol 34. Springer, Boston, MA, 2008, doi: 10.1007/978-0-387-70992-5.

[8] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern, ” Worst-case background knowledge for privacy-preserving data publishing” Proceedings of 23rd International Conference on Data Engineering, 126–135, 2007

[9] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” ACM Trans Knowl Discov Data, vol. 1, no. 1, 2007, doi: 10.1145/1217299.1217302.

[10] Ninghui Li, Tiancheng Li, and S. Venkatasubramanian, “Closeness: A New Privacy Measure for Data Publishing,” IEEE Trans Knowl Data Eng, vol. 22, no. 7, pp. 943–956, Jul. 2010, doi: 10.1109/TKDE.2009.139

[11] X. Xiao and Y. Tao, “Anatomy: Simple and effective privacy preservation,” VLDB 2006 - Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 139–150, 2006,doi: 10.5555/1182635.1164141

[12] T. Li, N. Li, J. Zhang, and I. Molloy, “Slicing: A new approach for privacy preserving data publishing,” IEEE Trans Knowl Data Eng, vol. 24, no. 3, pp. 561–574, 2012, doi: 10.1109/TKDE.2010.236.

[13] D. Li, X. He, L. bin Cao, and H. Chen, ”Permutation anonymization”, Journal of Intellegent Information Systems,47(3) 427–445, 2016.

[14] He, X., Xiao, Y., Li, Y., Wang, Q., Wang, W., Shi, B: ”Permutation Anonymization: Improving Anatomy for Privacy Preservation in Data Publication”, Cao, L., Huang, J.Z., Bailey, J., Koh, Y.S., Luo, J. (eds) New Frontiers in Applied Data Mining. PAKDD 2011. Lecture Notes in Computer Science, Springer, Berlin, 7104, 111-123,2012.

[15] M. Bahrami and M. Singhal,”A light-weight permutation based method for data privacy in mobile cloud computing”, Proceedings - 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud, pp. 189–196,2015.

[16] K. Oishi, Y. Sei, Y. Tahara, and A. Ohsuga, ”Semantic diversity: Privacy considering distance between values of sensitive attribute”, Computer Security,94, 1–49,2020.

[17] H. Wang, J. Han, J. Wang, and L. Wang ”(l, e)-Diversity – A Privacy Preserving Model to Resist Semantic Similarity Attack”, Journal of Computers,9(1),59–64, 2014.

[18] Aggarwal, C. C.,”On k-anonymity and the curse of dimensionality”, VLDB, 5, 901-909,2005.

[19] X.-C. Yang, Y.-Z. Wang, B. Wang, and G. Yu,”Privacy Preserving Approaches for Multiple Sensitive Attributes in Data Publishing”, Chinese Journal of Computers,31(4),574–587,2009.

[20] A. Anjum, N. Ahmad, S. U. R. Malik, S. Zubair, and B. Shahzad, ”An efficient approach for publishing microdata for multiple sensitive attributes”,Journal of Super Computers, 74(10), 5127–5155,2018.

[21] F. Liu, Y. Jia, and W. Han, ”A new k-anonymity algorithm towards multiple sensitive attributes”, In:Proceedings of IEEE 12th International Conference on Computer and Information Technology, CIT 2012,768–772,2012.

[22] T. S. Gal, Z. Chen, and A. Gangopadhyay, ”A Privacy Protection Model for Patient Data with Multiple Sensitive Attributes”, International Journal of Information Security and Privacy (IJISP), 2(3), 28–44, 2008.

[23] T. Yi and M. Shi, ”Privacy Protection Method for Multiple Sensitive Attributes Based on Strong Rule”, Mathematical Problems in Engineering, 2015, 1024-123X, 2015.

[24] T. Kanwalet al,”A robust privacy preserving approach for electronic health records using multiple dataset with multiple sensitive attributes”, Computer Security, 105,102224, 2021.

[25] R. Wang, Y. Zhu, T.-S. Chen, and C.-C. Chang, ”Privacy-Preserving Algorithms for Multiple Sensitive Attributes Satisfying t-Closeness”, Journal of Computer Science Technology, 33(6) 1231–1242, 2018.

[26] Widodo, M. Nugraheni, and I. P. Sari, ”Simple Distribution of Sensitive Values for Multiple Sensitive Attributes in Privacy Preserving Data Publishing to Achieve Anatomy”, Proceedings of 2nd International Conference on Innovative and Creative Information Technology(ICITech 2021),216–220,2021.

[27] LeFevre, K., DeWitt, D. J., Ramakrishnan, R,”Mondrian multidimensional k-anonymity”,Proceedings of 22nd International conference on data engineering (ICDE’06), 25-25,2006.

[28] T. M. Truta and B. Vinay, ”Privacy Protection: p-Sensitive k-Anonymity Property”, Proceedings of 22nd International Conference on Data Engineering Workshops (ICDEW’06), 94–94, 2006.

[29] X. Sun, H. Wang, T. M. Truta, J. Li, and P. Li: (p+,α)-sensitive k-anonymity: A new enhanced privacy protection model. In: Proceedings of IEEE 8th International Conference on Computer and Information Technology(CIT 2008) 59–64 (2008).

[30] CN Sowmyarani, Veena Gadad, Dayananda P, ”(p+, α, t)-Anonymity Technique Against Privacy Attacks”, International Journal of Information Security and Privacy (IJISP),15(2),68–86,2021.

[31] J. Liu, J. Luo, and J. Z. Huang,” Rating: Privacy Preservation for Multiple Attributes with Different Sensitivity Requirements”,Proceedings of 11th International Conference on Data Mining Workshops,666–673,2011.

[32] Y. Wu, X. Ruan, S. Liao, and X. Wang, ”P-cover k-anonymity model for protecting multiple sensitive attributes”,Proceedings of 5th International Conference on Computer Science and Education(ICCSE 2010), 179–183, 2010).

[33] Y. Ye, Y. Liu, C. Wang, D. Lv, and J. Feng,”Decomposition: Privacy preservation for multiple sensitive attributes”,Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5463,486–490,2009).

[34] D. Das and D. K. Bhattacharyya,”Decomposition+: Improving l-diversity for Multiple Sensitive Attributes”, Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST. 85(2),403–412,2012.

[35] S. A. Onashoga, B. A. Bamiro, A. T. Akinwale, and J. A. Oguntuase,”KC-Slice: A dynamic privacy-preserving data publishing technique for multisensitive attributes”, Information Security Journal.26(3) 121–135, 2017.

[36] N. V. S. Lakshmipathi Raju, M. N. Seetaramanath, and P. Srinivasa Rao,”An enhanced dynamic KC-slice model for privacy preserving data publishing with multiple sensitive attributes by inducing sensitivity”, Journal of King Saud University - Computer and Information Sciences,34(1),1394-1406,2018.

[37] J. Jayapradha, M. Prakash, Y. Alotaibi, O. I. Khalaf, and S. A. Alghamdi, ”Heap Bucketization Anonymity—An Efficient Privacy-Preserving Data Publishing Model for Multiple Sensitive Attributes”, IEEE Access,10, 28773–28791, 2022.

[38] D. W. Murphy, P. M., and Aha: UCI repository of machine learning databases. https://archive.ics.uci.edu/ml/datasets/adult,1996.

Demand Forecasting Models for Food Industry by Utilizing Machine Learning Approaches

Nouran Nassibi[1], Heba Fasihuddin[2], Lobna Hsairi[3]
Department of Computer Science and Artificial Intelligence[1]
Department of Information Systems and Technology[2,3]
University of Jeddah, Jeddah 23443, Saudi Arabia[1,2,3]

***Abstract*—Continued global economic instability and uncertainty is causing difficulties in predicting sales. As a result, many sectors and decision-makers are facing new, pressing challenges. In supply chain management, the food industry is a key sector in which sales movement and the demand forecasting for food products are more difficult to predict. Accurate sales forecasting helps to minimize stored and expired items across individual stores and, thus, reduces the potential loss of these expired products. To help food companies adapt to rapid changes and manage their supply chain more effectively, it is a necessary to utilize machine learning (ML) approaches because of ML's ability to process and evaluate large amounts of data efficiently. This research compares two forecasting models for confectionery products from one of the largest distribution companies in Saudi Arabia in order to improve the company's ability to predict demand for their products using machine learning algorithms. To achieve this goal, Support Vectors Machine (SVM) and Long Short-Term Memory (LSTM) algorithms were utilized. In addition, the models were evaluated based on their performance in forecasting quarterly time series. Both algorithms provided strong results when measured against the demand forecasting model, but overall the LSTM outperformed the SVM.**

***Keywords*—Machine learning; long short-term memory; support vector machine; food industry; supply chain management; demand forecasting; product sales**

I. Introduction

Supply Chain Management (SCM) has been a key area of study and professional practice since the 1980s. However, in recent years, supply chains have come under increased scrutiny due to their critical role in business success or failure. A supply chain relies on a coordinated network of companies and sectors.Within this network, materials are obtained and processed into intermediate or final products so that the final products can be sent to users [1]. SCM has four main processes: plan, source, execute and deliver as shown in Fig. 1. Demand forecasting is one of the main axes of SCM [1]. In a changing world, forecasting has increased in importance across many sectors and forecasting has a particular relevance to supply chain management: accurate forecasting allows a company to ensure that supply exists to meet demand. Demand forecasting involves utilizing a probabilistic assessment of the available data, for aims to quantify and forecast future consumer demand for a good or service [2]. A corporation can improve it's supply decisions by using demand forecasting to predict possible sales volume and profitability. By estimating future sales from analyzing historical consumer trends, a business can use demand forecasting to make the most of their inventory [3]. Machine learning (ML) algorithms can forecast food sales by analysing the wealth of historical sales data and adapting to changes within it. ML models have greater predictive power than linear models with progressive parameter selection. Furthermore, the use of ML algorithms in the forecasting process provides adaptive capabilities to members of the supply chain. The system can be considered adaptable through its flexibility in improving the agreement between supply and demand. As a result, it improves the inventory balance throughout the chain by avoiding overstocking of products that are not in high demand [4]. The main focus of this research was forecasting demand within supply chains in the food industry. Although demand forecasting is important for the success of all supply chain processes, it has a critical role in the food industry because products are perishable. Thus, in this case demand forecasting directly contributes to resource preservation and sustainability. More specifically, this study focuses on using ML on long-shelf-life products, especially confectionery (such as chocolates). Such a model may support distribution companies in demand forecasting and stakeholder management with manufacturers and retailers. LSTM and SVM models were built to forecast demand of individual features for each city, or distribution channel, and product.

The remaining sections of this paper are as follows: Section II offers a brief literature review of research concerning demand forecasting in the food industry, Section III presents the research methodology, Section IV outlines the confectionery distribution company's data set, and Section V analyses the results of the forecasting model. Finally, Section VI shares the conclusion of this research and discusses directions which future research could undertake to further the field.

II. Literature Review: Demand Forecasting Models in the Food Industry

Demand forecasting plays crucial role in supply chain management. Using machine-learning algorithms in demand forecasting aids decision-makers in making effective and prescient choices. Accurate demand forecasting leads to increases in company revenue and stock value. Therefore, over the last decade, significant research has been conducted on sales demand forecasting in the food industry. A general overview of forecasting models for sales demand in food industry is provided in systematic review of [6]. This demonstrates the benefits of using ML techniques in the food industry especially for forecasting sales across several types of outlets including confectionery stores, grocer's shops and restaurants. ML techniques have greater predictive power than conventional approaches, which are subject to human error: the

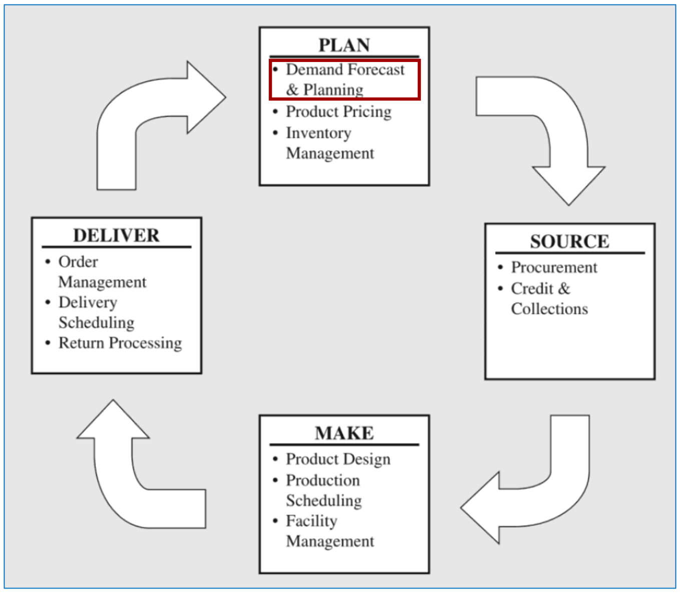


Fig. 1. SCM's four phases. [5]

main advantages of accurate forecasting are that it allows stores to dispose of expired products and minimize stock levels [6]. "Shelf-life" refers to the length of time a given foodstuff remains viable for purchase as a quality product and determines how long products remain on sale [7]. The paradox of having to throw away excessive amounts of products is a significant challenge for retailers. This is especially the case when selling items with a short shelf-life, such as fruits and vegetables. Researchers have made continuous efforts to improve demand forecasting for short shelf-life products. For instance, [3] estimated the sales demand for an Austrian retailer operating in the food industry, specifically with regards perishable products. The researchers used two different models Seasonal Autoregressive Integrated Moving Average (SARIMA) and Long Short-Term Memory (LSTM) and historical daily sales from January 2017 to December 2019 of four perishable products sold in over 90 stores. Both models produced useful outcomes, however, LSTM outperformed SARIMA for products with stable demand, while SARIMA outperformed LSTM for seasonal products. Furthermore, the researchers compared results with SARIMAX after including the external factors such as promotions sales and discovered that SARIMAX performed significantly better for products with external variables. Similarly, [8] used a neural networks approach to construct a forecasting model to predict annual import fruit for the following year. Moreover, [7] used various forecasting models such as LSTM networks, Support Vector Regression (SVR), Random Forest Regression (RFR), Gradient Boosting Regression (GBR), Extreme GBR (XGBoost/XGBR) and Autoregressive Integrated Moving Average (ARIMA) to attain the daily optimal order quantity of fresh produce and to avoid specific vegetables becoming out-of-stock. The study was conducted on the fresh vegetables section of a retail outlet of a college campus. They selected a range of products to fit the categories of low, moderate and long shelf-life (tomatoes, onions and potatoes, respectively). The results indicated that ML algorithms, namely, LSTM and SVR, produced better results as compared with other demand forecasting models. In another study [9], the forecast demand for a Portuguese company's sales was obtained by comparing various statistical methods (moving average, exponential smoothing and ARIMA). In this case, historical weekly sales of delicatessen products from 2013–2017 were used as the data. The author in [9] combined different forecasting techniques to produce consistently good results. Moreover, they used a simple average to combine the three different results of forecasts. In addition to previous studies, [10] proposed a model of Long Short-Term Memory (LSTM) to forecast daily consumer demand using Moroccan supermarket data from a period of six months. In this study, multi-layer neural was found to be the best neural network framework for demand forecasting. Forecasting future products sales enables stores and companies to avoid food waste. Therefore, [11] presented a case study of several ML models using real-time sales data from a restaurant. They applied data by using over 20 models to demonstrate the impact of creating stationary data-sets on the pre-processing of the feature and model training processes. The results showed that Recurrent Neural Network (RNN) models outperformed other models. The author in [12] also studied the efficacy of ML techniques in forecasting daily customer demand for beer in a restaurant setting. Their predictions were based on combining two different kind of data: internal data (such as point-of-sale (POS) transaction data) and external data (such as weather conditions). Demand forecasting models are not only employed within the food industry, but can also be applied to many other sectors. In e-commerce settings, some products are interconnected, and can be categorized into one subcategory; thus, they have correlated sales and demand patterns. Therefore, [13] suggested that better predictions would arise from using historical data from related products to forecast product demand. They applied an LSTM model by using historical data of interconnected products from Walmart.com in order to forecast the demand of other products within the same category, and achieved more accurate results by using LSTM model. To date, the field has seen a significant amount of research in demand forecasting. However, none of this deals with multi-faceted nature of the supply chain for such products, including channel distributions and city. The present study aims to apply some of these insights to the confectionery distribution industry. Table I summarizes previous studies that are related to the current research. There are two important attributes to consider when making comparisons:

Attribute 1: That the study considers sales by city to forecast demand.

Attribute 2: That the study examines the geographic distribution of products across different channels, such as larger stores, mini-markets and wholesale retailers.

This presence of these criteria are indicated in Table I as:

Y: Yes, this criterion is applied or considered.

N: No, this criterion is not applied or considered.

III. Methodology

This section reviews the methodology applied to build the forecasting model. The research methodology is based on artificial intelligence (AI) in food supply chains, taking into account a number of factors which significantly affect sales,

TABLE I. Comparison with Previous Research using the Proposed Models

Ref.No	Year	Products Type	Demand Type	Methodology	Attribute 1 (City)	Attribute 2 (Channel)
[10]	2017	Moroccan supermarket products	Daily sales	LSTM	N	N
[7]	2019	Vegetables	Daily order quantity	LSTM networks, SVR, RFR, GBR, XGBoost and ARIMA	N	N
[9]	2019	Delicatessen products	Weekly sales	Moving Average, Exponential smoothing and ARIMA	N	N
[13]	2019	E-commerce products	Daily sales	LSTM	N	N
[12]	2020	Beer	Daily order quantity	Random Forest Regression	N	Y
[3]	2022	Vegetables	Daily sales	SARIMA and LSTM	N	Y
[11]	2022	Mid-sized restaurant	Daily and weekly sales	Over 20 models,Recurrent Neural Network (RNN) is best	N	N
This Study	2022	Chocolate	Quarterly sales	LSTM and SVM	Y	Y

such as city distribution channels and actual sales revenue. The methodology is based on a ML algorithm in order to map between input and output data and to discover the underlying rules governing the movement of the time chain so that realistic future predictions can be made. Fig. 2 outlines the methodology for the proposed model.

The proposed model comprises three major phases:

A. Data Pre-Processing

After collecting data from the chosen confectionery company in the Kingdom of Saudi Arabia, it was processed through noise removal. Effective pre-processing of data is essential for network input, it is better to convert raw time-series data into indicators which represent basic information more clearly. Therefore, features were classified as follows:

- **Three types of product sales are considered:**
  - ○ Actual sales revenue during regular periods, which represents the income the company or factory generates from the sale of its products.
  - ○ Sales promotions refers to sales records during offer periods (such as those occasioned by holidays, back-to-school offers and special events). Such promotions play a crucial role in demand forecasting as they can skew the results.
  - ○ Returns of products.

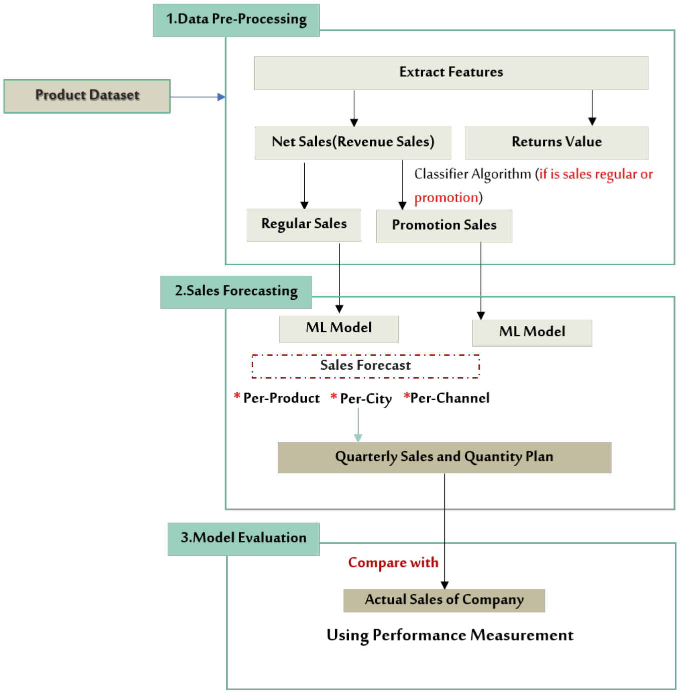


Fig. 2. Methodology for the proposed model.

- City.
- Distribution Channel.

B. Applying ML Algorithms in Forecasting Models

Based on the pre-processing phase, the appropriate model was chosen and applied to forecast quarterly sales volume and the required quantities of the product based on various factors such as city and distribution channel. Two machine learning algorithms were used in this study to determine the optimal order quantity of different chocolate products. This section describes each algorithm in overview.

1) Support vector machine (SVM): One of the supervised learning methods used to solve classification and regression problems is the Support Vector Machine (SVM). SVM is best suited to forecasting products with high dimensional margins, where the number of features exceeds the number of inputs [14]. SVM has often been used as the solution to demand forecasting in the food industry due to the need to solve regression problems. A key advantage of the SVM method over more conventional prediction techniques is that it does not require any previous information regarding the link between the input and the output [16].

[18] obtained more robust results using the SVM algorithm when forecasting demand for perishable foods. Furthermore, [7] produced good results with low forecast error when using the SVM algorithm to predict daily sales of vegetables. The findings from studies which have used this method, indicate that SVM produces strong results [10]. Therefore, SVM was selected for making predictions in the current study.

2) Long short-term memory (LSTM): Recurrent Neural Networks (RNNs) are a popular method for modelling time series data. However, RNN has some limitations. Firstly, all inputs must be the same size. Secondly, RNN suffers from the disappearing gradient problem. Due to these limitations a Long Short-Term Memory (LSTM) algorithm can be used to solve this problem. LSTM can save information from inferences in sequential data in long memory. This algorithm describes data properties without requiring previous knowledge of parameters or distribution of features [15].

C. Model Evaluation

To evaluate each model, the sales it predicted were compared with the actual sales data. The accuracy of the forecasting model was measured using two common performance measures:

- **Mean Absolute Percentage Error (MAPE).**
- **Root Mean Square Error (RMSE).**

IV. Confectionery Dataset

A large dataset describing customer transactions with the distribution company was provided by a chocolate distribution company in the Kingdom of Saudi Arabia. This dataset was obtained from the company's SAP platform [17], which saves detailed transaction records. This dataset provided more than ten attributes, including those most common to the food sales field as well as factors which might be useful in forecasting product sales. These were factors such as distribution channel, product code, date, plant, returns value, net quantity, and sales in Saudi Riyal (SAR). The target variable is forecasted sales quantities. Table II outlines the dataset dictionary.

Net value and quantity features are the units used by the company to measure the sales of their products. A returns value is represented by a negative value of net value. The data was in the form of daily transactions for 200 products in 11 cities and 6 distribution channels across 3 years (January 1 2018 - December 31 2020). This led to dataset with almost quarter of a million data rows. Table III shows the description of six distribution channels feature that are used in the dataset.

TABLE II. Dataset Dictionary

No.	Feature Name	Description
1	Product Code	A unique 6-digit number referring to an individual product with name of product.
2	Posting Date	Date of transaction sales.
3	City	Location of stores.
4	Distribution Channel	The way of distribute products in stores (6 chanels are available).
5	Net Value	Actual sales of product in Saudi Riyal (SAR).
6	Net Quantity	Actual quantities of product per transaction.
7	Returns Value	Value of purchased products returned to stores in Saudi Riyal (SAR).
8	Returns quantity	Quantity of purchased products returned to stores.

TABLE III. Description of Distribution Channel Feature

No.	Name of channel	Description
1	Key Account	Important major and big-sized main stores i.e (hyper market.)
2	Wholesale	Stores buying large quantities of products from distribution company.
3	Mini Markets	Small stores such as corner shops or newsagents.
4	Convenience Stores	Small stores located in gas stations (for example NAFT, SASCO).
5	New Channel	Stores whose main product line is different, such as Toys "R" Us, Sky sales (Saudi Airline).
6	Cash Van	A car with products in the custody of the company's salesman. The customer pays via cash.

V. Analysing Forecasting Model Performance

Forecasting demand for a company with several different products, across multiple cities and distribution channels is a challenging task. Creating one model which covers all products, cities and channels might not be accurate, because each of these factors affects the inputs and is subject to changing circumstances. The main objective of this study was to enhance predictive ability for sales, thus minimizing food wastage and supply chain issues by forecasting supply and demand, and improving the efficiency of the whole system by minimizing errors like data loss in traditional ways. As such, the developed model aims to forecast demand of products for next year's quarters, while the inputs for the models comprised city, distribution channel and sales quantities of products by quarter for the previous year. Therefore, the dataset was split into different time steps. The training dataset consisted of daily sales transactions during 2018 for products and the testing dataset used 2019 data. For model testing, we chose sales data for the top five products (product 1, product 2, product 3, product 4, and product 5) across the ten cities (Riyadh, Jeddah, Taif, Dammam, Qassim, Makkah, Eisha, Madina, Tabuk, Jizan, and Khamis).

The above-mentioned cities were mapped along the x-axis with product demand along the y-axis. With regards the distribution channel, only one channel was used in testing which was the key account channel. This channel represents the most significant markets in Saudi Arabian cities.

A model validation method was implemented in order to compare the performance of the models. After identifying and developing all of the forecasting models, the performance measures for validating and comparing these models was implemented by applying the following equations, using Python:

- $\sum$: Symbol means "sum", **E** = Actual value – Forecast value and **n** = sample size.
- Mean absolute percentage error, MAPE = $\frac{\sum_{i=1}^{n}|E|*100/A}{n}$ [7], where the actual demand value is indicated by A.
- Root mean squared error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}|E|^2}{n}}$ [7].

Several features potentially play a significant role in forecasting product sales. Separating the sales by channel and city revealed that each of them has unique purchasing trends. This justifies the need for separating out these factors and showing the results of the models for each product by city and by channel. The results of models for multiple cities across one channel using the **LSTM** model are presented in Section 5.1, while Section 5.2 describes the results when the **SVM** model was applied. Section 5.3 discusses the results for both models.

A. LSTM

As explained, time is an important factor in prediction models. Each year's data was split into quarters to observe the accuracy of the forecasting models' output in relation to actual sales. Each algorithm was executed separately to predict sales of the top five products in the given year.

Fig. 3 shows the actual values (coloured full lines) and the predicted values (dotted lines) respectively. Chocolate product demand is shown in the below graph for these cities. The actual values and predicted values by our models are almost identical, indicating the accuracy of the models' output. The LSTM model shows that **Product 2** saw the highest demand in the capital city RUH **(Riyadh)** during **quarter-1, quarter-2 and quarter-4**, as shown in Fig. 3 during **quarter-3**, however, **Product 2** saw the highest demand in the city of **(Dammam)**. Noteworth, **Product 4** not buy any items in **(Tabuk)** city during **quarter-2**.

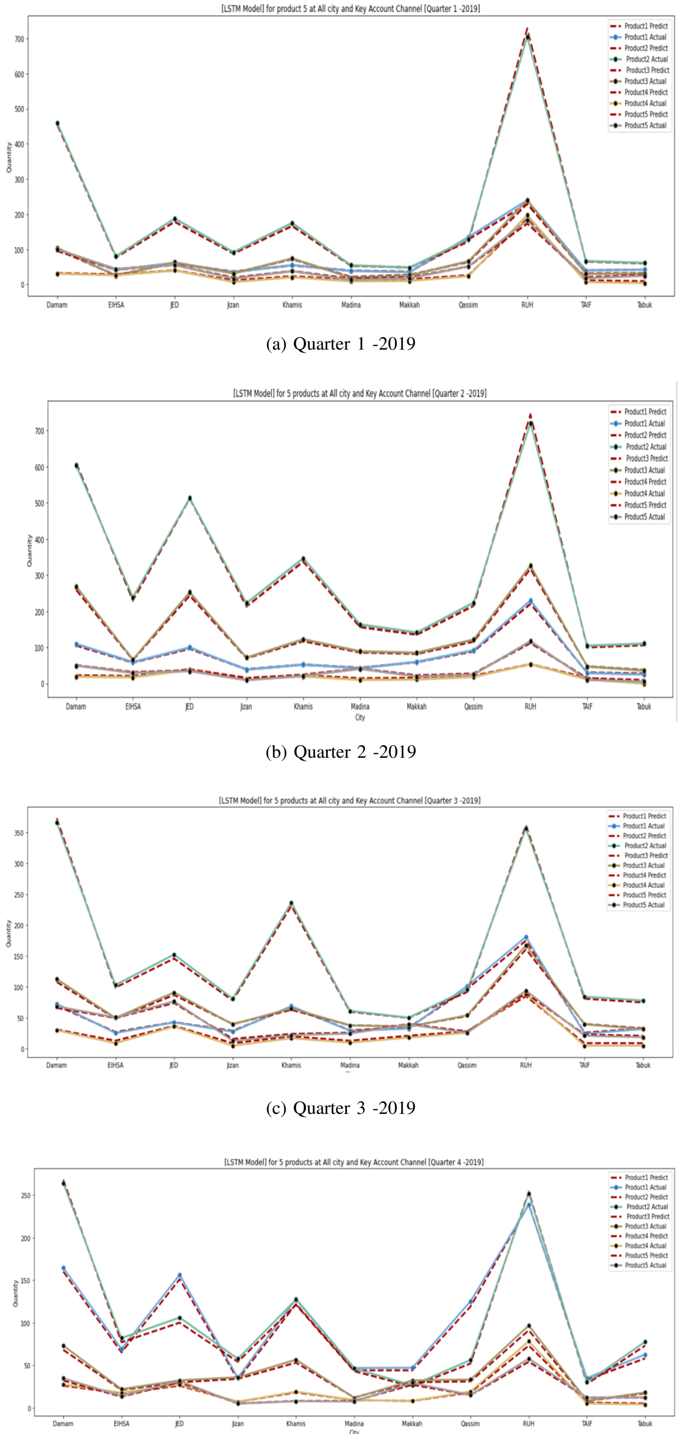
B. SVM

The second model used for sales demand forecasting in this study was SVM. Fig. 4 shows the sales forecasting for products per quarter. Fig. 4 shows that **Product-2** had the most demand in **(Dammam)** during **quarter-4**. During **quarter-2 and quarter-3**, **Product 2** saw the most demand in the capital city of **Riyadh (RUH)**. The graph indicates that the SVM model is less accurate because the predicted values are higher than the actual values during **quarter-3**.

C. Discussion

The key research question for this study concerned which algorithm could better forecast demand, LSTM or SVM. A Support Vector Machine (SVM) is a classification algorithm used for a small amount of data, and it is less accurate than the LTSM. A Long Short-Term Model (LSTM) is used as a deep learning algorithm that performs effectively when used for a large amount of data. The overall results show that the LSTM model is more accurate than the SVM model, because of LSTM's ability to remember the data more efficiently than the SVM. The LSTM algorithm performed best when used for a large amount of data. Overall, LSTM performs better than SVM across all scenarios. However, both algorithms are useful in forecasting demand and when used together provide a more comprehensive picture.

Performance statistics like MAPE and RMSE enable the forecasting models to be evaluated. For this study, the algorithms resulting in the lowest RMSE and MAPE are the most effective. As explained above, the city and distribution channel destination is an important variable and a significant factor in the forecasting sales models. Therefore, Fig. 5 shows a comparison between LSTM and SVM models by city. Fig. 5 indicates that the LSTM forecasting is better than the SVM model's because the MAPE and RMSE values are lower for the LSTM model than for the SVM model. Additionally, Table IV shows all the results of the MAPE and RMSE values. Riyadh and Jeddah have the lowest MAPE and RMSE values as compared with other cities (blue line). This is because Riyadh is the capital and largest city of the Kingdom of Saudi Arabia, and Jeddah is also among the largest cities. Due to their geographical size, the demand for products is higher when



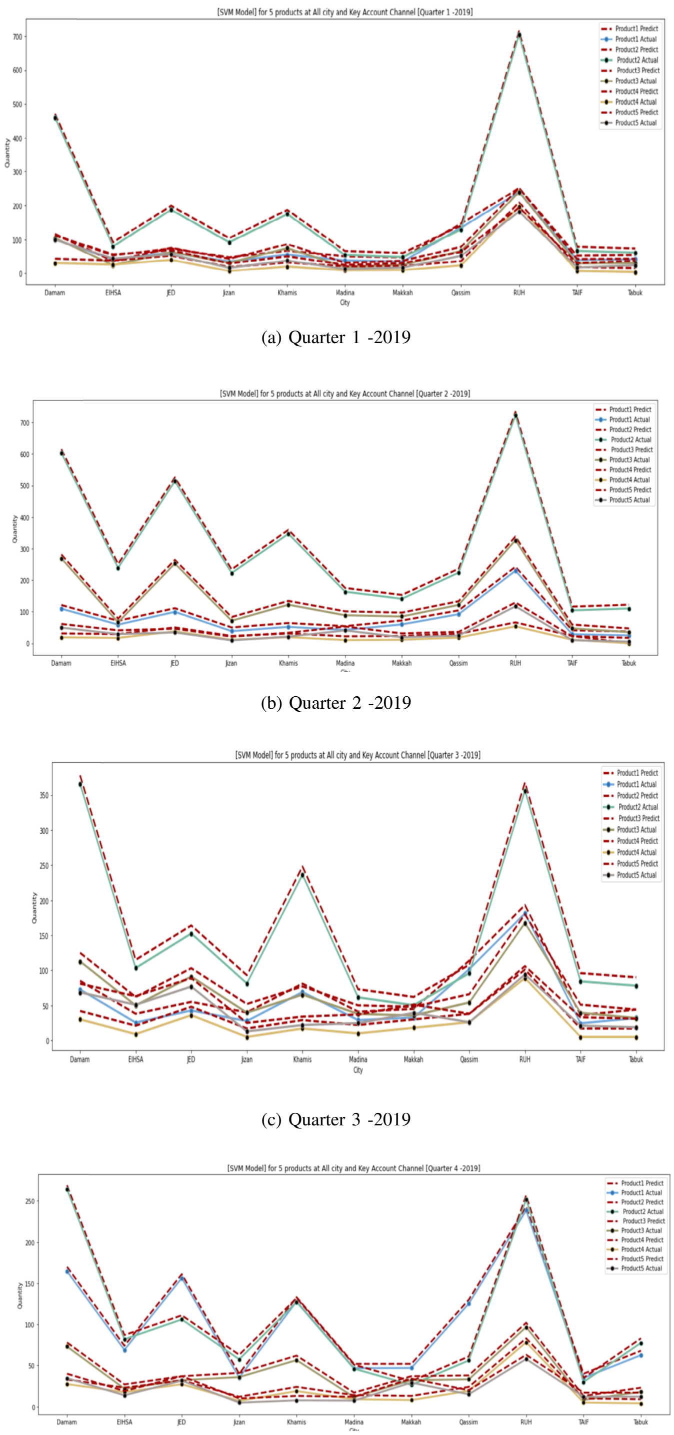
(a) Quarter 1 -2019

(b) Quarter 2 -2019

(c) Quarter 3 -2019

(d) Quarter 4 -2019

Fig. 3. LSTM model for forecasting quarterly sales.



(a) Quarter 1 -2019

(b) Quarter 2 -2019

(c) Quarter 3 -2019

(d) Quarter 4 -2019

Fig. 4. SVM Model for forecasting quarterly sales.

compared with other cities, the number of sales transactions in these two cities creates significant data and both cities had daily sales in the dataset. Since the forecasting model were trained to look for trends in large numbers of sales, this might have influenced their ability to reduce the rate of forecast error in these cities. The values shown in Fig. 5 were obtained by taking values of all four quarters for each city. In order to evaluate the performance of this study's LSTM and SVM models, a comparison against previously studies was undertaken. Table V illustrates those previous studies (in blue) which used similar factors to this study, such as store location. Other studies used general sales with category of products as the main factor for the LSTM and SVM models. This study obtained lower MAPE values compared with previous studies using LSTM (in green). However, this study's results using the SVM model (in yellow) are comparable with [7]. This study selected the most commonly-used algorithms for predictions [10]. Looking at the sequence of charts depicting the results, predicted and actual values agree more closely. Accurately predicting demand will help businesses to make better decisions and consequently save food, generate increased revenue, and solve food supply issues. This study shows that the implementation of LSTM and SVM models for real-life food items and the retail market helps to reduce forecast error, improve daily retail inventory and increase product sales. This will help small businesses to reduce the risk of particular items falling out-of-stock and optimize their sales.

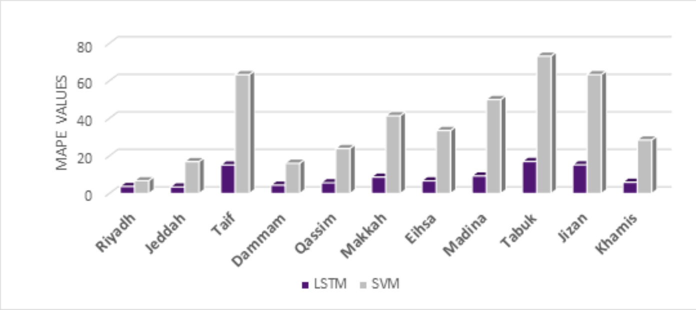


Fig. 5. A comparison of LSTM and SVM city forecasting.

TABLE IV. A COMPARISON OF MAPE AND RMSE VALUES USING LSTM AND SVM

	MAPE		RMSE	
City	LSTM	SVM	LSTM	SVM
Riyadh	3.78	6.8	2.24	7.69
Jeddah	3.48	15.9	2.26	8.21
Taif	15.20	63.3	2.37	10.26
Dammam	4.44	16.1	4.64	10.33
Qassim	5.71	24	4.03	10.25
Makkah	8.75	41.4	2.37	9.90
Eihsa	6.74	33.6	3.02	12.35
Madina	9.37	50	2.52	10.33
Tabuk	17.06	73.2	2.82	10.37
Jizan	15.30	63.4	3.14	9.78

TABLE V. COMPARISON OF MAPE VALUES OF LSTM AND SVM IN PREVIOUS STUDIES

Ref.No	Algorithms	MAPE
[7]	LSTM	9.23
[3]	LSTM	27.01
[11]	LSTM	19.02
[7]	SVM	26.03
The current study	LSTM	8.71
	SVM	37.8

VI. Conclusion and Future Work

In an era where information and data are increasingly available, ML is an important tool from which industries can greatly benefit to future-proof their supply chains. The ability to accurately forecast demand assists distribution companies to manage their supply chains more effectively. This study presents two models for a distribution company in the food industry by using the LSTM and SVM algorithms to forecast demand for products across a variety of factors. In particular, the demand forecasting models here were applied to the individual level of factors such as city and distribution channel. The evaluation of the experiment showed that the LSTM model outperformed SVM. In general, the findings demonstrate that the LSTM model reduces forecasting errors up to **77%** compared to the SVM model. This study has generated key insights concerning the sales of chocolate products within different cities of Saudi Arabia. Sales promotions are one of the most common phenomena in the retail industry. Special events such as marketing campaigns or holiday promotions are examples of valuable retail data that are often not incorporated into single-variable statistical forecasting models. Currently, this study only takes standard sales patterns into account when forecasting demand, so future work needs to examine promotion sales as an independent factor. Furthermore, the dataset used here was the company's sales record for 2020, a year in which the COVID-19 pandemic had a significant negative impact on almost all industries. Future work will extend the sales analysis depicted here to understand how the pandemic affected standard sales behaviour.

Acknowledgment

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-21-DR-138). The authors, therefore, acknowledge with thanks the University of Jeddah technical and financial support.

References

[1] Gružauskas, V., Gimžauskienė, E.,& Navickas, V. (2019). Forecasting accuracy influence on logistics clusters activities: The case of the food industry. *Journal of Cleaner Production*, 240, 118225.

[2] Feizabadi, J. (2022). Machine learning demand forecasting and supply chain performance. *International Journal of Logistics Research and Applications*, 25(2), 119-142.

[3] Falatouri, T., Darbanian, F., Brandtner, P., & Udokwu, C. (2022). Predictive Analytics for Demand Forecasting–A Comparison of SARIMA and LSTM in Retail SCM.*Procedia Computer Science*, 200, 993-1003.

[4] Garre, A., Ruiz, M. C., & Hontoria, E. (2020). Application of Machine Learning to support production planning of a food industry in the context of waste generation under uncertainty. *Operations Research Perspectives*, 7, 100147.

[5] Hugos, M. H. (2018). *Essentials of supply chain management*. John Wiley & Sons.

[6] Tsoumakas, G. (2019). A survey of machine learning techniques for food sales prediction. *Artificial Intelligence Review*, 52(1), 441-447.

[7] Priyadarshi, R., Panigrahi, A., Routroy, S., & Garg, G. K. (2019). Demand forecasting at retail stage for selected vegetables: a performance analysis. *Journal of Modelling in Management*, 14(4), 1042-1063.

[8] Zhang, H., & Lin, A. (2020, December). Research on Demand Analysis Model of Hot Product in Food Industry. *In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 595-602). IEEE.

[9] Silva, J. C., Figueiredo, M. C., & Braga, A. C. (2019, July). Demand forecasting: A case study in the food industry. *In International Conference on Computational Science and Its Applications* (pp. 50-63). Springer, Cham.

[10] Bousqaoui, H., Achchab, S., & Tikito, K. (2017, October). Machine learning applications in supply chains: Long short-term memory for demand forecasting. *In International Conference of Cloud Computing Technologies and Applications* (pp. 301-317). Springer, Cham.

[11] Schmidt, A., Kabir, M. W. U., & Hoque, M. T. (2022). Machine Learning Based Restaurant Sales Forecasting. *Machine Learning and Knowledge Extraction*, 4(1), 105-130.

[12] Tanizaki, T., Hoshino, T., Shimmura, T., & Takenaka, T. (2020). Restaurants store management based on demand forecasting. *Procedia CIRP*, 88, 580-583.

[13] Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., & Seaman, B. (2019, December). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *International conference on neural information processing* (pp. 462-474). Springer, Cham.

[14] Khemakhem, S., Said, F. B., & Boujelbene, Y. (2018). Credit risk assessment for unbalanced datasets based on data mining, artificial neural network and support vector machines. *Journal of Modelling in Management*.

[15] Abbasimehr, H., Shabani, M., & Yousefi, M.(2020).An optimized model using LSTM network for demand forecasting. *Computers & industrial engineering*, 143, 106435.

[16] Ho, T. T., Tran, L. V., Tran, H. M., & Dao, S. V. (2022) Machine Learning in Demand Forecasting. *International Research Journal of Advanced Engineering and Science*.

[17] What is SAP?: Definition and meaning. SAP. (n.d.). Retrieved March 16, 2023, from https://www.sap.com/about/company/what-is-sap.html

[18] Du, X. F., Leung, S. C., Zhang, J. L., & Lai, K. K. (2013). Demand forecasting of perishable farm products using support vector machine. *International journal of systems Science*, 44(3), 556-567.

Method for Inferring the Optimal Number of Clusters with Subsequent Automatic Data Labeling based on Standard Deviation

Aline Montenegro Leal Silva[1], Francisco Alysson da Silva Sousa[2], Alysson Ramires de Freitas Santos[3],
Vinicius Ponte Machado[4], André Macedo Santana[5]
Federal University of Piaui[1,2,4,5], Brazil
Unified Teaching Center of Piaui[3], Brazil

Abstract—**Machine learning is a suitable pattern recognition technique for detecting correlations between data. In the case of unsupervised learning, the groups formed from these correlations can receive a label, which consists of describing them in terms of their most relevant attributes and their respective ranges of values so that they are understood automatically. In this research work, this process is called labeling. However, a challenge for researchers is establishing the optimal number of clusters that best represent the underlying structure of the data subjected to clustering. This optimal number may vary depending on the data set and the grouping method used and influences the data clustering process and, consequently, the interpretability of the generated groups. Therefore, this research aims to provide an inference approach to the number of clusters to be used in the grouping based on the range of attribute values, followed by automatic data labeling based on the standard deviation to maximize the understanding of the groups obtained. This methodology was applied to four databases. The results show that it contributes to the interpretation of the groups since it generates more accurate labels without any overlap between ranges of values, considering the same attribute in different groups.**

Keywords—Inference approach; range of attribute values; labeling; standard deviation; interpretation of the groups

I. Introduction

The rapid popularization of computers in many sectors of society has resulted in significant data volume growth [1]. The researchers then began to use pattern recognition techniques by detecting correlations between the data, which could bring to light relevant and valuable knowledge potentially contained in these databases [2].

One of these pattern recognition techniques is machine learning (ML), which emerged from the realization of creating computer programs that learn a particular behavior or pattern automatically from examples or observations. The idea behind learning is that after looking at some data, a computer builds a model based on that data and uses that model as a hypothesis about the world and a piece of software that can solve problems [3].

Machine learning can consist of two main paradigms: supervised and unsupervised. For supervised learning, the aim is to create an accurate model for predicting values for new data. As for unsupervised learning, the objective is to find characteristics that can summarize the data. Other paradigms exist, such as reinforcement learning, multitasking, and semi-supervised.

As one of the best-known techniques in the area of unsupervised learning, grouping or clustering of data consists of defining a set of groups or clusters in which the elements of the same group are as similar as possible to each other, and the elements of different groups are as distinct as possible [4]. Because it is subjective, it does not provide clear information that allows inferring the characteristics of each cluster formed [5] due to the algorithms' limitations.

Establishing the optimal number of groups in a clustering algorithm is one of the most challenging and fundamental tasks for researchers since different amounts of clusters cause different results, influencing the performance of the clustering process [6], [7], [8]. For example, Cobweb [9] is a hierarchical algorithm whose order of factors affects the grouping and is very sensitive to data input. K-means [10] is an algorithm based on Euclidean distance dependent on the initial partition generated by the random choice of centroids, requiring the number K of groups to be informed in advance.

Labeling seeks to synthesize its definition, describing the groups' most relevant attributes and respective value ranges to understand the specialist better. Due to some limitations resulting from the grouping, auxiliary techniques can infer characteristics that identify the formed groups. Among these techniques are dispersion metrics, such as standard deviation. Therefore, the resulting clusters are labeled to be understood automatically. In this research work, this process is called automatic data labeling, which aims to identify the characteristics of each group and, later, allow the complete interpretation of the generated clusters.

In this sense, this research aims to provide an inference approach to the number of groups to be used in the clustering process, based on the range of attribute values, with subsequent automatic data labeling from the standard deviation to maximize the understanding of the groups obtained, without overlapping any range of values in the same dataset, considering the same attribute. It consists of calculating the standard deviation according to the value of each attribute. When necessary, the value of this deviation is increased to the lowest value of each attribute in the dataset or decreased by the highest value of each attribute, or both steps have been performed. This methodology was applied to four databases of different sizes.

The results show that it contributes to the interpretation of groups, as it generates more accurate labels since the algorithm developed to choose the optimal number of groups performs better when compared to other isolated methods, such as Elbow, Silhouette Coefficient, and Calinski-Harabasz, for example. Furthermore, the labels obtained do not overlap between ranges of values considering the same attribute in different groups. It contributes to better interpretability of the generated clusters.

In addition to the Introduction, the rest of this article is organized as follows. Section II presents the theoretical framework used in this model, Section III addresses influence studies for this research, Section IV displays the methodology used, Section V presents the results obtained and, finally, Section VI describes the conclusion of the research work.

II. Theoretical Reference

In this session, the selection processes of relevant attributes and measures of dispersion of data used in this research will be presented, followed by clustering, in addition to the labeling problem.

A. Dispersion Measures

According to [10], a measure of dispersion for a quantitative variable indicates the degree of spread of sample values around the centrality measure, indicating how much the elements differ from the mean of the data set. Greater dispersions exhibit less representativeness of the central values. One of the advantages of using these metrics is that they observe the data set as a whole and assess the degree of homogeneity or dispersion of this set, favoring a reduced computational cost. Next, a relevant metric used in data labeling will be presented to understand the *clusters* formed.

1) Standard Deviation: Mathematically, the standard deviation (SD) is a measure of dispersion used to quantify the variation or dispersion of a set of data values [11]. Equation 2 shows how the standard deviation calculation is performed.

$$DP = \sqrt{\frac{\sum_{i=1}^{n}(x_{\mathrm{i}} - MA)^2}{n}} \tag{1}$$

For clarification:

- SD: standard deviation;
- $\mathrm{x_i}$: value at position i in the data set;
- MA: arithmetic mean of the data;
- n: the amount of data.

A low standard deviation means that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread over a wide range of values.

B. Clustering

The basic idea of clustering is those elements that make up the same *cluster* must show high similarity (i.e. be very similar elements and follow a similar pattern). Still, it must be very dissimilar from objects in other groups. In other words, all clustering is done to maximize homogeneity within each *cluster* and maximize heterogeneity between groups.

K-means is one of the most popular clustering algorithms. The result of the K-means [12] method is generally influenced by the K-partition chosen in the initial step. If K is too small, there will be distinct elements in the same cluster. On the other hand, if K is very high, similar elements will be in different clusters. For this reason, it is recommended to validate the result of the cluster analysis based on inference criteria of the optimal number of groups in a data set, bearing in mind that different amounts of *clusters* generate different results, influencing the performance of clustering and consequently in understanding the *clusters* formed.

C. Labeling Problem

The task of interpreting clusters is commonly assigned to a specialist in the field under study who examines each group with respect to its objects to label them, describing the nature of the group. This process tends to be too laborious concerning time and resources, considering the amount of data and subjectivity of the task.

In view of this, [14] proposed a method for automatic extraction of characteristics from the groups, providing specialists a label with a selection of the most relevant characteristics of the elements of each group. These features are composed of attribute values range, so the labeling problem is defined as:

Given a set of clusters C $=\{\mathrm{c_1, ..., c_k} \mid \mathrm{k}\geq 1\}$, so that each cluster contains a set of elements $\mathrm{c_i} = \{\vec{e}_1, ..., \vec{e}_{\mathrm{n(c_i)}} | \mathrm{n}^{(\mathrm{c_i})}\geq 1\}$ which can be represented by a vector of attributes defined in $\mathrm{R^m}$ and expressed by $\vec{e}_{\mathrm{j}}^{\,(\mathrm{c_i})}$ = ($\mathrm{a_1}$, ..., $\mathrm{a_m}$) and even though $c_1 \cap c_{1'} = \emptyset$ with 1 $\geq$ i, i '$\geq$ K and i $\neq$i'; it aims to present a set of labels R $=\{\mathrm{r_{c_i}}, ..., \mathrm{r_{c_k}}\}$ in which each specific label is given by a set of pairs of values, attributes and their respective range $r_{(\mathrm{c_i})} = (a_1, [p_1, q_1]), ..., (a_{\mathrm{m(c_i)}},]p_{\mathrm{m(c_i)}}, q_{\mathrm{m(c_i)}}])$ able to better express the associated $\mathrm{c_i}$ cluster.

In order to clarify:

- K is the number of clusters;
- $\mathrm{c_i}$ is any cluster;
- $n^{(c_{\mathrm{i}})}$ is the number of elements in cluster $\mathrm{c_i}$;
- $\vec{e}_{\mathrm{j}}^{\,(c_{\mathrm{i}})}$ refers to the j-th element belonging to cluster $\mathrm{c_i}$;
- m is the dimension of the problem;
- $\mathrm{r_{(c_i)}}$ is the label for cluster $\mathrm{c_i}$;
- $]\mathrm{p_{m(c_i)}},\mathrm{q_{m(c_i)}}]$ represents the values range of attribute $\mathrm{a_{m(c_i)}}$ where $\mathrm{p_{m(c_i)}}$ is lower limit and $\mathrm{q_{m(c_i)}}$ is upper limit;
- $m^{(c_{\mathrm{i}})}$ is the number of attributes present in a label for cluster $\mathrm{c_i}$.

Finally, the method has as input a set of clusters and must present as output a specific label for each group that best defines it, according to the specifications already presented.

III. Related Work

This section addresses some methods for inferring the number of groups in the data clustering process, as well as automatic cluster labeling models that influenced this research, presenting its methodologies and results obtained.

In [15], a model was proposed that can be applied to the segmentation of products for inventory management based on the analysis of three basic principles, which are: history (recency (R)), frequency (F), and money spent (monetary (M)) from the K-means algorithm. Meanwhile, the determination of the optimal number of clusters was evaluated using eight validation indices, namely, Elbow Method, Silhouette Index, Calinski-Harabasz Index, Davies-Bouldin Index, Ratkowski Index, Hubert Index, Ball-Hall and Krzanowski-Lai Index to improve objectivity and accuracy in product segmentation compared to using only one method. The result obtained in all these criteria was 3 clusters, the optimal number of groups, with a low variance between the intra-cluster data, resulting in a high similarity between the elements of the same group.

According to [16], a method was presented to classify the egg production of laying hens in Indonesia based on the K-Means clustering algorithm. The survey data was taken from the National Statistics Center of Indonesia and corresponded to the period from 2018 to 2020 from 34 provinces. To validate the number of groups to be used, the researcher evaluated the Davies Bouldin Index (DBI) criterion for each number of existing clusters, which consists of the ratio between the intra-cluster and inter-cluster distances. In this study, 8 clusters were used, and the DBI value was calculated for each. It was observed that the optimal number of groups is four since it has the lowest DBI value.

A model proposed by [14] groups data based on the centroids of the clusters and uses Artificial Neural Networks (ANN) to generate labels for each of them. Initially, a dataset was provided as input to the model. To obtain better performance for continuous values, a discretization process was performed, in which different possible values for each attribute were divided into intervals, which represent the range of values. In the second stage, the clustering process was carried out using an unsupervised algorithm (K-means). Once the groups were generated, a supervised algorithm (ANN) was applied to each of them, using the discretized base to detect which attributes were relevant in the formation of each generated group. This methodology was applied to three databases (Glass, Seeds, and Iris), and the results were obtained with an average more excellent than 88.79% of correctly labeled elements.

The work of [1] used unsupervised and supervised machine learning methods for data clustering and labeling tasks. To group the data, the DAta MIning COde REpository (DAMICORE) algorithm was used, and to label, the Automated Labeling Method (ALM) based on Artificial Neural Networks (ANN) was used. Before data grouping, the data sets were submitted to the discretization step, and the continuous attributes were discretized by the EWD and EFD methods. The results were compared with those presented in the [13] model, and the analysis showed that applying the ALM method generated better results. The groups formed by DAMICORE are more accurate than those obtained by applying the K-Means cluster, with an average accuracy above 90%.

IV. Proposed Method

This research aims to provide an approach for inferring the number of groups to be used in the grouping process, based on a range of attribute values, with subsequent automatic data labeling from standard deviation to maximize the understanding of the groups obtained. Initially, the new method was validated based on a model already proposed in the literature, that of [14], which developed an algorithm for automatic extraction of the characteristics of the groups, contributing to the interpretability of these clusters.

The algorithm K-means in the proposed model used Python's sklearn library, and the ease and robustness of the environment could provide tests that led to a better understanding of the problem addressed.

Table I presents the four databases used, starting from the UCI Repository[1], including Wine, Breast Cancer, Quality White Wine, and Credit Card.

TABLE I. Databases Obtained from the UCI Repository

Databases	Amount of Data	Attributes
Wine	178	13
Breast Cancer	699	10
Wine Quality White	4.898	12
Credit Card	30.000	24

The methodology used by [14] to aid this task of interpreting clusters is illustrated in Fig. 1.

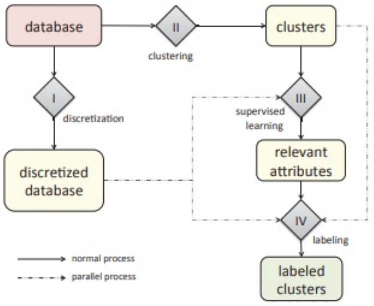


Fig. 1. Labeling template flowchart from [14].

At first, this model receives a database as an input parameter. This base can contain different types of data - discrete or continuous. In some cases, it will be necessary to apply a discretization method (I), which consists of assigning discrete values to attributes that can assume a wide variety of values within a given domain. Thus, the supervised learning algorithm used in step III will be able to identify a possible relationship between attributes with less complexity, showing better results when dealing with the classification problem involving such attributes. According to [17] and [18], there may be an increase in accuracy and speed during the training stage when using a discretization method. In addition, this discretization process allows the inference of a value range, which happens in step IV.

[1] https://archive.ics.uci.edu/ml/index.php

The discretization process starts with selecting which attributes should be discretized and which type to use. Therefore, for this model, the number of ranges of values was defined as FX, and the discretization technique used was by equal frequencies (EFD) to avoid an inadequate distribution of the values of an attribute due to identical elements and consequently cause an imbalance in the distribution of these elements about the ranges of values. In EFD, the number of components with different values between the cut-off points remains constant.

The second step (II) corresponds to using an unsupervised algorithm that receives as input a set of elements (in this case, the database) and presents as output the association of each component to a respective created cluster. The discretized database is not used in this step, but the initially provided database. The algorithm used was K-means, but any other algorithm with unsupervised learning capable of dealing with the clustering problem can be used.

Then step (III), an algorithm with supervised learning is applied to detect the relevant attributes for the definition of each group once the clusters are appropriately formed and the data to be worked on, if necessary, are already discretized. Then, the actual labeling work begins. Each label referring to any group is based on a set of attributes and their respective ranges of values. Therefore, this step has as input a set of clusters and presents as output a set of attributes for each generated group that will be used in its labeling. For this, artificial neural networks of the Multi-Layer Perceptron (MLP) type were used. However, in principle, any other algorithm with supervised learning capable of detecting relationships between variables or any other technique capable of selecting attributes could be chosen. In this case, each neural network presents a hit rate for its learning, performed only with the elements of their respective clusters.

Finally step (IV), a strategy that selects the value (for discrete attributes) or value range (for continuous attributes) for each chosen relevant attribute is applied to generate labels. This strategy seeks to represent the majority of the group so that the selected values for each attribute are those with the highest frequency in the group. The neural network chose the most relevant attributes in step III.

It was noticed, therefore, that the [14] model did not use any inference criteria to optimize the number of groups in the clustering process and did not verify the best amount of range of values for the composition of the labels.

Thus, this research work has the initial intention of providing a method for inferring the optimal number of groups considering different ranges of attribute values, since in the [14] model, as the number of ranges increases of values for the same group K, the hit rate of this model decreases. This fact was found in the four databases used, as shown in Table II.

Based on Table II, this research work developed the method *Optimization of the Number of Clusters Based on a Range of Values* according to the performance analysis (hit rate) of the [14] model to find the optimal number of groups and range of values to be used in the grouping step.

TABLE II. PERFORMANCE ANALYSIS (HIT RATE (%)) FOR VARIATIONS OF K AND VALUE RANGE (FX) ACCORDING TO THE [14] MODEL

WINE	**K=2**	**K=3**	**K=4**	**K=5**	**K=6**
FX=2	95.51	98.96	97.17	98.84	98.90
FX=3	87.29	90.78	96.06	98.21	98.59
FX=4	78.93	86.23	85.07	91.24	92.71
FX=5	71.41	75.36	82.06	87.22	91.88
FX=6	67.27	70.48	73.16	85.84	87.96

BREAST_CANCER	**K=2**	**K=3**	**K=4**	**K=5**	**K=6**
FX=2	91.34	93.41	92.55	92.04	91.58
FX=3	90.48	86.44	91.05	91.86	90.96
FX=4	86.28	82.17	87.67	91.05	89.95
FX=5	77.39	79.98	75.40	80.01	78.70
FX=6	76.90	74.31	74.50	79.17	78.25

WINE QUALITY WHITE	**K=2**	**K=3**	**K=4**	**K=5**	**K=6**
FX=2	99.90	99.98	100	100	100
FX=3	99.50	99.80	99.94	99.96	99.97
FX=4	98.51	98.66	99.10	99.38	99.61
FX=5	97.24	98.14	97.90	98.38	98.74
FX=6	94.34	94.63	96.06	96.98	97.12

CREDIT CARD	**K=2**	**K=3**	**K=4**	**K=5**	**K=6**
FX=2	92.60	99.94	99.96	99.97	99.98
FX=3	91.50	98.80	98.94	98.90	98.92
FX=4	90.51	97.66	97.10	96.96	97.01
FX=5	87.24	94.14	95.90	95.38	95.12
FX=6	85.34	93.63	94.06	93.98	93.06

1) The starting point of the method was to consider the initial K as the element with the highest frequency among the three inference criteria used (Elbow, Silhouette Coefficient, and Calinski-Harabasz) for the same database, which we call fashion. Table III displays the found values.

   Therefore, the initial K for each of the databases was the following: Wine (3), Breast Cancer (2), Wine Quality White (2), and Credit Card (2).

TABLE III. CRITERIA FOR GROUP INFERENCE

Database	**Elbow**	**Silhouette Coefficient**	**Calinski-Harabasz**
Wine	3	2	3
Breast Cancer	2	2	2
Wine Quality White	2	2	2
Credit Card	2	2	3

2) Next, the range of initial value (FX) equal to the chosen K value was selected, that is, FX = K. For example, the Wine database results will be displayed initially. Therefore, K=3 and FX=3;
3) The model's hits rates for K-1 to K+1 were shown, according to Table IV.

TABLE IV. HIT RATE (%) FOR WINE BASE WITH K BETWEEN K-1 AND K+1

WINE	K=2	K=3	K=4
FX=2			
FX=3	87.29	90.78	96.06
FX=4			

4) The highest hit rate among those displayed in the previous step was verified. In this case, the highest rate is 96.06% for K=4 and FX=3.
5) Next, the model's hit rate for FX-1 and FX+1 were presented, as shown in Table V.
6) Finally, there was the highest hit rate among the last ones calculated. In this case, for K=4 and FX=2, it was 97.17% hit, and for K=4 and FX=4, it corresponded to 85.07%. Therefore, for this database, the

TABLE V. HIT RATE (%) FOR WINE BASE WITH FX BETWEEN FX-1 AND FX+1

WINE	K=2	K=3	K=4
FX=2			97.17
FX=3	87.29	90.78	96.06
FX=4			85.07

optimal number of groups and range of values would be K=4 and FX=2 since it was the one that presented the best hit rate. However, in this [13] model, it was found that for this value of K and FX considered optimal, there was partial or complete overlapping of labels in at least two different groups of the same base, considering the same attribute. Therefore, this hit rate is disregarded when such situations occur, and the immediately lower one calculated so far is considered. Therefore, the optimal number of groups and hit rate for the Wine database becomes K=4 and FX=3. Table VI presents what this overlapping is.

TABLE VI. RESULT OF THE GROUP LABELING OF THE [14] MODEL FOR THE WINE DATABASE WITH K=4 AND FX=2

Grupo	Elementos	Rótulo		Análise
		Atributos	Faixa	Êxito (%)
0	66	Proline	276.6∼979.0	100
1	23	Alcohol Malic.Acid OD Proline	12.93∼14.83 0.73∼3.27 2.63∼4.0 979.0∼1680.0	100
2	57	Proline	276.6∼979.0	100
3	32	Malic.acid	0.73∼3.27	88.70

According to Table VI, for the [14] model, considering the Wine base with K=4 and FX=2, the following overlaps were found.

- r_{c_0} = (Proline, [276.6∼979.0]) e r_{c_2} = (Proline, [276.6∼979.0]);
- r_{c_1} = (Malic.Acid, [0.73∼3.27]) e r_{c_3} = (Malic.Acid, [0.73∼3.27]).

It means the proline attribute, in *clusters* 0 and 2, and the malic.acid attribute, in *clusters* 1 and 3, have the same range of values. Therefore, these values overlap and no longer represent a single group, making it difficult to interpret the label.

This method was performed for four databases of different sizes and quantities of attributes. It was found that the optimal number of K groups varies between the mode (referring to values from the Elbow, Silhouette, and Calinski-Harabasz methods) and the mode+1. The range of values varies from K-2 to K-1, according to Table VII. The Algorithm 1 summarizes this proposal in pseudocode.

TABLE VII. OPTIMAL NUMBER OF GROUPS AND RANGE OF VALUES

	Optimal Number	
Database	Clusters	Range of Values
Wine	4	3
Breast Cancer	3	2
Wine Quality White	3	2
Credit Card	3	2

Algoritmo 1: Method for Optimizing the Number of Clusters based on the Range of Attribute Values

```
Select initial K by the mode of group inference criteria (Elbow, Silhouette, and
    Calinski-Harabasz);
Make FX = K;
Display the average hit rate of [14]'s method for K between K-1 and K+1;
Display the average method hit rate of [14] for FX between FX-1 and FX+1;
Consider the pair (K, FX) with the highest hit rate of the method among those
    displayed;
while not finding optimal K and FX do
    if complete overlapping of labels in at least two groups of the same base
      then
        Discard the pair (K, FX);
        Search for the next pair (K, FX) whose hit rate is the second highest
    end
    else
        Set the value of K and FX, whose method hit rate is the highest.
    end
end
until Até encontrar K e FX ótimos.
```

Therefore, to develop a more optimized model about the interpretability of the group hit rate and the specificity of the generated labels, considering that no [14] model was found to overlap between ranges of values, assuming the same attribute in different groups, this research work presented a method based on dispersion metrics to solve this limitation of the overlap between ranges of values. The methodology used in this research work went through the following steps, as shown in Fig. 2:

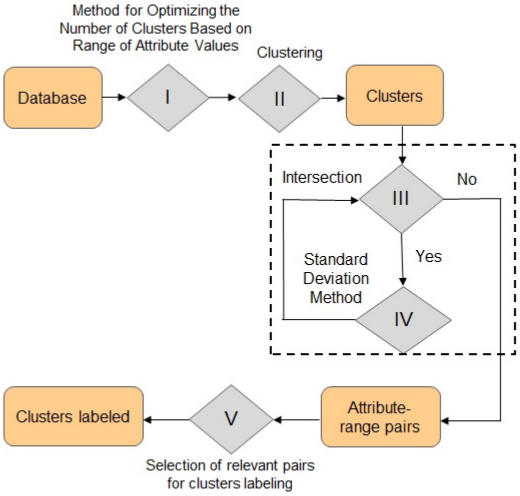

Method for Optimizing the Number of Clusters Based on Range of Attribute Values
Clustering
Database
I
II
Clusters
Intersection
III
No
Standard Deviation Method
Yes
IV
Clusters labeled
V
Attribute-range pairs
Selection of relevant pairs for clusters labeling

Fig. 2. Flowchart of the proposed labeling model.

A. Step 1 - Using the Value Range-Based Cluster Quantity Optimization Method

This step consists of using the proposed method based on the mode of the criteria found in the literature (Elbow, Silhouette Coefficient, and Calinski-Harabasz criterion) for the same data set to find the optimal number of clusters based on a range of values to be used in the data grouping and applied to four original databases of different sizes and quantities of attributes (Wine, Breast Cancer, Wine Quality White, and Credit Card), whose result can be seen in Table VII.

B. Step 2 - Clustering Data

After using the proposed method, the next step corresponds to the grouping of the data, which consists of submitting the original database composed of unlabeled examples to an algorithmic solution of unsupervised machine learning for the formation of groups. The basic idea is that elements that comprise the same group must present high similarity but are very dissimilar from objects in other clusters. The K-means algorithm was used for clustering, but any different clustering algorithm can be used.

C. Step 3 - Standard Deviation Method

Mathematically, standard deviation (SD) is a dispersion measure that is used to quantify the amount of variation or dispersion of a set of data values [11]. A low standard deviation value signifies that data points tend to be close to the mean of the set, while a high standard deviation indicates that data points are spread out over a wide range. Equations 2 show how standard deviation calculation is performed.

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_{\mathrm{i}} - M_{\mathrm{A}})^2}{n}} \tag{2}$$

For clarification:

- $\mathrm{x_i}$: value at the i position in the dataset.
- $\mathrm{M_A}$: arithmetic mean of the data.
- n: the amount of data.

Based on the use of standard deviation, a formal definition of the method for automatic cluster labeling is presented below:

Given a set of clusters $\mathrm{C_1}$, $\mathrm{C_2}$, $\mathrm{C_3}$...$\mathrm{C_k}$ and $\mathrm{A_1}$, $\mathrm{A_2}$, $\mathrm{A_3}$...$\mathrm{A_n}$ the attributes of this subset, the attribute's values distribution $\mathrm{A_1}$ in group $\mathrm{C_1}$ was represented by $\mathrm{A_1C_1(v_i,\ v_j)}$, where i indicates the minimum value and j the maximum value. The technique updates values by applying $\mathrm{v_i}$+SD, $\mathrm{v_j}$-SD, when necessary, considering as a condition exists or not of an intersection between values that $\mathrm{A_1}$ represents, observing the other groups.

Thus, after grouping performed by K-means, attribute values, observing their representation in each group, are organized into temporary structures in which indexes corresponding to the lowest and highest value are identified (Table VIII). Values contained in this table are random for purposes of the understanding method.

TABLE VIII. Indexing of Attribute Values

Attribute Values	8	5	10	6	7
Indexes	0	1	2	3	4
		min.	max.		
		Range: 5 ∼10			

At that moment, it was necessary to check possible overlapping values range in the same dataset, considering the same attribute.

D. Step 4 - Intersection Check

For each attribute, the existence or not of an intersection between values range is verified. If any overlap is found, the standard deviation of the segment under analysis is then calculated based on the arithmetic mean of distances between smallest and largest values, as seen in Table VIII. The updating of these values that identify the ends is applied, observing the need to increase the lowest value or decrease the highest value or even both procedures that, when performed, count as interactions. This process is carried out until there is no longer any intersection between the labels, considering the same attribute in all clusters in the dataset. Fig. 3 contains a representation of values the update applies to until they no longer overlap.

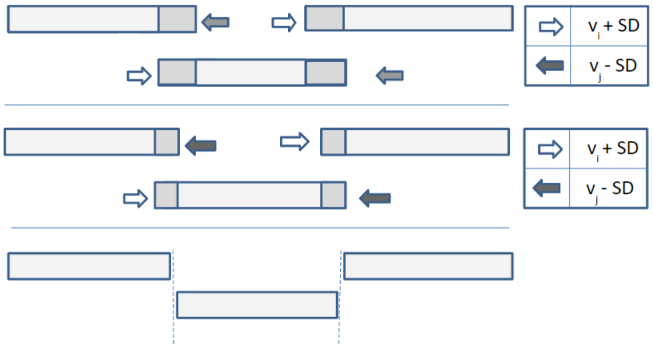


Fig. 3. Sequence of proposed updates.

Considering, for example, the $Petal\_length$ attribute of the Iris dataset, Table IX presents initial values range that represents this characteristic in different groups. In column 1 of this table, groups are identified, column 2 shows the range obtained with grouping, and in columns 3 and 4, operations must be performed when the intersection is found. For this purpose, the standard deviation value found in the respective distribution is applied as an updating factor.

TABLE IX. First SD Iteration for the $Petal\_length$ Attribute of the Iris Dataset

Clusters	Range	Decrement	Increment	SD
C0	3.0 ∼ 5.1	true	false	0.5
C1	1.0 ∼ 1.9	false	false	0.17
C2	4.9 ∼ 6.9	false	true	0.48

This same scenario is also illustrated in Fig. 4. Note that the analyzed attribute overlapped when considering its ranges, which justifies the decrement and increment in $C0$ and $C2$, respectively.

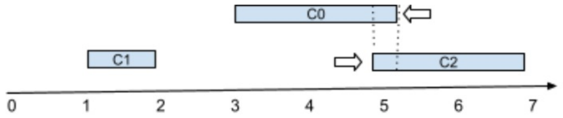


Fig. 4. Scenario described in table IV.

After performing this first iteration, Table X presents the new values resulting from the previous necessary update, also shown in Fig. 5.

TABLE X. SECOND SD ITERATION FOR THE $Petal\_length$ ATTRIBUTE OF THE IRIS DATASET

Clusters	Range	Decrement	Increment	SD
C0	3.0 ∼ 5.0	true	false	0.46
C1	1.0 ∼ 1.9	false	false	0.17
C2	5.0 ∼ 6.9	false	false	0.47

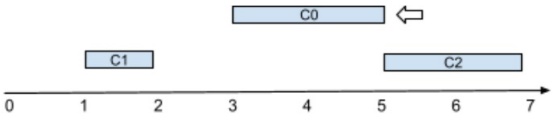


Fig. 5. Scenario described in table V.

Table XI presents data from this update process and verification of possible coincidences in interval segments. As described, it appears that the representation of attribute is distinct in observation of groups formed, thus dispensing with additional iterations. This process can be seen in Fig. 6.

TABLE XI. FINAL RESULT OF THE SD ITERATION FOR THE $Petal\_length$ ATTRIBUTE OF THE IRIS DATASET

Clusters	Range	Decrement	Increment	SD
C0	3.0 ∼ 4.9	false	false	2
C1	1.0 ∼ 1.9	false	false	0
C2	5.0 ∼ 6.9	false	false	1

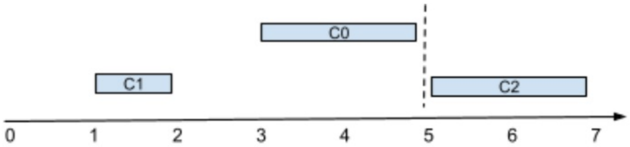


Fig. 6. Scenario described in Table VI.

Algorithm 2 summarizes this Cluster Labeling Proposal in pseudocode, where n is the number of attributes in the dataset, k is the number of groups, and C corresponds to a cluster.

Algoritmo 2: Pseudocode of Cluster Labeling Model Proposed

```
Input: K clusters
for attrA ← 1 until n do
    while there is an intersection of $att_A$ between two groups do
        $C_i \leftarrow 1$
        for $C_j \leftarrow 2$ until k do
            if max. of attrA in $C_i \geq$ min. of attrA in $C_j$ and max. of attrA in
              $C_i \leq$ max. of attrA in $C_j$ then
                decrement max. of $attr_A$ applying standard deviation of
                  $att_A$ distribution
            end
            if if min of attrA in $C_i \geq$ min attrA in $C_j$ and min. of $att_A$ in $C_i$
              $\leq$ max. $att_A$ in $C_j$ then
                increment min. of $att_A$ applying standard deviation of $att_A$
                  distribution
            end
        end
    end
end
```

E. Step 5 - Selection of the Relevant Attribute-Range Pair

This final step of labeling process consists of selecting relevant attribute–range pair for cluster labels composition, ensuring that each label represents specifically and exclusively one cluster. This selection was based on following measure:

1) Correlation Coefficient: Pearson's association [19] reflects direct relationship between two components, i.e. how much variables are associated, and reach out from +1 to - 1. An association of +1 suggests that there is an extraordinary positive direct relationship between elements, while an association of -1 demonstrates that elements have negative relationship. Table XII presents a categorization for Pearson's correlation coefficient values.

TABLE XII. CATEGORIZING FOR PEARSON'S CORRELATION COEFFICIENT VALUES

Correlation Coefficient	Classification
0	Null
0.01 - 0.19	Very weak
0.2 - 0.39	Weak
0.4 - 0.69	Moderate
0.7 - 0.89	Strong
0.9 - 0.99	Very Strong
1	Perfect

Therefore, the pair(s) of attributes that are most positively correlated in each data set are used in the composition of the final labels of the model.

V. RESULTS AND DISCUSSIONS

The results of applying the proposed method are presented with subsequent cluster labeling based on the correlation coefficient for the selection of the most relevant attributes and on the standard deviation metric to improve the specificity of the range of values of each selected attribute to eliminate overlaps between ranges of values, considering the same attribute in different groups.

In addition, the results obtained were compared with other methods proposed in the literature to show that the model guarantees an improvement in the specificity of the labels, reducing the computational effort to generate them.

A. Iris Dataset

Table XIII presents the analysis result for automatic rotation of the Iris database, for K=4 and FX=2 according to the proposed method. A label describes each cluster with a pair of attribute-value ranges, according to Pearson's association [19].

TABLE XIII. ANALYSIS FOR IRIS DATABASE LABELING

Cluster	Elements	Label		Analysis	Iterations
		Attributes	Range	Hits (%)	
0	28	SL	4.3∼6.1	98	2
		SW	2∼3.2	94	6
1	50	PL	0.99∼3.95	100	0
		PW	0.1∼1.3	100	0
2	32	SL	6.1∼7.9	95	5
		PW	1.3∼2.5	100	0
3	40	PL	3.95∼6.9	99	1
		SW	3.2∼4.4	94	6

Iterations correspond to steps to remove, when necessary, an intersection between clusters in the same dataset, considering the same attribute. It means that the number of iterations of the proposed method for each attribute can differ in each cluster, as it will depend on the dispersion of the group's elements about the mean of the data set. For cluster 0 and the SL attribute, for example, two iterations were necessary to obtain the result, while six iterations for the SW attribute of the same cluster were necessary.

The attribute pairs displayed are the ones that best correlate according to Pearson's correlation. The label obtained provides, as an aid to the specialist, the interpretation that:

- Cluster 0 is composed of plants whose sepal length (SL) varies between 4.3 cm and 6.1 cm, and the sepal width (SW) varies between 2 cm and 3.2 cm;
- Cluster 1 is formed by plants whose petal length (PL) varies between 0.99 cm and 3.95 cm, and petal width (PW) varies between 0.1 cm and 1.3 cm;
- Cluster 2 is composed of plants whose sepal length (SL) varies between 6.1 cm and 7.9 cm and petal width (SW) varies between 1.3 cm and 2.5 cm;
- Cluster 3 is composed of plants whose petal length (PL) varies between 3.95 cm and 6.9 cm and sepal width (SW) varies between 3.2 cm and 4.4 cm;

It was verified, therefore, that after executing the proposed method, fewer values are included as the range of values decreases, resulting in a non-overlap between labels generated by the same attribute in different clusters. This fact also occurred with the other data sets used.

The Table XIV displays the result of labeling the method of [20] for the Iris dataset. It was observed that the result was generated after 568 iterations of the method based on degrees of membership, which corresponds to a very high computational cost, in addition to having 12 database elements that could not be labeled. This model, proposed in this research, generated a maximum of six iterations to form labels, corroborating and reducing the computational effort. No criteria were used to infer the optimal number of groups in [20], so the author used K=3 for clustering.

TABLE XIV. Rótulos Gerados Por [20], Considerando a Base De Dados Iris

Cluster	Elements	Label		Analysis
		Attributes	Range	Hits(%)
1	50	PW	0.1 ∼0.6	100
		PL	1 ∼1.9	100
2	52	PL	3.5 ∼5.0	82.69
3	36	PL	5.1 ∼6.9	91.66

Table XV compares the models for the Iris dataset, considering the average hit rate of the labels, the number of attribute-value range pairs, and the maximum number of iterations that compose them.

TABLE XV. Comparison between the Labeling Models, Considering the Iris Database

Model	(%)	Attribute-Range Pairs	Maximum Iterations
Model of [20]	91.45	4	568
Proposed Model	97.5	8	6

In this proposed model, it was observed that the computational cost spent on forming the labels was extremely low, favoring a minimum number of iterations, compared to the [20] model, which presented a lower rate than this research model. Using the proposed method for inferring the optimal amount of *clusters* favored the obtained result. Also, in both methods, no range of attribute values overlapped for the same cluster when considering the same attribute.

B. Wine Dataset

Table XVI displays the analysis result for the Wine data set labeling for K=4 and FX=3 according to the proposed method. The clusters were labeled by alcohol and proline for clusters 0, 1, and 2 and by total phenols and flavanoids for cluster 3, as they are the attributes that best correlate according to the correlation coefficient, with a value of 0.86 for alcohol and proline and 0.64 for WK and LK.

TABLE XVI. Wine Database Labeling Analysis

Cluster	Elements	Label		Analysis	Iterations
		Attributes	Range	(%)	
0	66	Alcohol	12.72 ∼13.5	86.76	8
		Proline	690 ∼937	100	0
1	23	Alcohol	13.51 ∼14.83	86.60	13
		Proline	970 ∼1680	100	0
2	57	Alcohol	11.03 ∼12.7	91.22	7
		Proline	278 ∼590	100	0
3	32	Total phenols	0.98∼1.42	100	0
		Flavanoids	0.34∼1.25	100	0

The clusters were labeled by the attributes Alcohol and Proline and also Total phenols and Flavanoids, as they are the ones that best correlate according to the correlation coefficient and also have the highest rates. The interpretation given by the labels is that:

- Cluster 0 is composed of alcohol between 12.72 and 13.5 and proline between 690 and 937;
- Cluster 1 is composed of alcohol between 13.51 and 14.83 and proline between 970 and 1680;
- Cluster 2 is composed of alcohol between 11.03 and 12.7 and proline between 278 and 590;
- Cluster 3 comprises total phenols between 0.98 and 1.42 and flavanoids between 0.34 and 1.25.

The Table XVII shows the result of labeling the method of [14] for the Wine dataset. No criteria were used to infer the optimal number of groups in lopes, so the author used K=3 for clustering and FX=3 for data labeling.

TABLE XVII. Labels Generated by [14], Considering the Wine Database

Cluster	Elements	Label		Analysis
		Attributes	Range	(%)
0	62	Proline	628.5∼979	85.48
1	47	Proline	979∼1680	97.87
2	69	Proline	278∼628.5	100

Table XVIII compares the models in the Wine dataset, considering the average hit rate and the number of attribute-value range pairs that compose them.

It was found that the proposed model has a higher average hit rate than the [14] model, in addition to not having any overlap between ranges of values. This result was favored using the proposed method to infer the optimal number of *clusters*.

TABLE XVIII. COMPARISON BETWEEN THE LABELING MODELS, CONSIDERING THE WINE DATASET

Template	Wine	
	Average Hit Rate (%)	Attribute-Value Range Pairs
Model of [14]	94.45	3
Proposed Model	95.57	6

C. Seeds Dataset

Table XIX shows the analysis result for labeling the Seeds dataset for K=3 and FX=2 according to the proposed method. The clusters were labeled by perimeter (P) and area (A) for clusters 0 and 1 and by seed width (WK) and seed length (LK) for cluster 2, as these are the attributes that best correlate accordingly, with the correlation coefficient, with a value of 0.99 for P and A and 0.86 for WK and LK.

TABLE XIX. SEED DATABASE LABELING ANALYSIS

Cluster	Elements	Label		Analysis	Iterations
		Attributes	Range	(%)	
0	72	P	12.41 ∼13.78	91.04	7
		A	10.59 ∼13.07	89.56	9
1	61	P	13.82 ∼15.33	87.80	8
		A	13.19 ∼16.44	87.80	9
2	77	WK	3.465∼4.033	100	0
		LK	5.826∼6.675	100	0

The attribute pairs displayed are the ones that best correlate according to Pearson's correlation. The following interpretations can be drawn from the dataset labels:

- In cluster 0, elements have a perimeter (P) from 12.41 cm to 13.78 cm and an area (H) from 10.59 cm^2 to 13.07 cm^2;
- In cluster 1, the elements have a perimeter (W) from 13.82 cm to 15.33 cm and an area (H) from 13.19 cm^2 to 16.44 cm^2;
- In cluster 2, elements have seed width (WK) from 3465 to 4033 and seed length (LK) from 5826 to 6675.

The Table XX presents the result of labeling the method of [14] for the Seeds dataset. No criteria were used to infer the optimal number of groups in [14], so the author used K=3 for clustering and FX=3 for data labeling.

TABLE XX. LABELS GENERATED BY [14], CONSIDERING THE SEEDS DATABASE

Cluster	Elements	Label		Analysis
		Attributes	Range	Hits (%)
0	67	A	12.78 ∼16.14	87.30
		P	13.73 ∼15.18	
1	82	P	12.41 ∼13.73	86.58
		A	10.59 ∼12.18	
2	61	P	15.18 ∼17.25	98.34
		A	16.14 ∼21.18	
		LK	5.826 ∼6.675	
		WK	3.465 ∼4.033	

Table XXI addresses a comparison between models for the Seeds dataset, considering the average hit rate and the number of attribute-value range pairs that compose them.

The average hit rate of this proposed model is higher than the [14] method. No overlapping of the range of attribute values was verified in this model under analysis, in addition to having generated more accurate labels. It was observed that this result was favored due to the proposed method for inferring the optimal number of *clusters*.

TABLE XXI. COMPARISON BETWEEN THE LABELING MODELS, CONSIDERING THE SEEDS DATABASE

Model	Iris	
	Average Hit Rate(%)	Attribute-Range Pairs
Model of [14]	90.74	8
Proposed Model	92.70	6

D. Breast Cancer Dataset

Table XXII presents the analysis result for labeling the Breast Cancer data set for K=3 and FX=2 according to the proposed method. The clusters were labeled by Uniformity of Cell Size (UCS) and Uniformity of Cell Shape (UCSH) for cluster 0, Brand Chromatin (BC) and Uniformity of Cell Size (UCS) for cluster 1 and Brand Chromatin (BC) and Uniformity of Cell Shape (UCSH) for cluster 2, as they are the attributes that best correlate according to the correlation coefficient, with a value of 0.91 for UCSH and UCS, 0.76 for BC and UCS and 0.74 for BC and UCSH.

TABLE XXII. BREAST CANCER DATABASE DATA LABELING ANALYSIS

Cluster	Elements	Label		Analysis	Iterations
		Attributes	Range	Hits (%)	
0	455	UCSH	1 ∼4	97.40	4
		UCS	1 ∼2	82.18	3
1	108	BC	1 ∼4	100	0
		UCS	3 ∼10	98.68	2
2	120	BC	5∼10	100	0
		UCSH	5∼10	96.51	5

Although it is a large database with many attributes, the dispersion of the group elements about the dataset's average is small, which generated a few iterations about the Iris, Wine, and Seeds datasets. The label obtained provides the interpretation that:

- Cluster 0 is composed of elements whose Cell Shape Uniformity (UCSH) varies between 1 and 4 and Cell Size Uniformity (UCS) varies between 1 and 2;
- Cluster 1 is composed of elements whose Soft Chromatin (BC) varies between 1 and 4, and Cell Size Uniformity (UCS) varies between 3 and 10;
- Cluster 2 is made up of elements whose Soft Chromatin (BC) ranges from 5 to 10 and Cell Shape Uniformity (UCSH) ranges from 5 to 10.

The Table XXIII shows the result of labeling the method of [21] for the Breast Cancer dataset. No criteria were used to infer the optimal number of groups in [21], so the author used K=2 for clustering.

TABLE XXIII. LABELS GENERATED BY [21], CONSIDERING THE BREAST CANCER DATABASE

Cluster	Elements	Label		Analysis
		Attributes	Range	Hits(%)
0	232	UCS	1 ∼ 5	99.14
		MA	1 ∼ 10	99.14
		BN	1 ∼ 5.97	99.14
1	451	SECS	2 ∼ 10	99.11
		UCS	1.9 ∼ 10	99.11

Table XXIV presents a comparison between the models of the Breast Cancer data set, considering the average hit rate of the labels and the number of attribute-value range pairs that compose them.

TABLE XXIV. Comparison between the Labeling Models, Considering the Breast Cancer Database

Model	Iris	
	Average Hit Rate(%)	**Attribute-Range Pairs**
Model of [21]	99.13	5
Proposed Model	95.79	6

Despite a slightly lower average hit rate, the proposed model does not have overlapping labels, considering the same attribute. It was observed that in the model proposed by [21], there is an overlapping range of values, which compromises the interpretation of the label since the same label referring to the UCS attribute belongs to more than one *cluster*, that is is, UCS(c_1=[1∼5]) and UCS(c_2=[1.9∼10]).

Considering the four sets of data presented, it was verified that the proposed labeling approach does not offer any overlap between ranges of values of the same data set, considering the same attribute. In addition, the number of iterations was greatly reduced, favoring a low computational cost. It was also found in this proposed model that some ranges of values of certain attributes did not require iterations, given the lack of overlap between labels, when considering the same attribute. In three (Iris, Wine, Seeds) of the four data sets compared, the hit rate for the proposed labeling was higher, considering the use of the proposed method for inferring the optimal amount of *clusters* favored this result. The method considered in this paper is free of errors or biases.

VI. Conclusion

The group inference method developed in this research work proved to be satisfactory, considering that it was able to display an optimal number of clusters correlating the value of K to the range of attribute values, contributing to improving the data grouping process about other criteria existing in the literature separately, such as Elbow, Silhouette Coefficient, and Calinski-Harabasz Criterion.

Through labeling, this work provided an improved approach for group interpretation capable of automatically labeling data without overlapping any range of values in the same dataset, considering the same attribute and still with a reduced computational effort. This study initially used four data sets obtained from the UCI Repository, including Iris, Wine, Seeds, and Breast Cancer.

The results obtained in the experiments showed that the approach contributes to the groups' interpretation. The standard deviation-based labeling model also generated satisfactory results, with an average hit rate above 92% for the data sets. The model guarantees an improvement in the specificity of the labels, reducing the computational effort to generate them compared to other methods proposed in the literature.

References

[1] De Araujo, F. N., Machado, V. P., Soares, A. H.,& de MS Veras, R. (2018, July). Automatic cluster labeling based on phylogram analysis. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[2] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37-37.

[3] Russell, S. and Norvig, P. (2021). Artificial intelligence: A modern approach, fourth edition.

[4] Manning, C. D., Raghavan, P., & Schütze, H. (2009). Probabilistic information retrieval. Introduction to Information Retrieval, 220-235.

[5] Anaya-Sánchez, H., Pons-Porrata, A., & Berlanga-Llavori, R. (2008). A new document clustering algorithm for topic discovering and labeling. In Progress in Pattern Recognition, Image Analysis and Applications: 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Havana, Cuba, September 9-12, 2008. Proceedings 13 (pp. 161-168). Springer Berlin Heidelberg.

[6] Di, J., & Gou, X. (2018). Bisecting K-means Algorithm Based on K-valued Selfdetermining and Clustering Center Optimization. J. Comput., 13(6), 588-595.

[7] Kingrani, S. K., Levene, M., & Zhang, D. (2018). Estimating the number of clusters using diversity. Artificial Intelligence Research, 7(1), 15-22.

[8] Zhou, S., Xu, Z., & Liu, F. (2016). Method for determining the optimal number of clusters based on agglomerative hierarchical clustering. IEEE transactions on neural networks and learning systems, 28(12), 3007-3017.

[9] MacQuuen, J. B. (1967). Some methods for classification and analysis of multivariate observation. In Proceedings of the 5th Berkley Symposium on Mathematical Statistics and Probability (pp. 281-297).

[10] Pinheiro, J., Cunha, S., Gomes, G., and Carvajal, S. (2013). Probabilidade e estatística: quantificando a incerteza. Elsevier Brasil.

[11] Bland, J. M. and Altman, D. G. (1996). Statistics notes: measurement error. Bmj, 312(7047):1654.

[12] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297. Oakland, CA, USA.

[13] Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics-theory and Methods, 3(1), 1-27.

[14] Lopes, L. A., Machado, V. P., Rabêlo, R. A., Fernandes, R. A., & Lima, B. V. (2016). Automatic labelling of clusters of discrete and continuous data with supervised machine learning. Knowledge-Based Systems, 106, 231-241.

[15] Gustriansyah, R., Suhandi, N., & Antony, F. (2020). Clustering optimization in RFM analysis based on k-means. Indonesian Journal of Electrical Engineering and Computer Science, 18(1), 470-477.

[16] Solikhun, S., Yasin, V., & Nasution, D. (2022). Optimization of the Number of Clusters of the K-Means Method in Grouping Egg Production Data in Indonesia. International Journal of Artificial Intelligence & Robotics (IJAIR), 4(1), 39-47.

[17] Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In Machine Learning—EWSL-91: European Working Session on Learning Porto, Portugal, March 6–8, 1991 Proceedings 5 (pp. 164-178). Springer Berlin Heidelberg.

[18] Hwang, G. J., & Li, F. (2002). A dynamic method for discretization of continuous attributes. In Intelligent Data Engineering and Automated Learning—IDEAL 2002: Third International Conference Manchester, UK, August 12–14, 2002 Proceedings 3 (pp. 506-511). Springer Berlin Heidelberg.

[19] Thirumalai, C., Chandhini, S. A., and Vaishnavi, M. (2017). Analysing the concrete compressive strength using pearson and spearman. In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), volume 2, pages 215–218. IEEE.

[20] Imperes Filho, F., Machado, V. P., Veras, R. d. M. S., Aires, K. R. T., and Silva, A. M. L. (2020). Group labeling methodology using distance-based data grouping algorithms. Revista de Informática Teórica e Aplicada, 27(1):48–61.

[21] Silva, L. E. S., Machado, V. P., Araujo, S. S., Lima, B. V. A. d., and Veras, R. d. M. S. (2021). Using regression error analysis and feature selection to automatic cluster labeling. In EPIA Conference on Artificial Intelligence, pages 376–388. Springer.

Heart Disease Classification and Recommendation by Optimized Features and Adaptive Boost Learning

Pardeep Kumar, Ankit Kumar
Computer Science and Applications, Baba Mastnath University
Asthal Bohar, Rohtak, India

***Abstract*—In recent decades, cardiovascular diseases have eclipsed all others as the main reason for death in both low and middle income countries. Early identification and continuous clinical monitoring can reduce the death rate associated with heart disorders. Neither service is yet accessible, as it requires more intellect, time, and skill to effectively detect cardiac disorders in all circumstances and to advise a patient for 24 hours. In this study, researchers suggested a Machine Learning-based approach to forecast the development of cardiac disease. For precise identification of cardiac disease, an efficient ML technique is required. The proposed method works on five classes, one normal and four diseases. In the research, all classes were assigned a primary task, and recommendations were made based on that. The proposed method optimises feature weighting and selects efficient features. Following feature optimization, adaptive boost learning using tree and KNN bases is used. In the trial, sensitivity improved by 3-4%, specificity by 4-5%, and accuracy by 3-4% compared to the previous approach.**

Keywords*—*Heart disease prediction; heart disease; machine learning; optimization; multi-objective features

I. Introduction

The cardiovascular system, which also includes the lungs, is powered by the heart, a muscular organ which circulates blood throughout the body. The cardiovascular system includes a blood vessel network in addition to arteries, veins, and capillaries. Blood is distributed by these blood vessels all over the body. Cardiac disorders, also known as heart illnesses, are defined by deviations in the normal blood circulation of heart. The leading causes of death globally are heart disorders. Based on a survey conducted by the World Health Organization (WHO), strokes and heart attacks are responsible for 17.5 million deaths worldwide. Over 75% of deaths from heart disease happen in middle-income and low-income countries. In addition, strokes and heart attacks account for 80% of CVD-related mortality [1], [2]. In light of this, the mortality rate from cardiovascular problems can be reduced with the use of early recognition of cardiac abnormalities and prediction tools. Predictive models for cardiovascular disorders can now be developed with the help of the vast amounts of patient data that are readily available thanks to the expansion of modern healthcare infrastructure (i.e. Big Data inside the Electronic Health Records System). Machine learning is a technique for finding new information by analysing large datasets from several angles. Numerous records on patients' health, disease diagnoses, and other topics are created every day in the modern healthcare sector [3], [4], [5]. Many methods for unearthing similarities or hidden patterns in data can be found using machine learning [6], [7], [8], [9]. Machine learning has proved to be beneficial when it comes to making predictions and judgments based on the massive amounts of data collected by businesses in the healthcare industry [7], [8], [9], [10], [11]. Machine learning allows computers to automatically learn from data sets and improve their performance based on past experiences with little to no human input. Each time a ML algorithm makes a good call, it gets smarter. Consequently, in this research, we present a ML algorithm for the development of a cardiovascular disease forecasting tool.

A. GAP in Previous Work

The fundamental challenge with heart disease classification is that there is a limited dataset and just five classifications, thus learning efficiently is critical. In prior work, the following problem was discovered:

- Previous research has ignored feature overlaps and increased noise during learning [1], [2].
- Formerly, the emphasis was mostly on accuracy, which ranged from 40-50% in the case of five classes [4], [5].
- Do not improve the features based on their classification capacity [8], [9].
- Learning with a single classifier that is highly polynomial and increases over fitting [11].
- The majority of research is focused on binary categorization, yet this is not a true condition [12], [13].

B. Contribution of Research

- Apply entropy and information gain constraints to optimize features.
- Optimize feature selection and feature weights by using a genetic algorithm to maximize the Pareto surface.
- Work on feature-by-feature and weighted-features analysis of several performance metrics.
- In brief, optimize feature space and learning through optimizing classifiers.
- Focusing on five classes with high accuracy, sensitivity, and specificity.

II. Related Work

The paper gives an in-depth analysis of how ML can be used to treat cardiovascular disease. We also examine numerous popular literature on predicting the course of heart disease.

Ali et al. (2021) determines which machine learning classifiers provide the most accurate performance for diagnostic applications. Several supervised ML methods were implemented and compared in the prediction of heart disease. For all deployed algorithms except KNN and MLP, feature significance scores were assessed for each feature. All of the features were sorted based on their importance score to identify which ones offer the most reliable predictions of heart illness. Using a heart illness database from Kaggle and three-classification algorithms depending on KNN, DT, and RF, the analysis revealed that the RF method achieved 100% sensitivity, specificity, and accuracy. In this study, Katarya et al. (2021) summarized a portion of the expertise automated processes. Prediction and Feature selection are key components of every automated process. By selecting features effectively, one can attain improved heart disease prediction outcomes. The researchers have demonstrated useful methods for selecting attributes, including the hybrid grid search method and random search algorithm. As per Princy et al. (2020), a cardiac database is classified utilizing multiple cutting-edge Supervised ML algorithms for disease prediction. The findings show that the DT classifying model accurately diagnosed cardiovascular problems more so than the LR, NB, SVM, RF, and KNN approaches. 73% of the time, the Decision Tree produced the best outcome. This strategy could aid physicians in predicting the onset of heart problems and providing adequate treatment. Shah et al. (2020) offers several heart disease-related variables and a model based on supervised learning techniques like DT, NB, KNN, and RF. It utilizes the current database from the Cleveland dataset of UCI's heart disease patient repository. There are 76 attributes and 303 instances in the collection. For the purpose of verifying the efficacy of different approaches, only 14 of these 76 attributes are chosen for testing. The purpose of this report is to illustrate the occurrence of heart disease among patients. As per the results, K-nearest neighbour provides the highest accuracy. Sharma et al. (2020) makes a ML model that uses the relevant parameters to predict heart disease. The scholars used a standard UCI Heart disease prediction database for this research. This database has 14 key factors that are related to heart disease. For the creation of the model, ML techniques such as RF, SVM, DT, and NB, have been utilized. The research has also attempted to identify correlations between the numerous qualities present in the dataset by employing standard ML techniques and then employing these correlations to accurately forecast the likelihood of heart disease. When compared to other ML algorithms, the RF technique provides superior prediction accuracy and processing speed. The use of this system to aid in making decisions, this model may be beneficial to medical professionals in their clinic. Krishnan et al. (2019) used two supervised algorithms for data mining on a dataset to determine the likelihood of a patients experiencing heart disease, which were analyzed using classification models such as DT Classification and NB Classifier. These two algorithms were compared on a similar dataset to evaluate which one was the most accurate. The Decision Tree model accurately predicted the cardiovascular disease patient 91% of the time, while the Nave Bayes classifier correctly guessed the heart disease patient 87% of the time. Mohan et al. (2019) strategies and related cardiovascular disease prediction via hybrid ML techniques, with the purpose of discovering essential aspects by applying ML hence boosting the accuracy in the detection of cardiovascular illness. The expectation model consists of common feature groupings and their numerous permutations. The predictive model for cardiovascular illness with hybrid RF using a linear model allows the research teams to produce an improved exhibition level with a precision level of 88.7 percent (Table I). Individuals also informed about various data mining methods and assumption methods, for example, LR, KNN, NN, SVM, and Vote, which have recently been fairly popular in distinguishing and predicting heart disease. Santhana et al. (2019) detect cardiovascular disease in male patients using categorization approaches. This document offers exhaustive information on Cardiac Heart Diseases, including Risk Factors, Facts, and Frequent Type. WEKA seems to be the Data Mining tool used, and it is a great Computational Tool for Bioinformatics Fields. All three WEKA interfaces are used here; NB, ANNs, and DT are the main methods of data mining employed in this system to forecast heart disease. DTs such as C4.5, CART, CHAID, ID3, and J48 Algorithms, and NBs Techniques are commonly used for prediction. Gavhane et al. (2018) trained and examined the dataset using the multi-layer perceptron (MLP) neural network algorithm. Any number of input layers, output layers, and hidden units may be present in this algorithm. To achieve their desired effect, these hidden layers connect all input nodes to all output nodes. This bond is allocated weights. To achieve equilibrium in the perceptron, a second identity input, bias, with weight b, would indeed be introduced into the node. The nature of the nodes' connections to one another (feedforward or feedback) is determined by the task being performed. Li et al. (2018) have created an efficient ML-based approach for the diagnosis of cardiac disease. System design utilizes ML classifiers including ANN, K-NN, NB, SVM, and DT. Four classic feature selection methods, comprising MRMR, Relief, LLBFS, and LASSO, in addition, the issue of feature selection was addressed by employing a unique feature selection method. The system uses the LOSO cross-validation approach to select the optimal hyperparameters. The system is evaluated utilizing the Cleveland cardiovascular disease database.

III. Proposed System

A. Dataset

In experiment use ‘‘https://archive.ics.uci.edu/ml/datasets/heart+disease” data set for classification and recommendation in which total 303 instances, five classes and thirteen features (see Fig. 1 and 2).

Steps for Analysis

Entropy=$\sum_{j=1}^{N} P_i ly_s P_i$ .... (1)

ly_srepresent classes

$P_i$probability of Instance

Information gain=1-Entropy.................(2)

Step 1: Input heart disease dataset with features and labels.

Step 2: Features optimize by multi-objective optimization by this process given the efficient weight to features. In equation (3), E represent Entropy IG represent Information gain

TABLE I. ML PREDICTION OF A VARIETY OF HEART DISEASE AILMENTS

Ref	Year	**Aim**	Techniques	Feature/Tool	Dataset	Findings
[1]	[2021]	Model-based prediction of coronary heart disease using supervised ML	Supervised ML algorithms	Weka version 3.8.3	Kaggle	Accuracy rates of 100% were achieved by all three methods (RF, KNN, and DT).
[3]	[2020]	Using ML for early-stage prediction of heart disease	Supervised ML algorithms	Features: height, weight, Age, ap_hi ap_lo, gender, gluc, smoke, cholesterol, intake alco, cardio, active	Kaggle	The DT classification model outperformed Naive Bayes, RF, LR, KNN, and SVM in predicting cardiovascular illnesses.
[5]	[2020]	ML for Predicting Heart Disease	Supervised learning algorithms	WEKA tool	Cleveland database	According to the findings, KNN yields the best accuracy score.
[6]	[2020]	Prediction of cardiac events using ML	ML algorithms	WEKA tool	Cleveland heart disease database	RF gives more accurate predictions in less time.
[7]	[2019]	Hybrid ML techniques can accurately predict heart disease.	Hybrid RF with a linear model	Features: sex, Age, cp, chol, treetops, FBS, thalach, restecg, exang, olpeak, ca, slope, that, target	Cleveland database	HRFLM was quite accurate in predicting cardiovascular disease.
[8]	[2018]	Cardiovascular disease detection utilizing a real-time cardiac health surveillance system and ML algorithms.	ML algorithms	The WEKA data mining application version 3.8.2	Cleveland Heart Disease and Statlog Heart Disease dataset	The suggested feature selection approach is workable with SVM classifiers for building an advanced smart system for cardiac illness diagnosis.

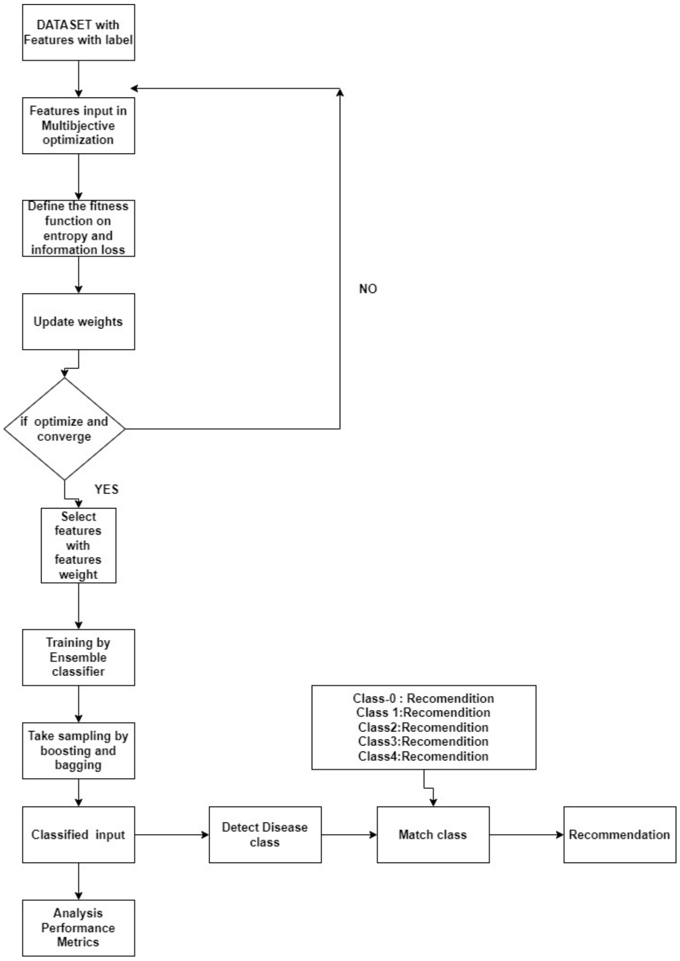

DATASET with Features with label
Features input in Multibjective optimization
Define the fitness function on entropy and information loss
Update weights
if optimize and converge
NO
YES
Select features with features weight
Training by Ensemble classifier
Take sampling by boosting and bagging
Classified input
Analysis Performance Metrics
Detect Disease class
Class-0 : Recomendition
Class 1:Recomendition
Class2:Recomendition
Class3:Recomendition
Class4:Recomendition
Match class
Recommendation

Fig. 1. Proposed classification and recommendation approach.

$$w_j^i = \begin{pmatrix} E & if\ E > IG \\ 0 & if\ IG = 0 \\ IG & otherwise \end{pmatrix} (3) \qquad (1)$$

Step 3: After crossover finish go to efficient Pareto space using following activation function or fitness function.

$$d\left(C_i, C_J\right) = \sum_{i=1}^{M} \left(W_i X_{i,J} - W_i X_i\right| \ldots\ldots\ldots\ldots..(4)$$

It finds the two-class distance and according to it finds Pareto space, here C_i, C_J are the classes, W_i are the weights as per the features.

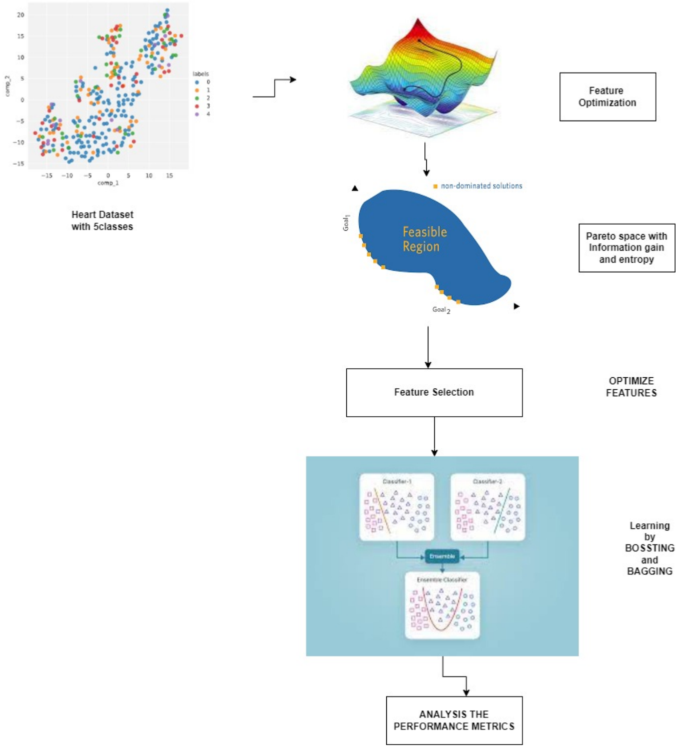


Fig. 2. Proposed classification approach.

Step 4: After finding the Pareto space optimize weight.

$$\delta^z = \frac{(\delta^z|^T.1}{\sum_{i=1}^{M}(\delta^K|^T.1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(5)$$

By equation (5) find the optimal solution in space of δ^z then

Step 5:

$$z = \underset{k}{max} \delta^k \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots...(6)$$

$$w = argmax\ (z) \ldots\ldots\ldots\ldots\ldots\ldots\ldots....(7)$$

By this find the maximize optimal weights o features

Step 6: After optimizing the weights of the weighted feature learn by classifier.

$$C_N\,(.) = \sum_{i=1}^{N} C_N * W_N\,(.) \ldots\ldots\ldots\ldots.(8)$$

$C_N\,(.)$Boosting Classifier

C_NN number of weak classifier

$W_N\,(.)$weight of features

After boosting all the possibilities send it to Bagging approach

$$B = \sum_{i=1}^{K} C_K\,(.) * \sum_{i=1}^{K} W_i \ldots\ldots\ldots\ldots.(9)$$

CM=$\alpha C_N\,(.)$+1-$\alpha\,(\delta_K) \ldots\ldots\ldots\ldots.(10)$

By (9) use bagging and use (10) for combining both develop a classification model. Here α is the learning parameter $\alpha[0,1]$

Step7: After step 6 recommendation part, according to Fig. 1, test one instance and according to predict class recommend the suggestion

IV. Experimental Results and Discussion

. Adaptive Boosting KNN with 60% accuracy outperforms Adaptive Boosting-Tree (54.34%), Adaptive Boosting (45.34%), KNN (40.2%), and SVM (40.11%) while using 5 features. Adaptive Boosting KNN with 70% accuracy outperforms Adaptive Boosting-Tree (50.12%), SVM (41.2%), Adaptive Boosting (40.12%) [2], and KNN (34.34%) [3] when using 6 features. With 8 features, however, Adaptive Boosting KNN (67.34%) achieves the highest accuracy, followed by Adaptive Boosting (47.44%), Adaptive Boosting-Tree (45.23%), KNN (42.12%), and SVM (40.34%). Similarly, with 10 features, Adaptive Boosting KNN (65%) achieves the highest accuracy, followed by Adaptive Boosting (48.12%), Adaptive Boosting-Tree (47.34%), SVM (39.5%), and KNN (36.44%). Adaptive Boosting KNN delivers superior accuracy with 12 and 13 features compared to previous approaches (Table II).

The comparison of the various features depending on their degrees of accuracy is shown in Fig. 3. When compared to other approaches, the accuracy that Adaptive Boosting KNN provides is far superior (Table III).

hows the sensitivity of features derived from various approaches. Adaptive Boosting KNN with 96.23 sensitivity

```
Proposed Algorithm
Input dataset with features and label
1. Mutation and crossover by eq. (3)
2. Extract newly generated vector
3. Update the fitness function of eq. (4) if optimize then go to the next step else go to the $3^{rd}$ step
4. Optimize fitness function and find the optimize Pareto space weight by eq. (5) and get weights eq. (6) and eq. (7)
5. Learning by Boosting eq. (8) and Bagging eq. (9)
6. Make Classifier model eq. (10) and analysis
7. Output <-Accuracy, Precision and Recall
```

TABLE II. Accuracy of Features based on Different Methods

Features	KNN	SVM	Adaptive Boosting	Adaptive Boosting-Tree	Adaptive Boosting KNN
5	40.2	40.11	45.34	54.34	60
6	34.34	41.2	40.12	50.12	70
8	42.12	40.34	47.44	45.23	67.34
10	36.44	39.22	48.12	47.34	65
12	43.22	40	43.23	50.12	56
13	35.5	41.34	43.2	52.33	60

TABLE III. Sensitivity of Features based on Different Methods

Features	KNN	SVM	Adaptive Boosting	Adaptive Boosting-Tree	Adaptive Boosting KNN
5	73.45	78.34	88.23	90.1	96.23
6	72.43	74.34	88.34	92.3	94.23
8	70.23	74.35	84.56	90	93.45
10	71.33	73.45	88.23	87.3	92.34
12	72.33	70.12	87.34	89.13	90.23
13	71.1	70.32	84.3	86.12	93

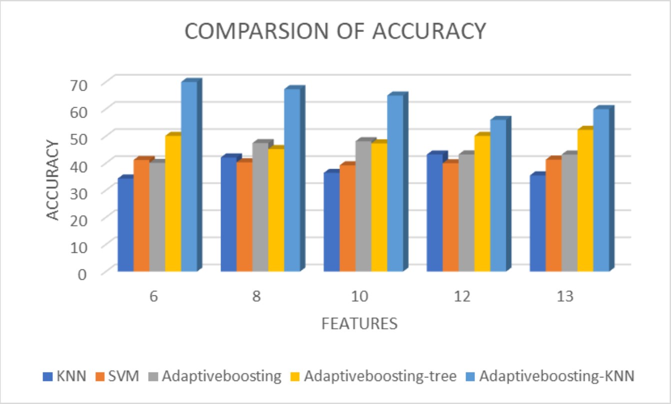


Fig. 3. Accuracy-based comparison.

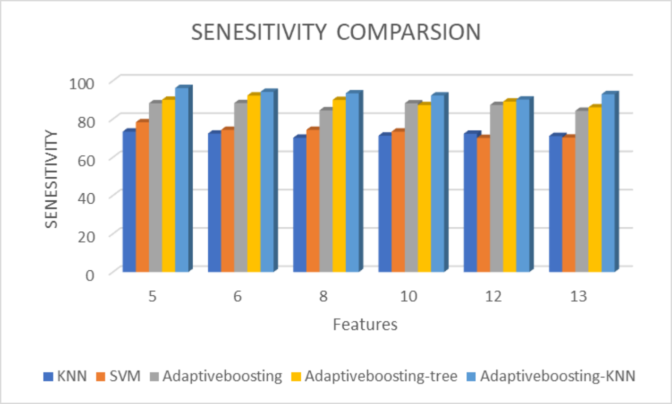


Fig. 4. Sensitivity-based comparison.

value outperforms Adaptive Boosting-Tree (90.1), Adaptive Boosting (88.23), SVM (78.34), and KNN (73.45) while using 5 features. Adaptive Boosting KNN with 94.23 sensitivity outperforms Adaptive Boosting-Tree (92.3), Adaptive Boosting (88.34), SVM (74.34), and KNN (72.43) when using 6 features. With 8 features, however, Adaptive Boosting KNN (93.45) achieves the highest sensitivity, followed by Adaptive Boosting Tree (90), Adaptive Boosting (84.56), SVM (74.35), and KNN (70.23). Similarly, with 10 features, Adaptive Boosting KNN (92.34) achieves the highest sensitivity, followed by Adaptive Boosting-Tree (87.3), Adaptive Boosting (87.34), SVM (73.45), and KNN (71.33). Adaptive Boosting KNN delivers superior sensitivity with 12 and 13 features compared to previous approaches.

Fig. 4 shows a comparison of the features according to their sensitivities. Adaptive Boosting KNN offers significantly higher sensitivity than the other competing methods (Table IV).

hows the specificity of features derived from various approaches. Adaptive Boosting KNN with 60.0 sensitivity value outperforms Adaptive Boosting-Tree (56.23), Adaptive Boosting (54.23), SVM (50.23), and KNN (45.12) while using 5 features. Adaptive Boosting KNN with 70.23 specificity outperforms Adaptive Boosting-Tree (60.23), Adaptive Boosting (53.12), KNN (46.23), and SVM (45.12) when using 6 features. With 8 features, however, Adaptive Boosting KNN (75.23) achieves the highest specificity, followed by Adaptive Boosting Tree (69.12), Adaptive Boosting (59.12), KNN (50.12), and SVM (42.34). Similarly, with 10 features, Adaptive Boosting KNN (60.13) achieves the highest specificity, followed by Adaptive Boosting-Tree (55.23), Adaptive Boosting (53.23), KNN (52.34), and SVM (40.12). Adaptive Boosting KNN delivers superior specificity with 12 and 13 features compared to previous approaches.

Fig. 5 shows a comparison of the features according to their specificities. Adaptive Boosting KNN offers significantly higher specificity than the other competing methods.

TABLE IV. Specificity of Features based on Different Methods

Features	KNN	SVM	Adaptive Boosting	Adaptive Boosting-Tree	Adaptive Boosting KNN
5	45.12	50.23	54.23	56.23	60
6	46.23	45.12	53.12	60.23	70.23
8	50.12	42.34	59.12	69.12	75.23
10	52.34	40.12	53.23	55.23	60.13
12	55.23	35.12	68.23	70.12	72.34
13	56.12	40.12	58.12	60.12	62

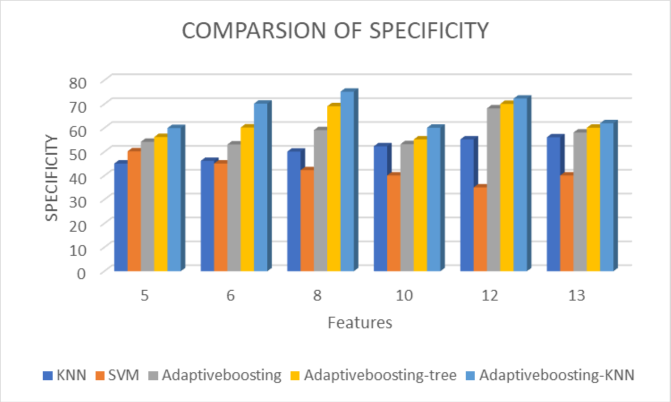


Fig. 5. Specificity-based comparison.

V. Observation of Results

- In the results, we compared existing and proposed adaptive boost approaches. There are three variants of adaptive boost in the results: one is basic adaptive boost, another is hybridized with tree, and the third is hybridized with KNN.
- Using multi–objective genetic optimization, features are given the appropriate weighting in all of the proposed methods. It makes sure that features don't overlap and boosts performance, as shown in the figures above.
- Adaptive boost tree improves all measures of performance because entropy and information gain map well on tree-based approaches.
- By maximizing performance improvement in sensitivity, the proposed model's recall value is raised.

VI. Conclusion

The long-term preservation of people's existence and the early detection of irregularities in heart problems will be made possible by recognizing the processing of primary health records of heart data. In order to process the raw data and deliver a new and unique insight towards heart disease, methods based on machine learning were applied in this study. Prediction of heart disease is difficult and crucial in the medical industry. However, if the disease is discovered in its initial stages and preventive measures are implemented as soon as feasible, the fatality rate can be significantly reduced. The proposed approach employs a five-class classification system to improve the diagnosis of specific heart disease and the subsequent recommendation. As a result, improving classification sensitivity is a significant task. Sensitivity is improved through feature optimization, and ensemble learning is enhanced through bagging and boosting. In comparison to traditional SVM and KNN methods, a 5% gain in sensitivity is highly significant.

In future, we enhance this work using non-linear mapping by deep learning approach and make optimize latent space for reducing overlapping between classes

References

[1] S. Nashif, M. R. Raihan, M. R. Islam, and M. H. Imam, "Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system," *World Journal of Engineering and Technology*, vol. 6, no. 4, pp. 854–873, 2018.

[2] E. Maini, B. Venkateswarlu, and A. Gupta, "Applying machine learning algorithms to develop a universal cardiovascular disease prediction system," in *International Conference on Intelligent Data Communication Technologies and Internet of Things*. Springer, 2018, pp. 627–632.

[3] M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. Quinn, and M. A. Moni, "Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison," *Computers in Biology and Medicine*, vol. 136, 2021.

[4] R. Katarya and P. Srinivas, "Predicting heart disease at early stages using machine learning: a survey," *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 302–305, 2020.

[5] R. J. Princy, S. Preetha, P. Parthasarathy, A. R. S. H. Jose, S. Lakshminarayanan, and Jeganathan, "Prediction of cardiac disease using supervised machine learning algorithms," *2020 4th international conference on intelligent computing and control systems (ICICCS)*, pp. 570–575, 2020.

[6] D. Shah, S. Patel, and S. K. Bharti, "Heart disease prediction using machine learning techniques," *SN Computer Science*, vol. 1, no. 6, pp. 1–6, 2020.

[7] V. Sharma, S. Yadav, and M. Gupta, "Heart disease prediction using machine learning techniques," *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 177–181, 2020.

[8] S. Mohan, , G. H. Thirumalai1, and Srivastava, pp. 81 542–81 554, 2019.

[9] A. Gavhane, G. Kokkula, I. Pandya, and K. Devadkar, "Prediction of heart disease using machine learning," *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pp. 1275–1278, 2018.

[10] S. Krishnan and S. Geetha, "Prediction of Heart Disease Using Machine Learning Algorithms," *2019 1st international conference on innovations in information and communication technology (ICIICT)*, pp. 1–5, 2019.

[11] S. Nashif, M. R. Raihan, M. R. Islam, and M. H. Imam, "Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system," *World Journal of Engineering and Technology*, vol. 6, no. 4, pp. 854–873, 2018.

[12] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, pp. 3–21, 2020.

[13] J. Li, Ping, A. U. Haq, J. S. U. Din, A. Khan, A. Khan, and Saboor, "Heart disease identification method using machine learning classification in e-healthcare," *IEEE Access*, vol. 8, pp. 107 562–107 582, 2020.

SSEC: Semantic Segmentation and Ensemble Classification Framework for Static Hand Gesture Recognition using RGB-D Data

Dayananda Kumar NC[1], K.V Suresh[2], Chandrasekhar V[3], Dinesh R[4]
Dept. of Electronics and Communication Engineering[1,2,3]
Siddaganga Institute of Technology, Tumkur, India[1,2]
IIIT Sricity, India[3]
Dept. of Information Science and Engineering, Jain University, Bangalore, India[4]

***Abstract*—Hand Gesture Recognition (HGR) refers to identifying various hand postures used in Sign Language Recognition (SLR) and Human Computer Interaction (HCI) applications. Complex background in uncontrolled environmental condition is the major challenging issue which impacts the recognition accuracy of HGR system. This can be effectively addressed by discarding the background using suitable semantic segmentation method, where it predicts the hand region pixels into foreground and rest of the pixels into background. In this paper, we have analyzed and evaluated well known semantic segmentation architectures for hand region segmentation using both RGB and depth data. Further, ensemble of segmented RGB and depth stream is used for hand gesture classification through probability score fusion. Experimental results shows that the proposed novel framework of Semantic Segmentation and Ensemble Classification (SSEC) is suitable for static hand gesture recognition and achieved F1-score of 88.91% on OUHANDS test dataset.**

Keywords*—*Hand gesture recognition; semantic segmentation; ensemble classification; score fusion

I. Introduction

Hand gestures plays significant role in many real time applications like robotics control, gaming, 3D modeling, virtual environment etc,. Various methods are used to detect and recognize the hand gestures depending on the data acquisition and processing system [1] [2]. Hand gesture recognition systems can be broadly classified into sensor based and vision based systems.

In sensor based systems, data is captured using the sensor modules connected to the hand which converts the hand movements into varying time series signal. These devices captures the hand movement data very precisely but not ease for usage as it impose the constraint to wear the device and not supporting the contact less operation [3] [4].

In vision based systems, RGB cameras are widely used to capture the hand pose data as color images. Vision based systems involve hand detection or segmentation as one of the important pre-processing step involved in gesture recognition pipeline to localize the hand region and discard the background in the image [5]. Feature extraction methods are used on the segmented hand data to obtain its characteristic representation which are used in classification stage to effectively recognize the various hand gestures. Current state-of-the-art recognition systems based on the color image data face many challenges in hand segmentation and recognition due to the complex background and varying illumination conditions. Gestures performed by various subjects differing in their hand size and color is difficult to identify due to large intra-class and inter-class variations. Addressing these issues is difficult by using the color modality data. Hence recent systems are developed on dual modality using RGB and Depth data known as RGB-D data. In these systems, data is simultaneously captured using both color and depth sensor to obtain the RGB-D aligned data pair registered on the same view of camera coordinate system [6].

Depth map can be obtained using various techniques like stereo, time-of-flight of IR etc., where it provides the distance information between depth sensor and the object scene [7]. Kinect device sensor uses time-of-flight between the emitted IR light and the reflected light on projector to provide raw depth map in which each pixel location represents the distance in millimeter. Depth modality can be effectively used to discard the far away background based on depth distance range. It helps in effective hand segmentation to group hand region pixels into foreground and rest of the image pixels into background [8]. Also in case of low light scenarios RGB sensors fails to capture the data, this issue can be resolved using Kinect depth sensor as it captures the data using IR light.

Various image processing and computer vision algorithms are discussed in literature for hand region segmentation. According to recent studies, CNN architectures are widely used to address various real time problems and segmentation is one of the majorly studied area. CNN based semantic segmentation networks performs pixel level localization of region of interest where it classifies each pixel into its corresponding segmentation class. It provides the fine boundary of each distinct region mapped to the unique segmentation class.

In this paper, we analyzed various segmentation methods on RGB and depth data and provided comparative analysis to identify the best suitable method for hand segmentation.

The organization of the paper is as follows. In Section II, a brief review of different methods that exist in vision based hand gesture recognition is presented. In Section III, problem statement and the proposed method is discussed. In Section IV, detailed experimental results is presented. Experimental outcomes are briefly discussed in Section V. Finally, conclusion is drawn in Section VI mentioning the limitation of current work and scope of future work.

II. Literature Review

In this section, we discuss on the state-of-the-art methods for hand region segmentation and gesture classification along with their advantages and limitations.

Earlier approaches of hand segmentation in color image were based on color intensity thresholding in RGB, HSV, YCbCr and other color spaces [9]. In these methods suitable color range was identified based on the experimentation to segment the skin region. Limitation of this approach is difficulty in selecting the threshold range to segment all variation of skin color and the lighting variations which significantly affect the segmentation accuracy.

Hand region segmentation based on human skin tones was proposed in [10] using an MLP network to learn the skin color tones and classify the pixels of image which belongs to the skin color sets.

User independent recognition system using low-cost Microsoft Kinect depth sensor was proposed in [11] to overcome illumination and background variations issue in color-based sign language recognition. Here hand region was segmented by using a pre-processing algorithm on depth image. Features are extracted from hand segmented data using CNN based unsupervised Principal Component Analysis Network (PCANet) and classified using Support Vector Machine (SVM) classifier.

Real-time hand gesture recognition method was put forth in [12] using light-weight semantic segmentation method (FASSD-Net) to produce hand segmentation masks which are combined with RGB frames in gesture classification using Temporal Segment Networks (TSN) and Temporal Shift Modules (TSM) tested on IPN Hand dataset.

Various interactive methods like Graph cut, Random walker, geodesic star convexity etc., were analyzed in [13] for hand region segmentation. Five distinct types of hand motions in various backdrops were tested using the Expectation Maximum technique to learn the parameters of the Gaussian Mixture Model and the Gibbs random field to image segmentation by minimising the Gibbs Energy using the Mincut theorem. According to experimental findings, utilising manually segmented photos improves recognition accuracy when compared to unsegmented images.

Bin et. al [14] proposed a fine-tuned Inception V3 RGB-D static gesture recognition method. This framework eliminates the gesture segmentation and feature extraction steps in traditional algorithms. The proposed framework consists of a CNN architecture in which feature concatenate layer concatenates the features of RGB and depth images. Compared with general CNN, the Inception V3 based gesture recognition resulted in improved accuracy.

D Kumar et. al [15] proposed a two stage approach for static hand gesture recognition using RGB-D data. In first stage k-means clustering algorithm is applied on the depth image to cluster the foreground and background depth pixels based on the distance. Depth threshold is computed as the mean of cluster centers and using this dynamic threshold background is discarded. In classification stage, segmented RGB-D data is stacked to form the input to data layer of custom CNN network.

Coarse to fine segmentation approach using depth map was proposed in [8] where pre-trained YOLO-v3 model was used to detect and localize the hand region at coarse level. The hand detected bounding region was used to initialize the foreground in graph cut segmentation algorithm which refines the hand region boundary and discards the background. Hand segmented RGB-D data was further used in classification stage to recognize the hand gestures.

The hand region in the depth map was segmented using the depth thresholding approach in [16]. Additionally, a two stream network with AlexNet and VGG16 was employed using score-level fusion technique to recognise the static hand gestures from the datasets from Massey University (MU) and HUST American Sign Language (HUST-ASL) with accuracy of 98.14 % and 64.55 % respectively.

Hand Gesture Recognition Approach called HGRA on RGB data using two stream was proposed by [17], in first branch U-Net combined with Multi-Scale Attention module is used to segment the hand region and extracting shape features. In second branch, Multi-Scale Fusion (MSF) and Light-Weight Multi-Scale (LWMS) modules are used to extract multi-scale appearance and color features. This method was evaluated on OUHANDS and HGR1 datasets and achieved the accuracy of 90.9% and 83.8% respectively.

Three stage spatial attention-based neural network was proposed in [18]. First two stages include generation of feature vector and attention map with the feature extraction architecture and self-attention technique. Final feature is generated after multiplying the features and attention map and feed to classification module in third stage to predict the label of hand gesture. This model achieved 99.75%, 99.46% and 99.67% accuracy in Kinematic, NTU and senz3D datasets respectively.

Dual-stream dense residual fusion network(DeReFNet) was proposed in [19] which utilizes the strength of global features and spatial information from the residual stream and other stream. Both the streams are fused using the feature concatenation module. Subject-independent cross-validation technique is used to validate DeReFNet four publicly available benchmark datasets.

Kinect sensor device is used to capture hand gesture depth images. Serial binary image extraction is used in [20] to eliminate the undesired shadow region in depth image and improve the recognition accuracy using VGG-type CNN. Emergence of industry 4.0 with need of natural human-robot interaction in manufacturing using vision-based and wearable-based approaches for gesture-based interaction is discussed in [21]. Position data from Microsoft Kinect RGB-D cameras and acceleration data from inertial measurement units (IMUs) is compared to evaluate the recognition accuracy.

Based on the brief literature review it can be observed that most of the recent research in hand gesture recognition use RGB-D data. Early methods of hand region segmentation used skin color based segmentation in different color space, later CNN based semantic segmentation methods gained much attention due to its efficiency and robustness even in complex scene. Depth modality can be used both in segmentation and classification, hence active research is being carried out in state-of-the art methods to evaluate various ensembling and fusion techniques of RGB and Depth modalities [22]. In further section, we discuss about the details of proposed method and experimental analysis.

III. Proposed Method

Based on the literature review, it is evident that hand gesture recognition is still an active area of research trying to solve the challenges of gesture recognition is real scenarios with complex background scene and varying lighting conditions. Current research methods have also showed that multi-modal RGB and depth stream data is effective than uni-modal RGB data for hand gesture recognition.

In this paper, we analyze various semantic segmentation methods to effectively segment the hand region using RGB and Depth stream data. Hand segmented RGB and Depth data are further used to train custom CNN model for gesture classification. Suitable approach for fusing the probability scores from both the models are analyzed and proposed ensemble classification framework for static hand gesture recognition.

The main contributions of this paper are:

1) Analysis of semantic segmentation model accuracy using RGB data, depth data and combined RGB-D data.
2) Proposed the ensemble approach of score fusion for static HGR classification on RGB and Depth data.

A. Semantic Segmentation

Semantic segmentation is a pixel-based classification in which each pixel of an image is classified to its corresponding class. Here the class labels of all the pixels of image are predicted, hence segmentation is also termed as dense prediction. Hand region segmentation is a binary case of segmentation which has two output class and provides the pixel level mapping into required foreground and background regions as in Fig. 1. In this work, we evaluate various CNN architectures like UNet, ResUNet and DeeplabV3-Plus for semantic segmentation of hand region on both RGB and Depth data.

B. UNet

U-Net architecture [23] adopts auto-encoder framework which consists of two components known as encoder and decoder as in Fig. 1. Encoder generates the compressed feature representation of the image using down sampling and strided convolution, these features contribute in classifying the pixels into its corresponding segmentation class. The encoder and decoder layers are symmetrical to each other. Decoder includes up-sampling and transpose convolution which generates the output segmentation map which has the same resolution as the input image to segmentation model. The least squares reconstruction error is back propagated from the decoder to encoder using which the weights are updated to obtain optimal feature representation.

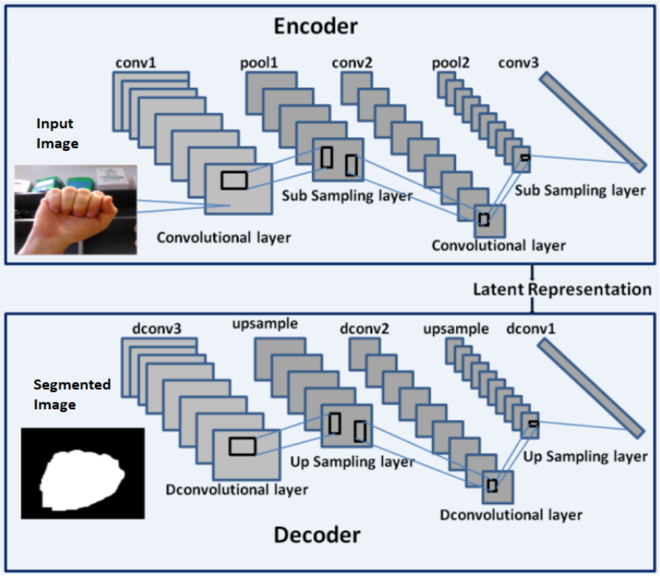


Fig. 1. Auto-encoder based segmentation network.

Encoder generally have the following sub layers, Convolution layer, Relu activation layer and pooling layer. The input image is fed in to the data layer followed by convolution layer which consists of filter of size 3x3 followed by Relu activation to add non-linearity. In the subsequent layers the number of kernels is doubled with constant kernel size and the max pooling layer is used to reduce or down sample the feature map and to maintain the local dominant features in the image patch. Here we have modified original architecture by removing last block of convolution layers and used only three blocks which consists of two convolution layers in each block for model convergence.

Decoder is used to reconstruct the input image using the reduced representation from encoder layer. Encoded input images are decoded by a series of up sampling and de-convolution block. The up-sampling operation of the decoder layers use the max-pooling indices of the corresponding encoder layers. The decoder architecture follows certain pattern based on its encoder design, where the decoder is mirror replica of encoder. The decoded image is evaluated against the input image while self learning the feature representation.

C. ResUNet

Residual U-Net [24] is a semantic segmentation network in which the residual blocks are used in encoder and decoder block of U-Net architecture. This residual learning helps to improve the U-Net results and only with fewer parameters. Fig. 3 shows the basic unit blocks of U-Net (a) and ResUnet (b). Each residual unit can be mathematically shown as in Eq.1.

$$\begin{aligned} y_l = h(X_l) + F(X_l, W_l), \\ X_l + 1 = f(y_l) \end{aligned} \tag{1}$$

where X_l and $X_l + 1$ are the input and output of the l^{th} residual unit, F is the residual function, $f(y_l)$ is activation function and $h(X_l)$ is a identity mapping function, where $h(X_l) = X_l$.

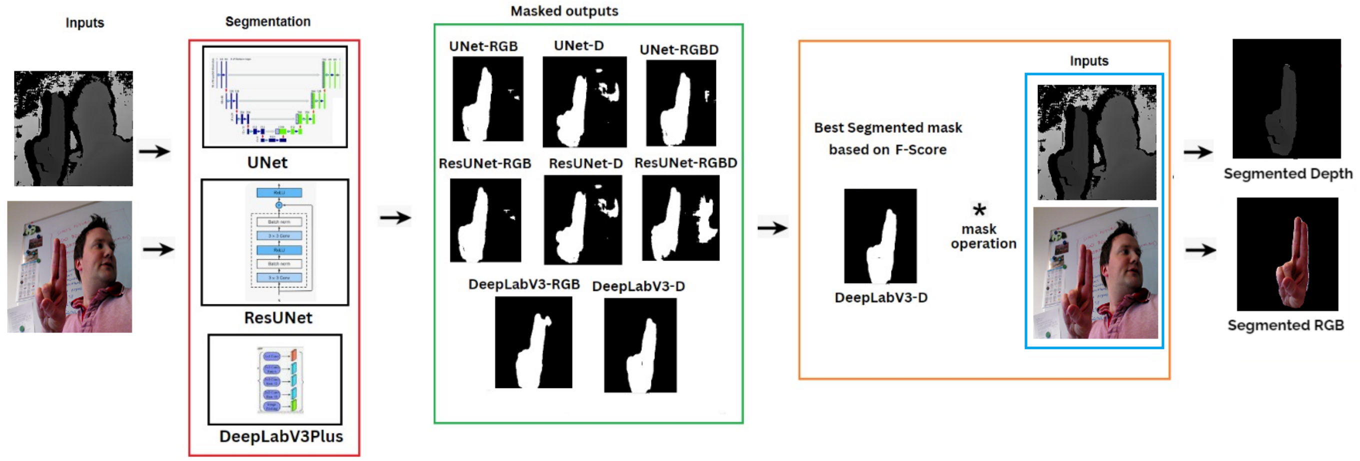


Fig. 2. Block diagram of RGB-D semantic segmentation analysis.

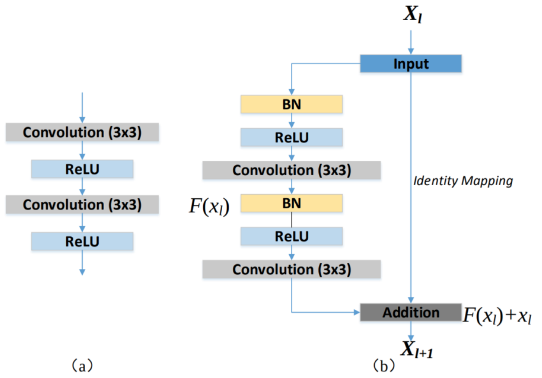


Fig. 3. (a) U-Net unit block and (b) ResUnet unit block.

ResUnet comprises of three parts built with residual units, encoding, bridge and decoding. The first part encodes the input image into latent feature representation. Second part forms a bridge connecting the encoding and decoding paths. Third part of decoding provides the semantic labels to each pixel by pixel-wise classification. Each residual units consists identity mapping and two 3 × 3 convolution blocks with BN layer and ReLU activation layer. The identity mapping connects the inputs and outputs of the unit. Decoding path consists of three residual units and concatenated with the feature maps from the corresponding encoding path. After the last level of decoding path, a 1 × 1 convolution and a sigmoid activation to obtain desired segmentation.

D. DeepLab-V3+

DeepLabv3+ [25] is the extended version of DeepLabv3 segmentation architecture. It follows encoder-decoder structure with Atrous Spatial Pyramid Pooling (ASPP) module in the encoder block. Hence encoder module processes multi-scale contextual information at multiple rates and multiple effective fields-of-view by applying dilated convolution at multiple scales. The decoder module with depthwise separable convolution refines the segmentation results along object boundaries by gradually recovering the spatial information.

Fig. 2 shows the block diagram of segmentation architecture analysis. UNet and ResUnet models are trained on RGB, Depth and stacked RGB-D data, DeepLabv3+ is trained on RGB and Depth data. All these models are trained separately and results are analyzed using mean IoU (Intersection over Union), average F1-score metrics. DeepLabv3+ trained on depth data gave better accuracy with comparatively less parameters, hence this model is selected as the best model for hand region segmentation. Predicted binary segmentation mask from this model is combined with RGB and depth data to obtain segmented RGB and segmented Depth data, this data is further used as input to the classification model.

E. Ensemble Classification

Classification block diagram is depicted in Fig. 4, where two classification models using segmented RGB and segmented depth data stream are trained independently using the custom CNN network as shown in Table I. Further, the classification probability of segmented RGB and depth model are analyzed and fused using max and average operator to select the best classification model.

Custom CNN-Net architecture in Table I consists of four groups with two layers of convolution CONV2D and RELU activation, Batch normalization and Max pooling. The number of kernels is increased as [16, 32, 64, 128] in subsequent groups. Global average pooling layer is used to get final feature map, two fully connected layers are used followed by Softmax activation to get the probability output of each class.

Model is trained using Adam optimizer with learning rate of 0.001 and categorical cross-entropy is used as loss function. Batch normalization and drop out layers are used avoid the model from over-fitting. RGB and depth data from OUHANDS dataset is resized to 320 x 320 and segmented using the binary mask obtained from Deeplabv3+ depth segmentation model.

Two classification models are trained using the segmented RGB and depth data, these models are evaluated on the test data and the miss-classified images are analyzed. It is observed that some of the images that were wrongly classified in the RGB stream are detected properly in the depth stream and

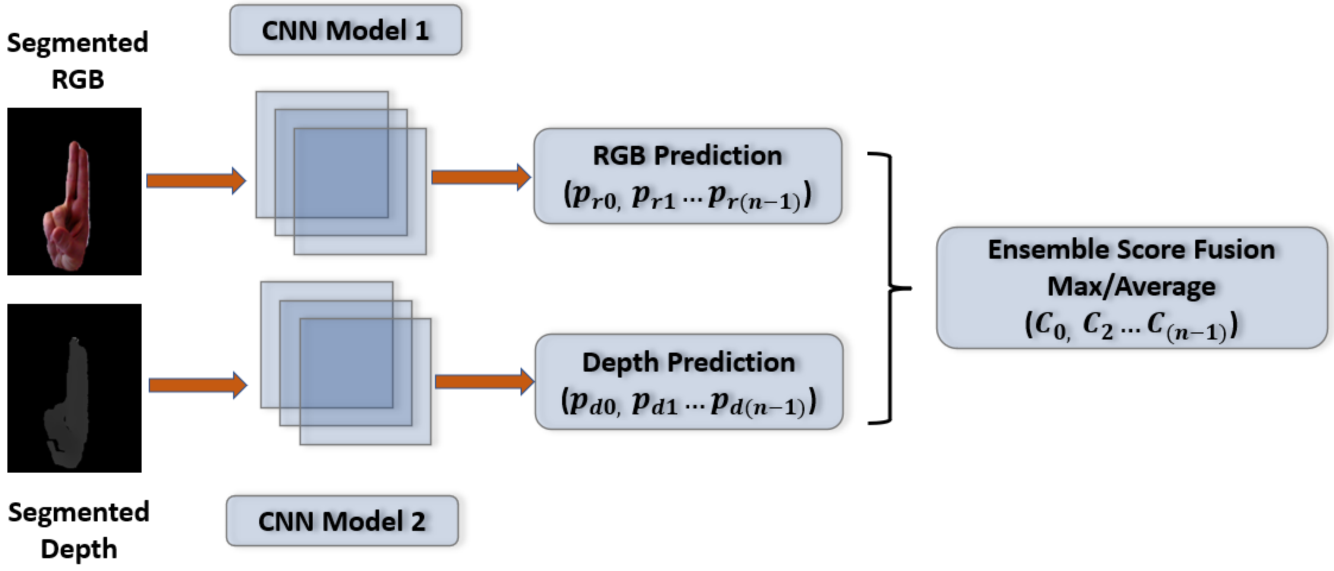


Fig. 4. Block diagram of ensemble score fusion for hand gesture classification.

TABLE I. Classification Model Architecture

Layer Type	Layer Name	Output Shape
Data	RGB input	(320, 320, 3)
CONV2D + RELU	conv2d_relu_1	(320, 320, 16)
CONV2D + RELU	conv2d_relu_2	(320, 320, 16)
BatchNormalization	batch_norm_1	(320, 320, 16)
MaxPooling	max_pooling2d_1	(106, 106, 16)
CONV2D + RELU	conv2d_relu_3	(106, 106, 32)
CONV2D + RELU	conv2d_relu_4	(106, 106, 32)
BatchNormalization	batch_norm_2	(106, 106, 32)
MaxPooling	max_pooling2d_2	(35, 35, 32)
CONV2D + RELU	conv2d_relu_5	(35, 35, 64)
CONV2D + RELU	conv2d_relu_6	(35, 35, 64)
BatchNormalization	batch_norm_3	(35, 35, 64)
MaxPooling	max_pooling2d_3	(11, 11, 64)
CONV2D + RELU	conv2d_relu_7	(11, 11, 128)
CONV2D + RELU	conv2d_relu_8	(11, 11, 128)
BatchNormalization	batch_norm_4	(11, 11, 128)
Global average pooling	gap2d_1	(128)
Dense	dense_1	(64)
RELU	activation_1	(64)
Dropout (0.5)	dropout_1	(64)
Dense	dense_2	(10)
Softmax Activation	activation_2	(10)

vice versa, hence this forms the basis to build a score fusion ensemble model with both RGB and depth stream which gives better accuracy as compared to the uni-modal results.

Let $P_R = (p_{r0}, p_{r1}, p_{r2} \ldots . p_{r(n-1)})$ and $P_D = (p_{d0}, p_{d1}, p_{d2} \ldots . p_{d(n-1)})$ be the probability vectors obtained from RGB stream and depth stream respectively, where $n = 10$ represents the number of class.

$$P_{E_{(max)}} = (max(p_{r0}, p_{d0}), max(p_{r1}, p_{d1}), \\ max(p_{r2}, p_{d2}) \ldots . max(p_{r(n-1)}, p_{d(n-1)})) \tag{2}$$

$$P_{E_{(avg)}} = 0.5 * ((p_{r0} + p_{d0}), (p_{r1} + p_{d1}), \\ (p_{r2} + p_{d2}) \ldots . (p_{r(n-1)} + p_{d(n-1)})) \tag{3}$$

Probability score fusion of max and average methods is mathematically shown in Eq. 2 and Eq. 3. In max fusion, maximum value of RGB and depth probability is considered for each class, where in average fusion mean probability is taken. Further, max of these fused probability is taken to decide the class label of ensemble classification model. From the experiments, it is found the average fusion gives better results as compared to max fusion.

F. Evaluation Metrics

Most commonly used principal measures to evaluate semantic segmentation and classification performance are briefly explained below.

Intersection over Union (IoU) - It is computed as intersection of the pixels from a given class in the predicted results with the ground truth divided by their union. IoU is computed class wise in case of multi-class segmentation. In our work, it is binary case of foreground hand region and the background hence only the class of pixels belonging to foreground is considered.

$$IoU = \frac{T_p}{T_p + F_p + F_n} = \frac{c_{jj}}{c_{ij} + c_{ji} + c_{jj}} i \neq j \tag{4}$$

where, c_{jj} = T_p is the number of pixels which are labeled as class j in ground truth and also predicted as class j, c_{ij} = F_p is the number of pixels which are labeled as class i, but classified as class j that is False Positives class for class j. Similarly, c_{ji} = F_n , the total number of pixels labeled as class j, but classified as class i are the False Negatives (misses) for class j.

Mean Intersection over Union (mIoU): mIoU is the class-averaged IoU across all the images, where k is the number of class.

$$mIoU = \frac{1}{k} \sum_{j=1}^{k} \frac{c_{jj}}{c_{ij} + c_{ji} + c_{jj}} \tag{5}$$

Precision - It is the ratio of hits over summation of hits and false alarms. It indicates total positive cases predicted correctly, over all the predicted positive cases.

$$Precision = \frac{T_p}{T_p + F_p} = \frac{c_{jj}}{c_{ij} + c_{jj}} i \neq j \quad (6)$$

Recall - It is the ratio of hits over summation of hits and misses. It indicates total positive cases predicted correctly, over all the actual positive cases.

$$Recall = \frac{T_p}{T_p + F_n} = \frac{c_{jj}}{c_{ji} + c_{jj}} i \neq j \quad (7)$$

F1score - This measure also known as the dice coefficient, computed as harmonic mean of the precision and recall.

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

In this work precision, recall and F1-score are computed for both segmentation and classification models. In segmentation predicted pixel class is considered whereas in classification predicted image class label. Additionally IoU metrics are used to evaluate segmentation models.

IV. Experimental Results

The appraise the proposed semantic segmentation and ensemble classification framework, experiments are conducted on widely used benchmark OUHANDS [26] dataset for static hand gesture recognition. The experiments are performed using TensorFlow2.0 Keras deep learning library in Google Colab environment with NVIDIA GPU.

OUHANDS dataset [26] - contains RGB, raw depth data and segmentation ground truth images. It consists of 10 unique gestures captured in complex backgrounds and lighting changes from 23 subjects with different hand gesture sizes and shapes. Training dataset consists of 2000 images split into 1600 for training, 400 for validation. Test dataset contains 1000 images of the unseen individuals in the training set. All images in the training and test datasets are resized to 320 x 320 image resolution and used in segmentation and classification.

As depicted in Fig. 2, We have chosen three well known segmentation networks namely: UNet, ResUNet and DeepLabV3+. RGB and Depth images are used as input into these networks, which outcomes the hand region segmented mask. We trained these networks with various type of input data like RGB, Depth and the stacked RGBD data. Out of these 3 combination Depth data based model gave better accuracy with fine segmented hand region mask.

As shown in Table II, it can be observed that DeepLabV3+ segmentation results are better than UNet and ResUNet models. Also it can be observed that, the Depth data based based models provide better F1-score as compared to RGB data and stacked RGBD data. The DeepLabV3+ depth segmentation model resulted in the highest F1 score of 0.9235 compared to other networks.

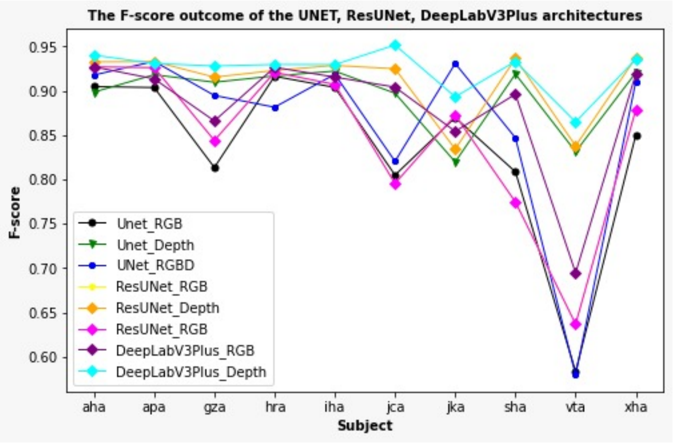


Fig. 5. Analysis of segmentation F1-score outcome by various networks specific to each subject.

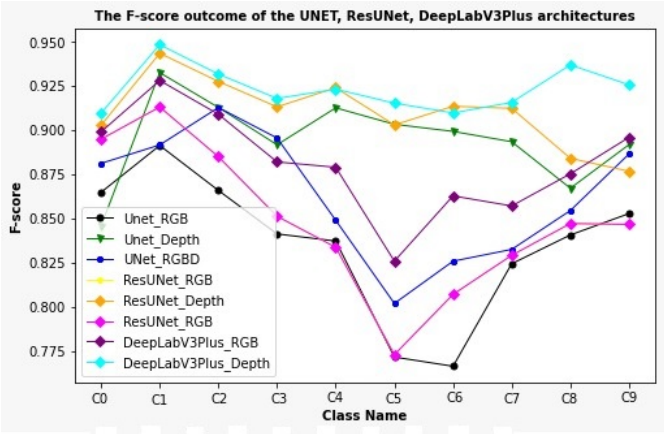


Fig. 6. Analysis of segmentation F1-score outcome by various networks specific to each class.

We have analyzed the segmentation accuracy for each subject performing various gestures under different lighting conditions as shown in Fig. 5. As it can be observed, Deeplabv3+ depth segmentation model gave better accuracy for all the subjects.

Segmentation results were analyzed over each gesture class as shown in Fig. 6. From the plot it is evident that Deeplabv3+ model gives better segmentation accuracy over UNet and ResUNet models. Hence we can conclude that DeepLabV3+ segmentation model trained with Depth data is suitable for efficiently segmenting the hand region in complex scenarios.

Based on experimental results DeepLabV3+ is selected as best segmentation model, predicted binary segmentation mask is used for masking to discard the background by bitwise AND operation on the input RGB and depth image. This results in fine segmented foreground hand region in RGB and Depth image constituting to segmented RGB and Segmented Depth images.

Table III shows the classification results on OUHANDS test dataset from the models trained on segmented RGB and Segmented Depth training data using custom CNN network

TABLE II. Comparison of Hand Region Segmentation Accuracy on OUHANDS Test Dataset

No	Method	Data Type	Mean IOU	Average Precision	Average Recall	Average F1 score	Num. of Parameters	Model Size
1	UNet	RGB	0.7404	0.8655	0.8303	0.8358	7.861 M	90.1 MB
2		Depth	0.8207	0.8898	0.9161	0.8952	7.861 M	90.1 MB
3		RGBD	0.7878	0.8905	0.8747	0.8633	7.861 M	90.1 MB
4	ResUNet	RGB	0.7577	0.8725	0.8503	0.8482	4.680 M	53.9 MB
5		Depth	0.8477	0.9223	0.9136	0.9102	4.680 M	53.9 MB
6		RGBD	0.7803	0.9250	0.8298	0.8542	4.680 M	53.9 MB
7	DeepLabV3 Plus	RGB	0.8020	0.8717	0.9055	0.8815	17.830 M	204.7 MB
8		Depth	0.8638	0.9022	0.9534	0.9235	17.830 M	204.7 MB

TABLE III. Comparison of Precision, Recall and F1-Score of OUHANDS Test Dataset Classification using Segmented RGB, Depth, Max and Average Ensemble Data

Class	Segmented RGB			Segmented Depth			Max Ensemble			Average Ensemble		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
C0	0.9091	0.9000	0.9045	0.8704	0.9400	0.9038	0.8455	0.9490	0.8942	0.8440	0.9583	0.8976
C1	0.8879	0.9694	0.9268	0.9700	0.9898	0.9798	0.9897	0.9796	0.9846	0.9896	0.9794	0.9845
C2	0.8588	0.7374	0.7935	0.9146	0.7576	0.8287	0.8941	0.7677	0.8261	0.8929	0.7732	0.8287
C3	0.9545	0.8400	0.8936	0.8557	0.8300	0.8426	0.8864	0.7879	0.8342	0.8876	0.7980	0.8404
C4	0.9029	0.9300	0.9163	0.8911	0.9000	0.8955	0.8205	0.9600	0.8848	0.8288	0.9684	0.8932
C5	0.9524	0.8247	0.8840	0.9659	0.8763	0.9189	0.9778	0.8889	0.9312	0.9773	0.8866	0.9297
C6	0.8776	0.8687	0.8731	0.8800	0.8889	0.8844	0.8980	0.8800	0.8889	0.9053	0.8776	0.8912
C7	0.8990	0.8900	0.8945	0.8500	0.8500	0.8500	0.8947	0.8500	0.8718	0.9043	0.8763	0.8901
C8	0.7870	0.8500	0.8173	0.8654	0.9000	0.8824	0.8505	0.9100	0.8792	0.8660	0.8936	0.8796
C9	0.7521	0.9192	0.8273	0.7768	0.8788	0.8246	0.8190	0.8687	0.8431	0.8286	0.8878	0.8571
Average	**0.8780**	**0.8730**	**0.8731**	**0.8836**	**0.8810**	**0.8809**	**0.8875**	**0.8841**	**0.8837**	**0.8926**	**0.8895**	**0.8891**

TABLE IV. Comparison of Hand Gesture Recognition Accuracy on OUHANDS Test Dataset

No	Method	Input data	F1-score	Input Size	# Parameters	Model Size
1	ResNet-50 [27]	RGB	0.8138	224×224	23.60 M	99 MB
2	DenseNet-121 [28]	RGB	0.8281	224×224	7.04 M	33 MB
3	Two stream CNN [29]	RGB & sMask	0.8621	256×256	-	-
4	MobileNet [30]	RGB	0.8650	224×224	3.22 M	16 MB
5	RGB-D Early fusion [15]	sRGB & sDepth	0.8757	320×320	0.3035 M	3.6 MB
6	HGR-Net [31]	RGB & sMask	0.8810	320×320	0.499 M	2.4 MB
7	Proposed SSEC	sRGB & sDepth	**0.8891**	320×320	sRGB = 0.3034 M sDepth = 0.3034 M	3.6 MB 3.6 MB

sRGB = Segmented RGB
sDepth = Segmented Depth
sMask = Segmented Binary Mask

shown in Table I. Accuracy of these two models are analyzed and found that few miss-classified images are detected complementary, hence score fusion is performed using max and average operations to get ensembled gesture classification results. It can be observed that average ensemble gives the best result of 88.91%. It is also evident from Fig. 7, average ensemble provides best accuracy over all models for all the gesture class.

Proposed framework of semantic segmentation and ensemble classification (SSEC), is compared with the state of the art methods as shown in Table IV. Deeplabv3+ depth segmentation model followed by classification using average score fusion gives the best F1 score accuracy on OUHANDS test dataset.

V. Discussion

Comprehensive experiments using RGB and Depth data stream are conducted in both segmentation and classification stage. As in Fig 2, hand region segmentation is performed using three segmentation networks UNet, ResUNet and Deeplab V3+ with RGB, Depth and stacked RGBD data. Corresponding experimental result is shown in Table II which indicates Deeplab V3+ with Depth data gives the better segmentation accuracy. This is also evident from the plots in Fig. 5 and Fig. 6, where segmentation accuracy is analyzed for each subject and each class respectively (both in aqua color plot).

Segmented RGB and Depth data is used to train the classification model as depicted in Fig. 4. Corresponding experimental results in Table III and plot in Fig. 7. shows that average ensemble (blue color plot) gives the better classification accuracy.

Proposed SSEC framework of Deeplab V3+ based semantic segmentation and average score ensemble classification is evaluated on OUHANDS benchmark dataset and compared the accuracy with existing methods as in Table IV. Experimental results shows that proposed methods gives the highest F1-score of 0.8891 and proved to be better than state-of-the-art methods.

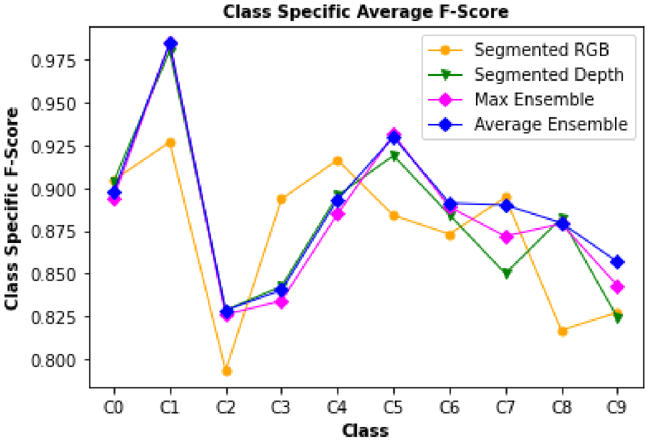


Fig. 7. Analysis of classification F1-score results of each class using segmented RGB, depth data models and various score fusion techniques.

VI. Conclusion

In this paper, we introduce SSEC, a novel semantic segmentation and ensemble classification framework on RGB-D Data for Static Hand Gesture Recognition. Specifically, we have analyzed three segmentation networks UNet, ResUNet and Deeplab V3+ with RGB, Depth and stacked RGBD data. Deeplab V3+ with Depth data gave higher F1 Score, the prediction outcome of this model is used as segmentation mask to discard background in RGB and depth data. In classification stage, custom CNN network was trained with segmented depth and segmented RGB data individually. Experimental results shows that, average score ensemble of these models can give better accuracy as compared to individual models. Hence it is inferred that RGB-D is with score fusion model ensembling is suitable for hand gesture classification as compare to the stat-of-the art methods.

Limitation of current approach is the usage of same network in both the RGB-D stream which may not give the diverse features, this can be further improved by using different CNN architectures in both streams to extract complimentary features.

In future work, we intend to develop a framework for dynamic hand gesture recognition considering the temporal sequence of RGB-D data for word and sentence level classification.

References

[1] F. Al Farid, N. Hashim, J. Abdullah, M. R. Bhuiyan, W. N. Shahida Mohd Isa, J. Uddin, M. A. Haque, and M. N. Husen, "A structured and methodological review on vision-based hand gesture recognition system," *Journal of Imaging*, vol. 8, no. 6, p. 153, 2022.

[2] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," *journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[3] S. Yuying, C. Sujie, L. Ming, L. Siying, P. Yisen, and G. Xiaojun, "Flexible strain sensors for wearable hand gesture recognition: From devices to systems," *Computers and Electrical Engineering*, vol. 1002, no. 170, pp. 1–17, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/epdf/10.1002/aisy.202100046

[4] S. Jaya Prakash, S. Suraj Prakash, A. Samit, and P. Sarat Kumar, "Rbi-2rcnn: Residual block intensity feature using a two-stage residual convolutional neural network for static hand gesture recognition," *Signal, Image and Video Processing*, vol. 1007, no. 170, pp. 1–17, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11760-022-02163-w

[5] B. Gopa, V. Monu, C. Mahesh, and V. Santosh Kumar, "Hyfinet: Hybrid feature attention network for hand gesture recognition," *Multimedia Tools and Applications*, vol. 1007, no. 170, pp. 1–17, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11042-021-11623-3

[6] H. Xu, G. Chen, Z. Wang, L. Sun, and F. Su, "Rgb-d-based pose estimation of workpieces with semantic segmentation and point cloud registration," *Sensors*, vol. 19, no. 8, p. 1873, 2019.

[7] M. Poggi, G. Agresti, F. Tosi, P. Zanuttigh, and S. Mattoccia, "Confidence estimation for tof and stereo sensors and its application to depth data fusion," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1411–1421, 2019.

[8] N. Dayananda Kumar, K. Suresh, and R. Dinesh, "Depth based static hand gesture segmentation and recognition," in *Cognition and Recognition: 8th International Conference, ICCR 2021*. Springer, 2021, pp. 125–138.

[9] C. N. Aithal, P. Ishwarya, S. S, Y. C. N, D. Kumar, and K. V. Suresh, "Dynamic hand segmentation," in *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2021, pp. 1–6.

[10] R. F. Pinto, C. D. Borges, A. M. Almeida, and I. C. Paula, "Static hand gesture recognition based on convolutional neural networks," *Journal of Electrical and Computer Engineering*, vol. 2019, pp. 1–12, 2019.

[11] W. Aly, S. Aly, and S. Almotairi, "User-independent american sign language alphabet recognition based on depth image and pcanet features," *IEEE Access*, vol. 7, pp. 123 138–123 150, 2019.

[12] G. Benitez-Garcia, L. Prudente-Tixteco, L. C. Castro-Madrid, R. Toscano-Medina, J. Olivares-Mercado, G. Sanchez-Perez, and L. J. G. Villalba, "Improving real-time hand gesture recognition with semantic segmentation," *Sensors*, vol. 21, no. 2, p. 356, 2021.

[13] D. Chen, G. Li, Y. Sun, J. Kong, G. Jiang, H. Tang, Z. Ju, H. Yu, and H. Liu, "An interactive image segmentation method in hand gesture recognition," *Sensors*, vol. 17, no. 2, p. 253, 2017.

[14] B. Xie, X. He, and Y. Li, "Rgb-d static gesture recognition based on convolutional neural network," *The Journal of Engineering*, vol. 2018, no. 16, pp. 1515–1520, 2018.

[15] N. D. Kumar, K. Suresh, and R. Dinesh, "Cnn based static hand gesture recognition using rgb-d data," in *2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP)*. IEEE, 2022, pp. 1–6.

[16] J. P. Sahoo, A. J. Prakash, P. Pławiak, and S. Samantray, "Real-time hand gesture recognition using fine-tuned convolutional neural network," *Sensors*, vol. 22, no. 3, p. 706, 2022.

[17] S. Wang, S. Zhang, X. Zhang, and Q. Geng, "A two-branch hand gesture recognition approach combining atrous convolution and attention mechanism," *The Visual Computer*, pp. 1–14, 2022.

[18] A. S. M. Miah, M. A. M. Hasan, J. Shin, Y. Okuyama, and Y. Tomioka, "Multistage spatial attention-based neural network for hand gesture recognition," *Computers*, vol. 12, no. 1, p. 13, 2023.

[19] J. P. Sahoo, S. P. Sahoo, S. Ari, and S. K. Patra, "Derefnet: Dual-stream dense residual fusion network for static hand gesture recognition," *Displays*, p. 102388, 2023.

[20] J. Ding and N.-W. Zheng, "Rgb-d depth-sensor-based hand gesture recognition using deep learning of depth images with shadow effect removal for smart gesture communication," *Sensors and Materials*, vol. 34, no. 1, pp. 203–216, 2022.

[21] L. Roda-Sanchez, C. Garrido-Hidalgo, A. S. García, T. Olivares, and A. Fernández-Caballero, "Comparison of rgb-d and imu-based gesture recognition for human-robot interaction in remanufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 124, no. 9, pp. 3099–3111, 2023.

[22] P. Sun, W. Zhang, H. Wang, S. Li, and X. Li, "Deep rgb-d saliency detection with depth-sensitive attention and automatic multi-modal

fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1407–1417.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[25] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[26] M. Matilainen, P. Sangi, J. Holappa, and O. Silvén, "Ouhands database for hand detection and pose recognition," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2016, pp. 1–5.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[28] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[29] G. Jianchun, G. Jiannuan, and W. Lili, "Gesture recognition method based on attention mechanism for complex background," *Journal of Physics: Conference Series*, vol. 1873, no. 1, p. 012009, apr 2021.

[30] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.

[31] D. Amirhossein, T. A. Tavakoli, M. Tahmasbi, and M. Mirmehdi, "Hgr-net: a fusion network for hand gesture segmentation and recognition," *IET Computer Vision*, vol. 13, no. 700-707, 2019.

COVID-19 Dataset Clustering based on K-Means and EM Algorithms

Youssef Boutazart[1], Hassan Satori[2], Anselme R. Affane M.[3], Mohamed Hamidi[4], Khaled Satori[5]
Computer Science Laboratory, Signals Automation and Cognition (LISAC)
Department of Mathematics and Computer Science, Faculty of Sciences, Dhar Mahraz,
Sidi Mohamed Ben Abdallah University Fez, Morocco[1,2,3,5]
Team of Modeling and Scientific Computing (LaMAO), FPN, UMP[4]

***Abstract*—In this paper, a COVID-19 dataset is analyzed using a combination of K-Means and Expectation-Maximization (EM) algorithms to cluster the data. The purpose of this method is to gain insight into and interpret the various components of the data. The study focuses on tracking the evolution of confirmed, death, and recovered cases from March to October 2020, using a two-dimensional dataset approach. K-Means is used to group the data into three categories: "Confirmed-Recovered", "Confirmed-Death", and "Recovered-Death", and each category is modeled using a bivariate Gaussian density. The optimal value for k, which represents the number of groups, is determined using the Elbow method. The results indicate that the clusters generated by K-Means provide limited information, whereas the EM algorithm reveals the correlation between "Confirmed-Recovered", "Confirmed-Death", and "Recovered-Death". The advantages of using the EM algorithm include stability in computation and improved clustering through the Gaussian Mixture Model (GMM).**

Keywords*—*COVID-19; clustering; k-means; EM algorithm; GMM

I. Introduction

Cluster analysis involves organizing data into meaningful and valid groups [1], which are homogeneous and similar. This technique involves classifying each data point into a specific set using clustering algorithms [2], [3]. A method proposed by the authors in [4] determines the optimal number of clusters, k, which represents the inherent significant clustering structures of the dataset. K-Means and Expectation Maximization (EM) algorithms are commonly used for clustering [5]. The proposed EM algorithm, initially designed for finding maximum likelihood parameters of a statistical model, has been applied to various domains such as speech recognition [6], interactive systems [7], etc.

On the other hand, the researchers in [8] have proposed a new epidemiological mathematical model for the spread of the COVID-19 disease with a special focus on the transmissibility of individuals with severe symptoms. Recently an important report using C++ can be used to "track" the daily evolution of new confirmed cases of the COVID-19 epidemic [9]. Rizvi et al. [10] have described K-Means clustering of 79 countries has been performed for COVID-19 confirmed cases and COVID-19 death cases based on 18 feature variables.

This study presents a fresh approach to analyzing the COVID-19 dataset using clustering techniques. Specifically, we apply a standard version of K-Means and EM algorithms based on GMM to partition the local COVID-19 Moroccan dataset into three sets: "Confirmed-Recovered", "Confirmed-Death", and "Recovered-Death", with varying cluster numbers. Our primary objective is to identify the optimal classification for each data cluster.

This paper is organized as follows: Section II gives the Literature Review. The K-Means and EM algorithms is introduced in Section III. The COVID-19 pandemic is presented in Section IV. Section V describes the COVID-19 dataset. Section VI exposes the results and discussion. Finally, in Section VII we conclude this work and gives perspectives.

II. Literature Review

The COVID-19 pandemic has led to an increase in the use of data mining and machine learning techniques to understand and analyze the spread of the virus. Clustering is a popular technique used to group similar data points together. K-Means, EM Algorithm and GMM are three commonly used clustering algorithms in machine learning. Several clustering methods have been developed with the objective to find the correct number of clusters [11], [12], [13], [14], [15]. In [16] the authors focus on utilizing Probabilistic Graphical Models for detecting COVID-19, resulting in excellent detection of the disease. One potential use of the EM algorithm is to estimate the parameters of a mixture model in cases where the data is incomplete. This technique is sometimes referred to as finding the parameters of Gaussian mixture densities [17], [18]. Eva and Dharmende [19] conducted a comparison between K-Means and GMM to assess their effectiveness in representing clusters of heterogeneous resource usage in Cloud workloads. Their experiments, which utilized Google cluster trace and business critical workloads by Bitbrains, revealed that K-Means provided a more generalized representation, whereas GMM resulted in better clustering with clearly defined usage boundaries. Despite Gaussian Mixture Model's longer computation time compared to K-Means, it is preferred for more detailed workload analysis and characterization.

Appiah et al. [20] proposed a study that utilizes the EM algorithm, which is initialized by a semi-supervised K-Means clustering approach based on geodesic distance classification of crime dataset. The aim is to track changes in cluster centroids (mean), shape and orientation, volume, and predictive trends of criminal activities. In this approach, the cluster assignment obtained from K-means is assumed as the distribution of GMM. The model-based clustering algorithm is then used to estimate the parameters of the mixed model while maintaining the probabilistic assignment and multivariate nature of the

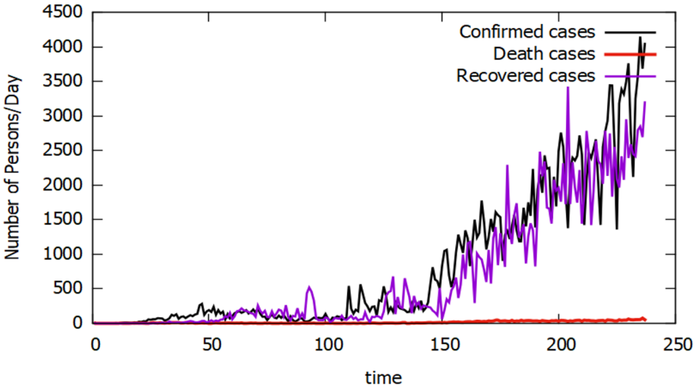


Fig. 1. COVID-19 confirmed-recovered and death cases in Morocco.

model. On the other hand, Zarikas et al. [21] developed a clustering algorithm designed specifically for grouping countries based on COVID-19 active cases, active cases per population, and per area. The results showed that countries facing similar impacts of COVID-19 also shared similar societal, economic, and other factors.

Aungkulanon et al. [22] clustered different regions of Thailand based on financial conditions and mortality differentials, revealing super-locale that are mainly urban and have a low all-cause normalized mortality proportion but a high colorectal disease-specific death rate. The study also found that deaths caused by liver cancer, diabetes, and renal diseases are common in low economic super-regions. Malav et al. [23] conducted a study to predict coronary heart disease using K-means and artificial neural networks. The combined approach led to a system with a very high accuracy rate. Another work by Singh et al. [24] used clustering and classification techniques to forecast heart diseases with high accuracy.

Isikhan et al. [25] clustered countries based on causes of deaths, health profiles, and risk factors using unsupervised K-means. The study analyzed clusters based on some financial and socio-demographic indicators and found that climate and ethnicity were more significant factors for clustering than socio-economic factors. These studies demonstrate the importance of COVID-19 dataset clustering in identifying patterns and trends associated with the virus, which can aid in developing effective strategies to combat its spread.

TABLE I. CONFIRMED-RECOVERED-DEATH COVID-19 DATASET

Day	1	2	20	30	40	50	60	90	...	237
Confirmed	1	0	22	63	74	191	102	69	...	4045
Recovered	0	0	0	10	13	23	56	141	...	3197
Death	0	0	0	3	10	2	2	2	...	50

III. K-MEANS AND EM ALGORITHMS

Given a set of observations $Y = (Y_1, ..., Y_N)$, independent and identically distributed (i.i.d) where each observation $Y_t = (y_{t1}, ..., y_{tj}, ..., y_{td}) \in R^d$ is a d-dimensional real vector. The objectives of K-Means and EM are to partition N observations into G clusters [26].

A. K-Means Algorithm

In this part, the objective is to find values for z_{tk} and μ_k the mean so as to minimize D. Let $\Phi = \mu = \{\mu_1, ..., \mu_G\}$ be the set represents the mean of each cluster c_k, where $C_k \in \{C_1, ..., C_G\}$ the set of G clusters, and let $Z = (z_1, z_2, ..., z_N)$ the set of binary indicator variables.

$$D(\Phi, Z) = \sum_{t=1}^{N} \sum_{k=1}^{G} z_{tk} \|Y_t - \mu_k\|^2 \qquad (1)$$

Where $z_{tk} = 1$ when Y_t is a member of C_k , otherwise $z_{tk} = 0$. Or more exactly $\arg\min_k D(\Phi, Z)$. when D achieved minimal value, sum of $\|Y_t - \mu_k\|^2$ is minimal [27].

$$d(Y_t, \mu_k) = \sqrt{\sum_{j=1}^{d} (y_{tj} - \mu_{kj})^2} \qquad (2)$$

by Euclidean distance. We can do this through an iterative procedure in which each iteration involves two successive steps corresponding to successive optimizations with respect to z_{tk} and μ_k. We initialize the class centers $\left\{\mu_1^{(0)}, ..., \mu_G^{(0)}\right\}$ for the $\{C_1, ..., C_G\}$ set of clusters; by some initial values called seed-points, using methodically sampling.

Step 1:

We minimize D and we update z_{tk}, keeping the μ_k fixed.

Step 2:

We minimize D and we update μ_k, keeping the z_{tk} fixed.

$$\mu_k^{(m+1)} = \frac{\sum_{t=1}^{N} z_{tk}^{(m)}.Y_t}{\sum_{t=1}^{N} z_{tk}^{(m)}} \qquad (3)$$

(m) being the current iteration. This two-stage optimization is then repeated until convergence.

B. Expectation Maximization Algorithm

In this work, EM algorithm is used to complete the missing COVID-19 data. We introduce the latent variable Z. Y_t can describe the mix “Confirmed cases-Recovered cases”. The same study for the mixture of confirmed cases - death cases and recovered cases - death cases. We will assume that the observations Y_t are i.i.d and the observations from different clusters have correlated Bivariate Gaussian Density. If data t belongs to cluster C_k (denoted by $t \in C_k$) then:

$$Y_t \setminus t \in C_k \sim f(y_t/\mu_k, \Sigma_k) \qquad (4)$$

$$f(y_t/\mu_k, \Sigma_k) = \frac{1}{2\Pi^{\frac{d}{2}} \sqrt{|\Sigma_k|}}$$
$$\exp \frac{-1}{2}[(y_t - \mu_k)^t \Sigma_k^{-1} (y_t - \mu_k)] \qquad (5)$$

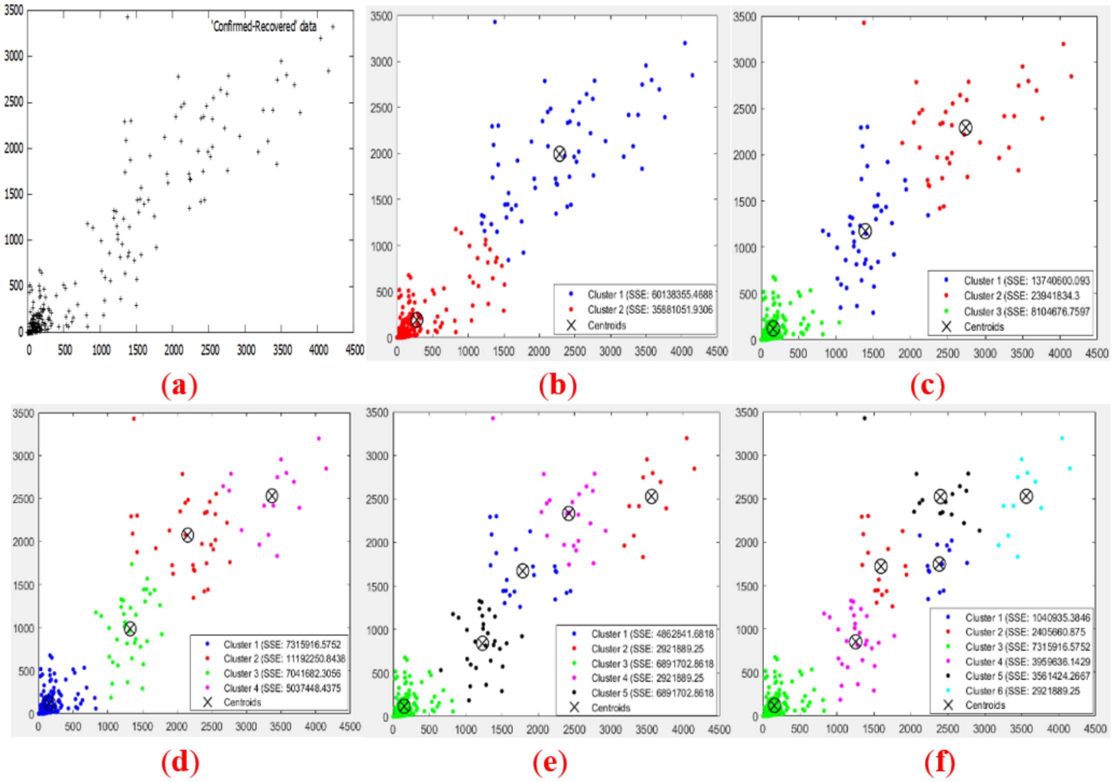


Fig. 2. (a) Two-dimensional input “Confirmed-Recovered” data; with no clustering. (b), (c), (d), (e) and (f) K-means partitions, respectively with k = 2, k = 3, k = 4, k = 5 and k = 6. The centroids are marked with a cross.

μ_k and Σ_k denote the mean vector and covariance matrix. Assign a data point to a nearest cluster, with calculate the following likelihood [28].:

$$\begin{aligned}\gamma(z_{tk}) &= \mathbf{E}(z_{tk}) = \mathbf{P}(z_{tk} = 1/y)\\ &= \frac{\mathbf{P}(z_{tk} = 1)\mathbf{P}(y_t/z)}{\mathbf{P}(y)}\end{aligned} \tag{6}$$

Where $\mathbf{P}(z_{tk} = 1/y)$ is a posterior probability of $y_t \in C_k$ the k^{th} – classes and z_t correspond to the Gaussian identity which generated an entry y_t.

Step 1 (Expectation): Given the current estimates, $[\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}, \boldsymbol{\Pi_k}]$

$$\gamma(\boldsymbol{z_{tk}}) = \frac{\boldsymbol{\Pi_k f_k(y_t/\mu_k, \Sigma_k)}}{\sum_{j=1}^{G} \boldsymbol{\Pi_j f_j(y_t/\mu_j, \Sigma_j)}} \tag{7}$$

Step 2 (Maximization): Compute the parameters that maximize the likelihood of the data set $\mathbf{P}(\boldsymbol{Y/\mu_k, \Sigma_k, \Pi_k, z_{tk}})$ which is the probability of all of the data under the GMM. Find the probability $\mathbf{P}(\boldsymbol{Y})$ that generated the COVID-19 dataset. Maximizing this with respect to each of the parameters can be done in closed form:

$$\boldsymbol{\Pi_k^{new\ ite}} = \frac{\sum_{t=1}^{N} \gamma(\boldsymbol{z_{tk}})}{\boldsymbol{N}} \tag{8}$$

$$\boldsymbol{\mu_k^{new\ ite}} = \frac{\sum_{t=1}^{N} \gamma(\boldsymbol{z_{tk}})\boldsymbol{y_t}}{\sum_{t=1}^{N} \gamma(\boldsymbol{z_{tk}})} \tag{9}$$

$$\boldsymbol{\Sigma_k^{new\ ite}} = \frac{\sum_{t=1}^{N} \gamma(\boldsymbol{z_{tk}})((\boldsymbol{y_t} - \boldsymbol{\mu_k^{new\ ite}}) \otimes (\boldsymbol{y_t} - \boldsymbol{\mu_k^{new\ \ ite}})^{\boldsymbol{t}}}{\sum_{t=1}^{N} \gamma(\boldsymbol{z_{tk}})} \tag{10}$$

1) Re-estimation of mixed weights: To find the parameter we using a Lagrange multipliers [29] with constraint $\sum_{i=1}^{G} \boldsymbol{\Pi_i} = \boldsymbol{1}$ and maximizing the following quantity:

$$\boldsymbol{L(l(\Phi), \lambda) = l(\Phi) + \lambda(\sum_{k=1}^{G} \Pi_k - 1)} \tag{11}$$

$$\boldsymbol{Where \frac{\partial L(l(\Phi), \lambda)}{\partial \Pi_k} = 0}$$

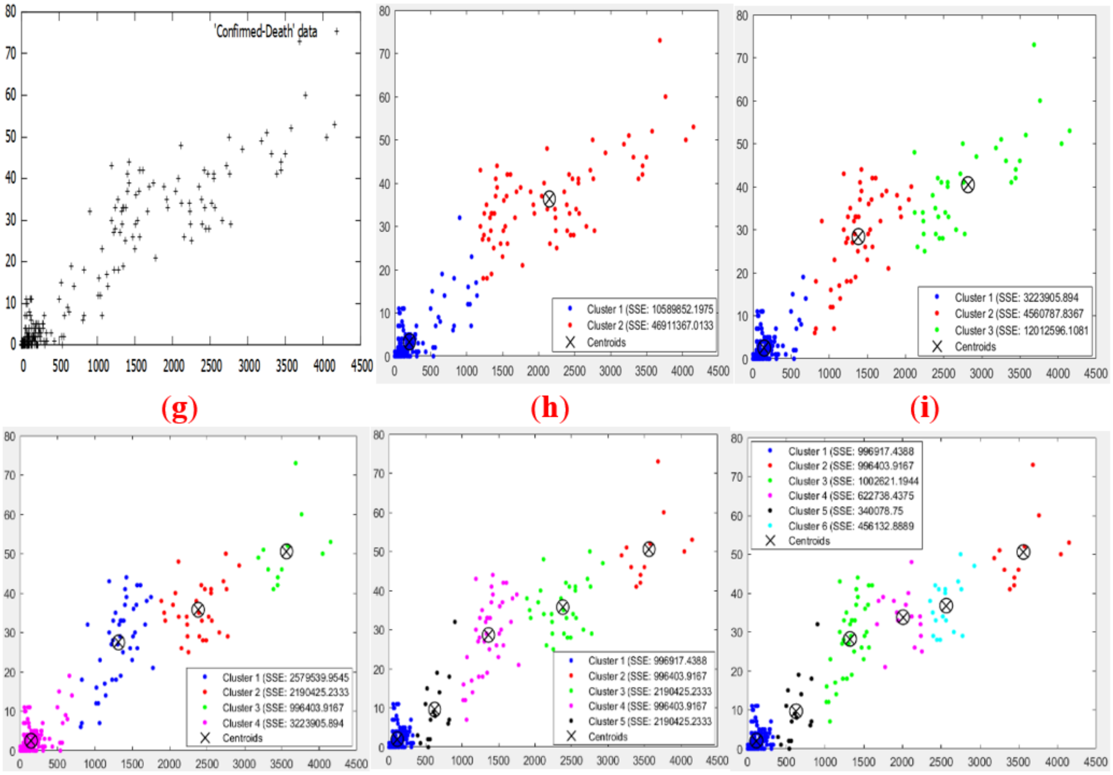
Then we obtain

$$\boldsymbol{\sum_{t=1}^{N} \frac{\Pi_k f(y_t/\mu_k, \Sigma_k)]}{\sum_j [\Pi_j f(y_t/\mu_j, \Sigma_j)]} + \lambda \Pi_k = 0}$$

and we have new estimation for $\boldsymbol{\Pi_k}$ (see Eq. 8).

2) Re-estimation of the means vectors: We assume $\gamma(\boldsymbol{z_{tk}})$ fixed. We derive this equation with respect to the means $\boldsymbol{\mu_k}$ at zero, we obtain:

$$\begin{aligned}\boldsymbol{l(\Phi)} &= \boldsymbol{\sum_{t=1}^{N} \ln[\sum_{k=1}^{G} \frac{\Pi_k}{2\Pi\sqrt{|\Sigma_k|}}}\\ &\boldsymbol{\exp \frac{-1}{2}[(y_t - \mu_k)^t \Sigma_k^{-1}(y_t - \mu_k)]]}\end{aligned} \tag{12}$$



(g) (h) (i)

(j) (l) (m)

Fig. 3. (g) Two-dimensional input “Confirmed-Death” data; with no clustering. (h), (i), (j), (l) and (m) K-means partitions, respectively with k = 2, k = 3, k = 4, k = 5 and k = 6. The centroids are marked with a cross.

$\boldsymbol{Where}\ \frac{\partial l(\Phi)}{\partial \mu_k} = \mathbf{0}$, Then we find:

$$\sum_{t=1}^{N} \frac{\Pi_k f(y_t/\mu_k, \Sigma_k)]}{\sum_j [\Pi_j f(y_t/\mu_j, \Sigma_j)]} \Sigma_k^{-1}(y_t - \mu_k) = 0$$

The new μ_k is gives in (Eq. 9)

3) Re-estimation of the covariance matrix: In the same way we derive $l(\Phi)$ with respect to Σ_k Where $\frac{\partial l(\Phi)}{\partial \Sigma_k} = \mathbf{0}$, then we obtain new values of covariance matrix (see Eq. 10).

IV. COVID-19 Pandemic

Later in 2019, in the city of Wuhan, in China, a new discovered version of coronavirus was detected as the principal reason for a strange aspect of pneumonia cluster. Local scientists react by isolating the SARS-CoV-2 into a patient on the earlier of January 2020, which led to the genome sequence of the SARS-CoV-2 [30].

According to the authors of sequencing, phylogenetic analysis this genome has made it possible to establish that the initial host of this virus is an animal sold on the market in Wuhan. Several studies have suggested bats could be at the origin of SARS-CoV-2 [31]. The virus was referred to as 2019-nCoV before the COVID-19 name. It is defined as a severe acute respiratory syndrome coronavirus number 2 (SARS-CoV-2). The WHO declares that the first the infection as a pandemic on March 11, 2020. It rapidly spread, followed by an increase in the number of infected cases around the globe. To this disease of August 16, 2020, the world has had 21.294.845 total confirmed cases, and 761.779 total deaths cases [32].

V. COVID- 19 Dataset Description

In the present study, we use public data from the COVID-19 outbreak in Morocco to estimate the evolution of this epidemic. The data is received through the official website created by the Moroccan Ministry of Health. For this disease, Morocco has had 194461 total confirmed cases, said the Director of epidemiology and disease control at the Ministry of Health as of October 24,2020 the total number of deaths has increased to 3255; and 160372 total cured cases (see Fig. 1) [33].

The training dataset is composed of the real COVID-19 cases daily collected Confirmed, Recovered, and Death patterns. The clustering is done with two-dimensional dataset “Confirmed – Recovered”, “Confirmed – Death” and “Recovered – Death” features of 237 samples. The Table I below shows a part of the complete data.

The recording of the $\mathbf{237^{th}}$ COVID-19 cases are store in the Table I. Each feature is a combination of two parameters, the Confirmed recorder and the Death cases, then the Confirmed recorder and the Recovered cases and the Recovered recorder and the Death cases, respectively.

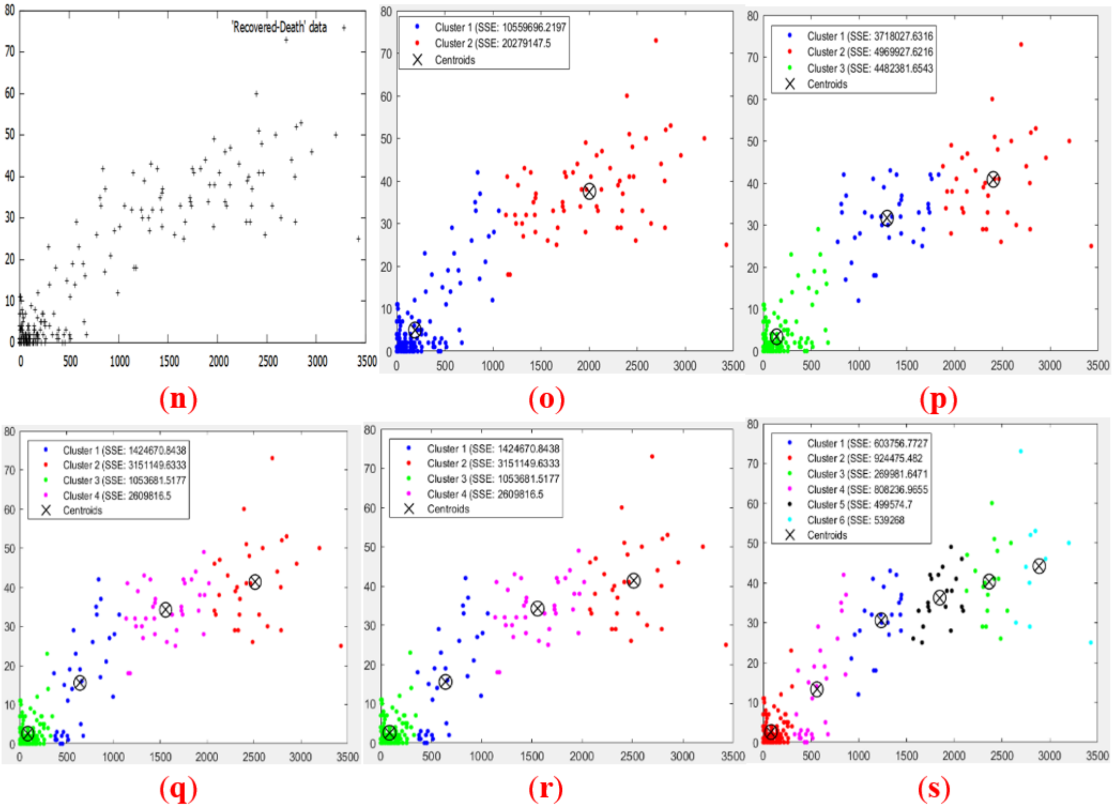


Fig. 4. (n) Two-dimensional input "Recovered - Death" data; with no clustering. Data to illustrate the K-means procedure. (o), (p), (q), (r) and (s) K-Means partitions, respectively with k = 2, k = 3, k = 4, k = 5 and k = 6. The red dots represent the centroid of each cluster.

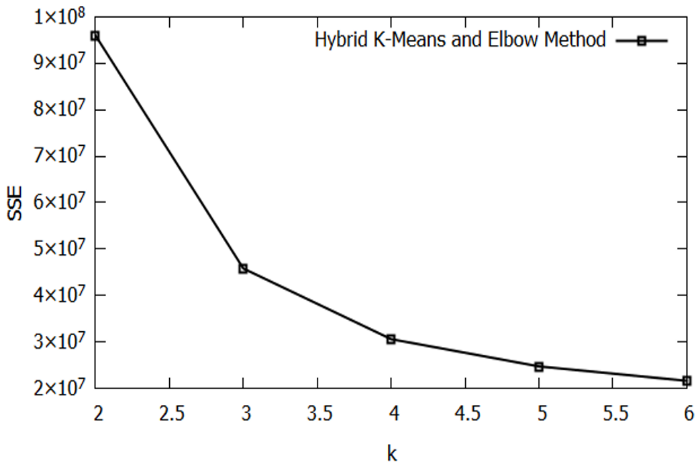


Fig. 5. Graph of sum square error depending on the number k for 'Confirmed-Recovered' data.

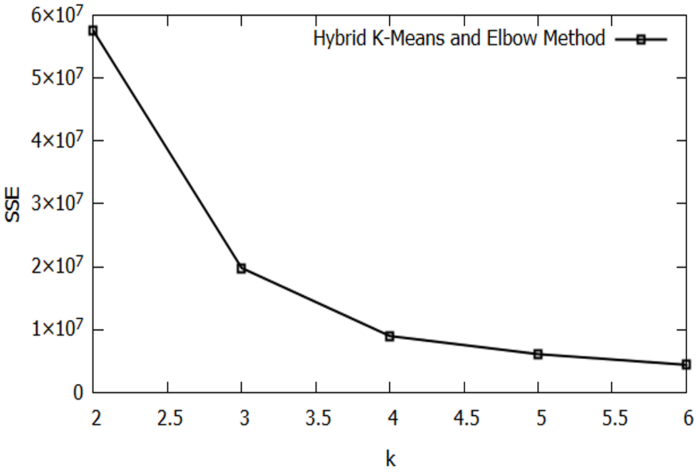


Fig. 6. Graph of sum square error depending on the number k for 'Confirmed-Death' data.

VI. Results and Discussion

In this work, we selected k-points as the primary group ranks as the points are calculated in order. The total number of initial points c_k is $237/k$ for all groups, then we define the initial centroids μ_k. The test data consists of three groups, "Confirmed – Recovered" (see Fig. 2), "Confirmed – Death" (see Fig. 3) and "Recovered– Death" (see Fig. 4). After divided each group into $k = 2$ to $k = 6$. The hybrid of K-Means algorithm and the Elbow method is been used to determine the best clustering as in [34].

Each data point is classified by computing the distance between that point and each group center, and then classifying the point to be in group whose center is closest to it. The results of sum square error calculations of each cluster have experienced the greatest decrease in k = 4 for groups "Confirmed-Recovered" and "Confirmed-Death", k=3

TABLE II. The Initial Values

‘Confirmed-Recovered’ data				
	K-Means	EM GMM		
C_1	$\mu_1 = [73.24, 15.73]^t$	$\mu_1 = [73.24, 15.73]^t$	$\Sigma_1 =$	$\begin{matrix} 4974.83 & 763.34 \\ 763.34 & 729.72 \end{matrix}$
C_2	$\mu_2 = [127.88, 132.76]^t$	$\mu_2 = [127.88, 132.69]^t$	$\Sigma_2 =$	$\begin{matrix} 11465.05 & -1746.47 \\ -1746.47 & 10188.45 \end{matrix}$
C_3	$\mu_3 = [722.90, 501.69]^t$	$\mu_3 = [722.90, 501.69]^t$	$\Sigma_3 =$	$\begin{matrix} 24987.30 & 122031.60 \\ 122031.60 & 133911.50 \end{matrix}$
C_4	$\mu_4 = [2332.72, 2041.35]^t$	$\mu_4 = [2332.72, 2041.35]^t$	$\Sigma_4 =$	$\begin{matrix} 606823, 60 & 256480.40 \\ 256480.40 & 321025.90 \end{matrix}$
‘Confirmed-Death’ data				
C_1	$\mu_1 = [73.24, 2.86]^t$	$\mu_1 = [73.24, 2.86]^t$	$\Sigma_1 =$	$\begin{matrix} 4974.83 & 95.79 \\ 95.79 & 11.27 \end{matrix}$
C_2	$\mu_2 = [127.88, 0.88]^t$	$\mu_2 = [127.88, 0.88]^t$	$\Sigma_2 =$	$\begin{matrix} 11465.05 & 1.02 \\ 1.02 & 15.10 \end{matrix}$
C_3	$\mu_3 = [722.90, 12.58]^t$	$\mu_3 = [722.90, 12.58]^t$	$\Sigma_3 =$	$\begin{matrix} 249875.30 & 5027.91 \\ 5027.91 & 133.19 \end{matrix}$
C_4	$\mu_4 = [2332.72, 38.37]^t$	$\mu_4 = [2332.72, 38.37]^t$	$\Sigma_4 =$	$\begin{matrix} 606823.60 & 4415.75 \\ 4415.75 & 81.86 \end{matrix}$
‘Recovered-Death’ data				
C_1	$\mu_1 = [49.38, 2.46]^t$	$\mu_1 = [49.38, 2.46]^t$	$\Sigma_1 =$	$\begin{matrix} 4480.11 & -31.63 \\ -31.63 & 9.34 \end{matrix}$
C_2	$\mu_2 = [211.65, 3.32]^t$	$\mu_2 = [211.65, 3.32]^t$	$\Sigma_2 =$	$\begin{matrix} 30352.05 & 396.53 \\ 396.53 & 20.27 \end{matrix}$
C_3	$\mu_3 = [1724.95, 35.56]^t$	$\mu_3 = [1724.95, 35.56]^t$	$\Sigma_3 =$	$\begin{matrix} 485690.30 & 3820.91 \\ 3820.91 & 102.70 \end{matrix}$

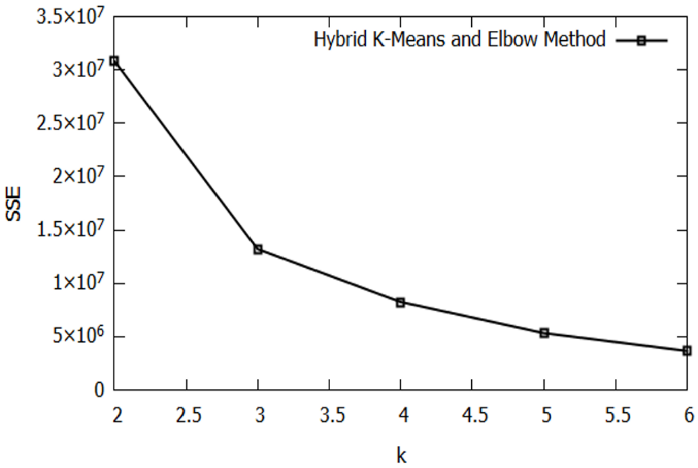


Fig. 7. Graph of sum square error depending on the number k for ’Recovered-Death’ data.

for “Recovered-Death” data can be seen in (Fig. 5), (Fig. 6) and (Fig. 7).

We used the hybrid K-Means algorithm and Elbow method, which gave best clustering with 4 and 3 clusters. This result is exploited in the EM classification based on GMM, we notice that the “Confirmed-Recovered”, “Confirmed-Death” and “Recovered-Death” can be divided into 4, 4 and 3 subsets, respectively. We analyze the correlation of feature variables for COVID-19, Correlation matrix is used to find the relationship between two variables “Confirmed-Recovered”, “Confirmed-Death” and “Recovered-Death”. Correlation Coefficient r is used to calculate the strength of this relationship between two quantitative variables $\boldsymbol{Y_i}$ and $\boldsymbol{Y_j}$ by using the formula given in (Eq. 13):

$$\boldsymbol{r} = \frac{(\boldsymbol{Y_i} - \boldsymbol{\mu_i})^{\boldsymbol{t}}(\boldsymbol{Y_j} - \boldsymbol{\mu_j})}{\sqrt{\|\boldsymbol{Y_i} - \boldsymbol{\mu_i}\|^2 \|\boldsymbol{Y_j} - \boldsymbol{\mu_j}\|^2}} \tag{13}$$

$\boldsymbol{i}$ and $\boldsymbol{j} =$ Confirmed, Recovered , Death r, the correlation coefficient is a unitless value between -1 and 1.

In Table II, we have the initial parameters of the different groups. To start the K-Means and EM algorithms, we use the same means values and the same coefficients of the found covariance matrix.

In this part, we aim to implement selected $\boldsymbol{C^{++}}$ object from [35] using K-Means algorithm is to partition the first, the second and the third group into four, four and three

TABLE III. VALUES AT CONVERGENCE FOR K-MEANS AND EM ALGORITHM

'Confirmed-Recovered' data				
	K-Means	EM GMM		
	Number of iterations: 86	Number of iterations: 445		
C_1	$\mu_1 = [157.36, 120.93]^t$	$\mu_1 = [17.81, 0.48]^t$	$\Sigma_1 =$	$\begin{matrix} 503.74 & 11.23 \\ 11.23 & 0.55 \end{matrix}$
C_2	$\mu_2 = [1323.33, 985.14]^t$	$\mu_2 = [135.36, 95.57]^t$	$\Sigma_2 =$	$\begin{matrix} 5532.70 & 1657.50 \\ 1657.50 & 5640.30 \end{matrix}$
C_3	$\mu_3 = [2152.97, 2074.56]^t$	$\mu_3 = [891.17, 660.67]^t$	$\Sigma_3 =$	$\begin{matrix} 320600.00 & 174070.00 \\ 174070.00 & 194950.00 \end{matrix}$
C_4	$\mu_4 = [3366.81, 2530.50]^t$	$\mu_4 = [2470.80, 2176.30]^t$	$\Sigma_4 =$	$\begin{matrix} 572850,00 & 175290.00 \\ 175290.00 & 249060.00 \end{matrix}$
'Confirmed-Death' data				
	Number of iterations: 88	Number of iterations: 456		
C_1	$\mu_1 = [148.60, 2.49]^t$	$\mu_1 = [88.88, 2.72]^t$	$\Sigma_1 =$	$\begin{matrix} 830.59 & 18.17 \\ 18.17 & 8.50 \end{matrix}$
C_2	$\mu_2 = [1315.52, 27.48]^t$	$\mu_2 = [114.66, 1.38]^t$	$\Sigma_2 =$	$\begin{matrix} 7680.60 & 72.23 \\ 72.23 & 2.21 \end{matrix}$
C_3	$\mu_3 = [2380.27, 35.77]^t$	$\mu_3 = [1124.20, 23.04]^t$	$\Sigma_3 =$	$\begin{matrix} 182120.00 & 4896.50 \\ 4896.50 & 180.76 \end{matrix}$
C_4	$\mu_4 = [3562.50, 50.58]^t$	$\mu_4 = [2599.90, 38.21]^t$	$\Sigma_4 =$	$\begin{matrix} 494200.00 & 5960.00 \\ 5960.00 & 116.10 \end{matrix}$
'Recovered-Death' data				
	Number of iterations: 77	Number of iterations: 219		
C_1	$\mu_1 = [140.29, 3.40]^t$	$\mu_1 = [59.08, 0.67]^t$	$\Sigma_1 =$	$\begin{matrix} 3067.70 & 10.24 \\ 10.24 & 0.61 \end{matrix}$
C_2	$\mu_2 = [1294.61, 31.66]^t$	$\mu_2 = [175.72, 4.22]^t$	$\Sigma_2 =$	$\begin{matrix} 26438.00 & -225.24 \\ -225.24 & 9.16 \end{matrix}$
C_3	$\mu_3 = [2403.27, 40.86]^t$	$\mu_3 = [1636.30, 33.23]^t$	$\Sigma_3 =$	$\begin{matrix} 602620.00 & 6191.30 \\ 6191.30 & 140.61 \end{matrix}$

TABLE IV. THE CORRELATION COEFFICIENTS OF THE COVID-19 DATA; FOR INITIAL VALUES WITH DIFFERENT CLUSTERS

	Confirmed cases					Confirmed cases				Recovered cases			
RC	0.40	0	0	0	DC	0.25	0	0	DC	0.40	0	0	0
RC	0	-0.16	0	0	DC	0	0.81	0	DC	0	0.14	0	0
RC	0	0	0.67	0	DC	0	0	0.69	DC	0	0	0.87	0
RC	0	0	0	0.58					DC	0	0	0	0.63

TABLE V. THE CORRELATION COEFFICIENTS OF THE COVID-19 DATA; FOR VALUES AT CONVERGENCE WITH DIFFERENT CLUSTERS

	Confirmed cases					Confirmed cases				Recovered cases			
RC	0.67	0	0	0	DC	0.24	0	0	DC	0.22	0	0	0
RC	0	0.30	0	0	DC	0	-0.40	0	DC	0	0.55	0	0
RC	0	0	0.70	0	DC	0	0	0.67	DC	0	0	0.85	0
RC	0	0	0	0.51					DC	0	0	0	0.78

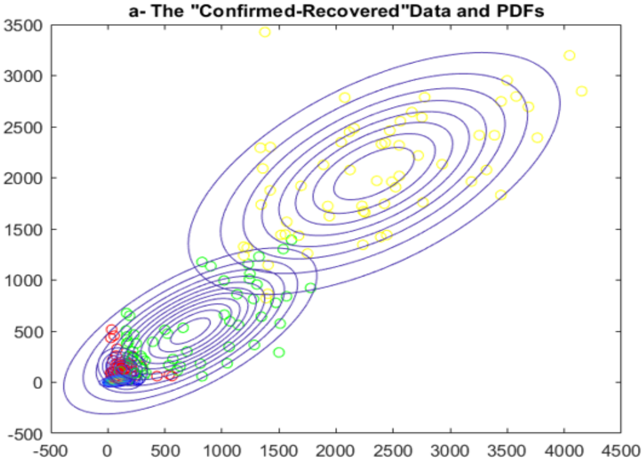


Fig. 8. Experiments result after implementation EM clustering for 'Confirmed-Recovered' 2-dimensional data generated by GMM with four mixture components. (a)-Graphs at initials values (b)-Graphs at convergences values.

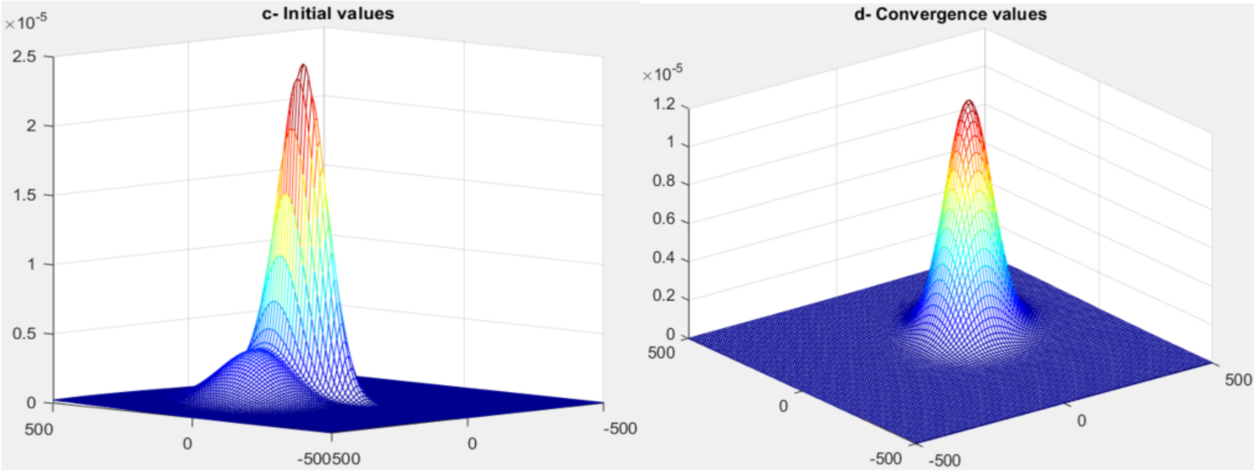


Fig. 9. Contours of probability density function (PDF) with four mixture components of "Confirmed-Recovered" data for (c) and (d) figures.

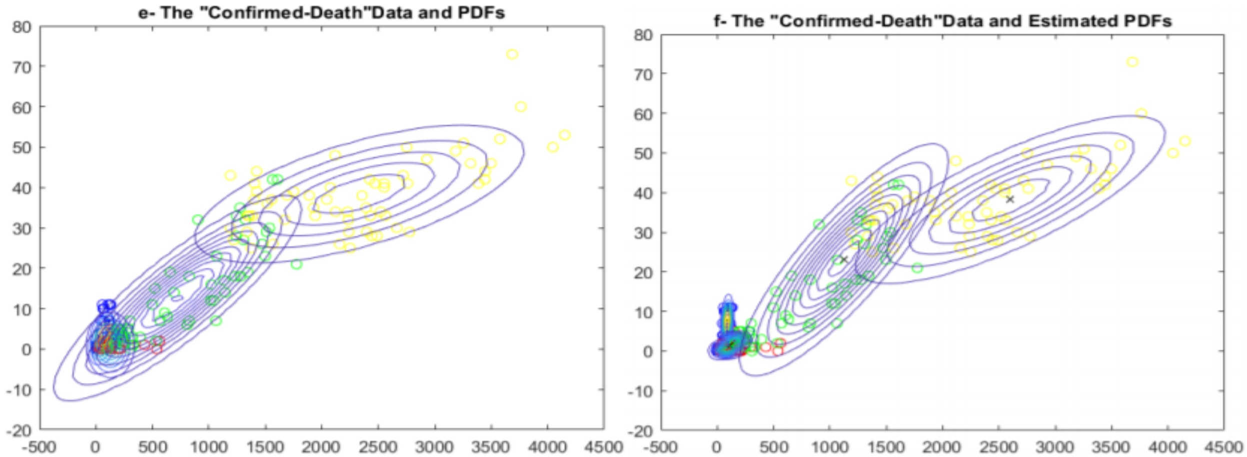


Fig. 10. Experiments result after implementation EM clustering for "Confirmed-Death" 2-dimensional data generated by GMM, with four mixture components. (e)-Graphs at initials values (f)-Graphs at convergences values.

clusters, respectively. Also, we apply EM by using GMM based on Matlab for all three groups 'Confirmed – Recovered'

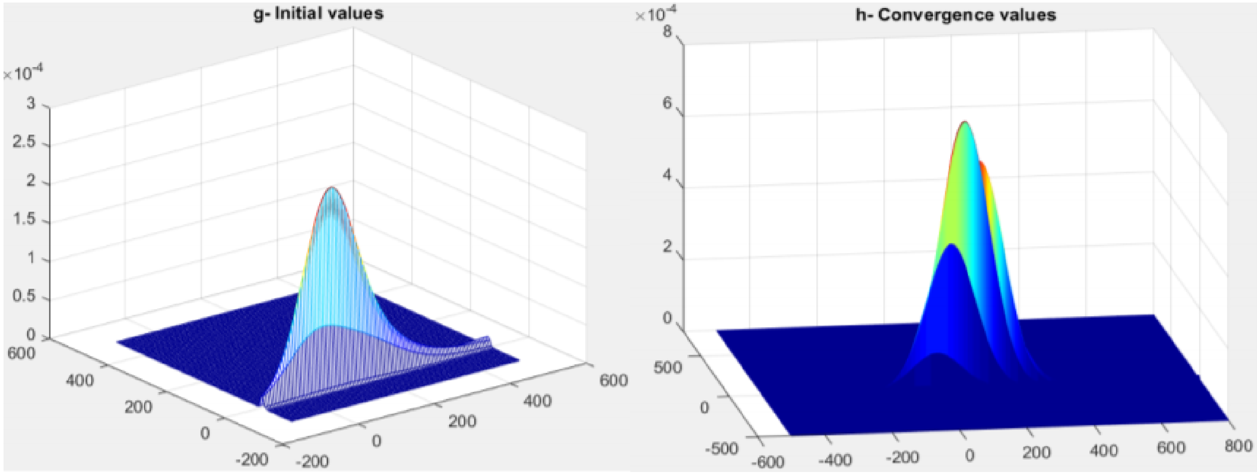


Fig. 11. Contours of probability density function (PDF) with four mixture components of “Confirmed-Death” data for (g) and (h) figures.

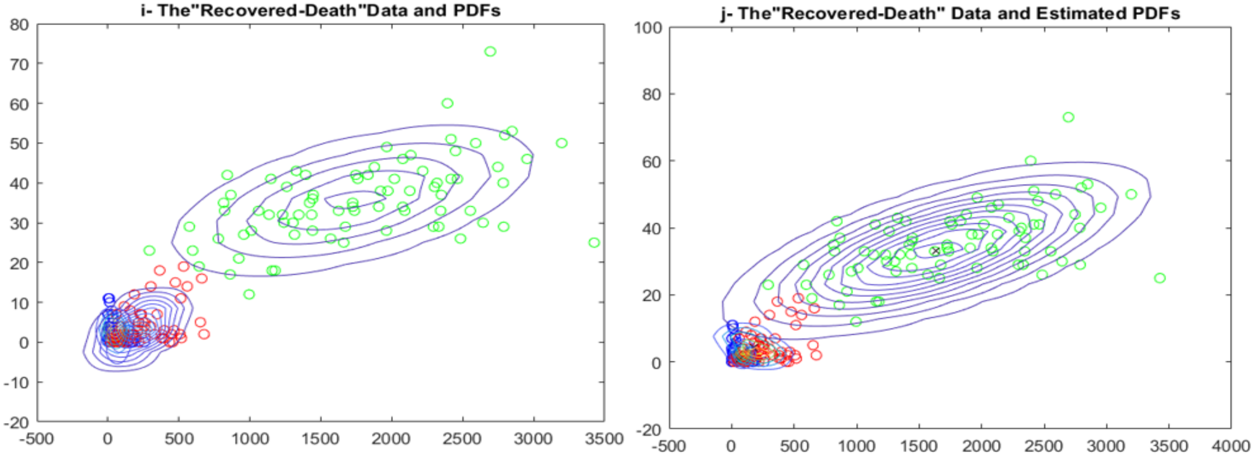


Fig. 12. Experiments result after implementation EM clustering for “Confirmed-Recovered” 2-dimensional Data generated by a GMM, with four mixture components. (i)-Graphs at initials values (j)-Graphs at convergences values.

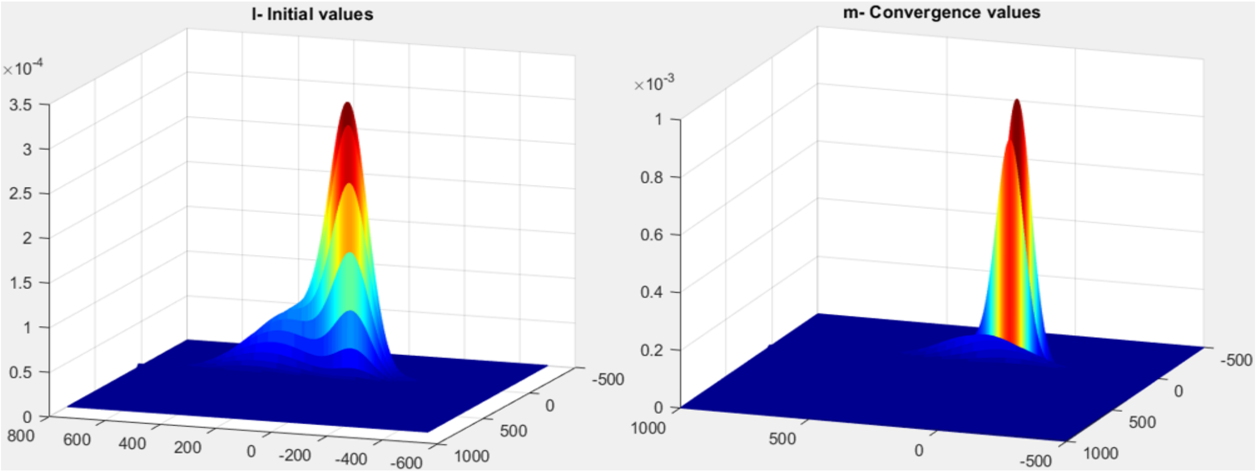


Fig. 13. Contours of probability density function (PDF) with four mixture components of “Confirmed-Recovered” data for (l) and (m) figures.

(see Fig. 8, 9), ‘Confirmed – Death’ (see Fig. 10, 11), and ‘Confirmed – Recovered’ (see Fig. 12, 13).

We obtain values at convergence by using K-Means algorithm and EM algorithm (see Table III).

The correlation matrix of the initial values (see Table IV), and values at convergences (see Table V) for features "Confirmed – Recovered – Death" cases using (Eq. 13):

Positive values of r indicate a positive correlation, as the values of the two variables tend to increase together. Negative values of r indicate a negative correlation when the values of one variable tend to increase and the values of the other variable decrease. In the data mining of COVID-19 in Morocco, K-Means is a simple and fast algorithm for solving clustering issues, but it requires clarification in advance the exact number of clusters k, which is often difficult.

The "Confirmed – Recovered", "Confirmed – Death" and "Recovered – Death" groups are Mixtures Models of four and three two-dimensional Gaussians. K-Means algorithm only considers the mean to update the new centroids nevertheless EM based GMM takes into account the mean value as well as the covariance matrix of this data groups. We use this partitioning to start K-Means and EM. We start from a real model with correlated covariance matrices, the values at convergence are of the same nature. It can be interpreted that high positive correlation exists, in the third phase of the epidemic's spread, between Confirmed cases and Recovered cases (0.70), Confirmed cases and Death cases (0.67) and Recovered cases and Death cases (0.85) [10]. To evaluate clusters "Confirmed – Recovered" and "Recovered – Death"; values are in forms four categories (low, lower-middle, uppermiddle, and high), on the order hand "Confirmed – Death" data is in forms three phase (low, medium, and high).

We notice a clear difference between means of the K-Means algorithm and the means of the GMM. The EM based GMM has higher computation time than K-Means; because K-Means does not account for variance. The findings are in according with those of [19]. The Data membership points to clusters in GMM is probabilistic as versus the non-probabilistic, hard clustering K-Means process, thus resolving the membership vagueness that may appear in overlapping clusters. The analysis exposes a more meaningful workloads clustering with GMM than with K-Means, enabling a detailed characterization of resource usage needs of Cloud workload. As a comparison, the clustering by using K-Means algorithm is faster than Gaussian Mixture Models method.

K-Means clustering faces a major challenge in determining the optimal number of clusters, especially when working with COVID-19 data. Depending on the type of data being analyzed, the number of clusters may vary, and selecting the correct number of clusters is crucial for obtaining meaningful results. Furthermore, K-Means clustering relies on the Euclidean distance metric, which may not be suitable for all COVID-19 data. Other distance metrics, such as cosine distance, may be necessary to accurately capture the similarity between data points. Another clustering algorithm, EM clustering, is also sensitive to the initial conditions of the algorithm. Different initial conditions may result in different cluster assignments, leading to inconsistent results. Additionally, EM clustering may struggle to converge to a solution when working with high-dimensional data or complex probability distributions. Preprocessing and tuning of the algorithm may be necessary to ensure reliable results.

VII. Conclusion

This study focuses on analyzing the COVID-19 situation in Morocco using K-Means and EM clustering algorithms. The dataset includes daily Confirmed, Death, and Recovered cases from March 2 to October 24, 2020. For the k-means algorithm, discovering intra-cluster similarity in complex nonlinear models using Euclidean distance is difficult. The EM algorithm is more computationally intensive and requires larger sample sizes for accurate parameter estimates. The results indicate that the EM-based GMM method is the preferred clustering method as it yields smaller classification error rates. The K-Means generated clusters provide limited information, and the best clustering was found with four and three clusters. Furthermore, the EM algorithm demonstrates the correlation between "Confirmed-Recovered", "Confirmed-Death", and "Recovered-Death". The number of clusters corresponds to the number of phases of the epidemic propagation, as determined by the process of identifying the optimal number of clusters. In the future work, we will be focused on the enhancement of our model clustering for multi-dimensional datasets with several features.

References

[1] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[2] G. Gan and E. A. Valdez, "Data clustering with actuarial applications," *North American Actuarial Journal*, vol. 24, no. 2, pp. 168–186, 2020.

[3] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[4] K. Chowdhury, D. Chaudhuri, A. K. Pal, and A. Samal, "Seed selection algorithm through k-means on optimal number of clusters," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18 617–18 651, 2019.

[5] H. Jiang and E. Arias-Castro, "$\boldsymbol{k}$-means and gaussian mixture modeling with a separation constraint," *arXiv preprint arXiv:2007.04586*, 2020.

[6] M. Hamidi, H. Satori, O. Zealouk, and K. Satori, "Amazigh digits through interactive speech recognition system in noisy environment," *International Journal of Speech Technology*, vol. 23, no. 1, pp. 101–109, 2020.

[7] M. Hamidi, H. Satori, O. Zealouk, K. Satori, and N. Laaidi, "Interactive voice response server voice network administration using hidden markov model speech recognition system," in *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2018, pp. 16–21.

[8] M. Amouch and N. Karim, "Modeling the dynamic of covid-19 with different types of transmissions," *Chaos, Solitons & Fractals*, p. 111188, 2021.

[9] A. Xavier Jr, "A c++ code for predicting covid-19 cases by least-squares fitting of the logistic model," *Pre-print available on Research Gate (https://www. researchgate. net/). DOI*, vol. 10, 2020.

[10] A. Rizvi, M. Umair, and M. A. Cheema, "Clustering of countries for covid-19 cases based on disease prevalence, health systems and environmental indicators," *medRxiv*, 2021.

[11] J. P. F. Arocutipa, J. J. Huallpa, G. C. Navarro, and L. D. B. Peralta, "Clustering k-means algorithms and econometric lethality model by covid-19, peru 2020," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021.

[12] P. V. Sagar, T. P. Kumar, G. K. Chaitanya, and M. N. Rao, "Covid-19 transmission risks assessment using agent-based weighted clustering approach," 2020.

[13] M. Zubair, M. Asif Iqbal, A. Shil, E. Haque, M. Moshiul Hoque, and I. H. Sarker, "An efficient k-means clustering algorithm for analysing covid-19," in *Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020*. Springer, 2021, pp. 422–432.

[14] B. A. Hassan, T. A. Rashid, and H. K. Hamarashid, “A novel cluster detection of covid-19 patients and medical disease conditions using improved evolutionary clustering algorithm star,” *Computers in biology and medicine*, vol. 138, p. 104866, 2021.

[15] R. Kurniawan, S. N. H. S. Abdullah, F. Lestari, M. Z. A. Nazri, A. Mujahidin, and N. Adnan, “Clustering and correlation methods for predicting coronavirus covid-19 risk analysis in pandemic countries,” in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*. IEEE, 2020, pp. 1–5.

[16] E. Alsuwat, S. Alzahrani, and H. Alsuwat, “Detecting covid-19 utilizing probabilistic graphical models,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[18] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[19] E. Patel and D. S. Kushwaha, “Clustering cloud workloads: K-means vs gaussian mixture model,” *Procedia Computer Science*, vol. 171, pp. 158–167, 2020.

[20] S. K. Appiah, K. Wirekoh, E. N. Aidoo, S. D. Oduro, and Y. D. Arthur, “A model-based clustering of expectation–maximization and k-means algorithms in crime hotspot analysis,” *Research in Mathematics*, vol. 9, no. 1, p. 2073662, 2022.

[21] V. Zarikas, S. G. Poulopoulos, Z. Gareiou, and E. Zervas, “Clustering analysis of countries using the covid-19 cases dataset,” *Data in brief*, vol. 31, p. 105787, 2020.

[22] S. Aungkulanon, V. Tangcharoensathien, K. Shibuya, K. Bundhamcharoen, and V. Chongsuvivatwong, “Post universal health coverage trend and geographical inequalities of mortality in thailand,” *International journal for equity in health*, vol. 15, no. 1, pp. 1–12, 2016.

[23] A. Malav, K. Kadam, and P. Kamat, “Prediction of heart disease using k-means and artificial neural network as hybrid approach to improve accuracy,” *International Journal of Engineering and Technology*, vol. 9, no. 4, pp. 3081–3085, 2017.

[24] R. Singh and E. Rajesh, “Prediction of heart disease by clustering and classification techniques prediction of heart disease by clustering and classification techniques,” *International Journal of Computer Sciences and Engineering*, 2019.

[25] S. Y. Işikhan and D. Güleç, “The clustering of world countries regarding causes of death and health risk factors,” *Iranian Journal of Public Health*, vol. 47, no. 10, p. 1520, 2018.

[26] Y. G. Jung, M. S. Kang, and J. Heo, “Clustering performance comparison using k-means and expectation maximization algorithms,” *Biotechnology & Biotechnological Equipment*, vol. 28, no. sup1, pp. S44–S48, 2014.

[27] I. B. Mohamad and D. Usman, “Standardization and its effects on k-means clustering algorithm,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 17, pp. 3299–3303, 2013.

[28] C. M. Bishop, “Pattern recognition,” *Machine learning*, vol. 128, no. 9, 2006.

[29] R. O. Duda, P. E. Hart *et al.*, *Pattern classification*. John Wiley & Sons, 2006.

[30] R. Lu, X. Zhao, J. Li, P. Niu, B. Yang, H. Wu, W. Wang, H. Song, B. Huang, N. Zhu *et al.*, “Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding,” *The lancet*, vol. 395, no. 10224, pp. 565–574, 2020.

[31] D. Paraskevis, E. G. Kostaki, G. Magiorkinis, G. Panayiotakopoulos, G. Sourvinos, and S. Tsiodras, “Full-genome evolutionary analysis of the novel corona virus (2019-ncov) rejects the hypothesis of emergence as a result of a recent recombination event,” *Infection, Genetics and Evolution*, vol. 79, p. 104212, 2020.

[32] W. H. Organization, “Coronavirus disease (covid-19): situation report, 209,” 2020.

[33] M. H. Ministry, “http://www.covidmaroc.ma/ (last accessed: November 30 2020, 17:00 gmt),” 2020.

[34] M. Syakur, B. Khotimah, E. Rochman, and B. D. Satoto, “Integration k-means clustering method and elbow method for identification of the best customer profile cluster,” in *IOP Conference Series: Materials Science and Engineering*, vol. 336, no. 1. IOP Publishing, 2018, p. 012017.

[35] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes with Source Code CD-ROM 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

Chicken Behavior Analysis for Surveillance in Poultry Farms

Abdallah Mohamed Mohialdin[1], Abdullah Magdy Elbarrany[2], Ayman Atia[3]
HCI-LAB, Faculty of Computers and Artificial Intelligence, Helwan University[3]
Faculty of Computer Science, October University for Modern Sciences and Arts (MSA)
Giza, Egypt[1,2,3]

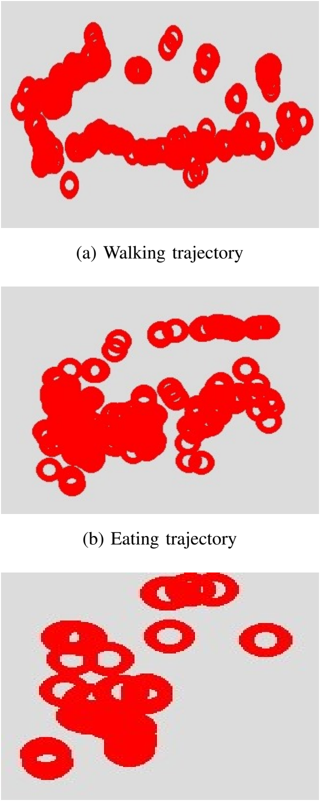
***Abstract*—Poultry farming is an important industry that provides food for a growing population. However, the welfare of the birds is a major concern, as poor living conditions leads to abnormal behavior that affects the health and productivity of the flock. In order to monitor and improve the welfare of the birds, it is important to have a surveillance system in place that monitors the behavior of the chickens and alert farmers to potential issues. This paper reviews the current state of the art in behavior analysis for surveillance in poultry farms and discuss potential future directions for research in this area. This paper presents a computer-vision-based system that detects and monitors the behaviors of the chickens in poultry farms. The system classifies three behaviors which are eating, walking and sleeping. The system takes videos as input and then classifies the behavior of the chicken. The proposed system produces an accuracy of 94.7% using Light Gradient Boosting Machine on a collected data-set of chickens, and a 98.4% accuracy on a benchmarked Human Activity Recognition data-set.**

Keywords*—*Chicken; poultry; abnormal; behavior; birds

I. Introduction

The study of chicken behavior plays a critical role in the poultry industry, as it provides valuable insights into the health and welfare of the birds. Farmers can detect any abnormal patterns that indicate illness, stress, or discomfort by analyzing chicken behaviors. This information is then used to improve their health condition, and ultimately, enhance production efficiency and profitability. In addition, an effective system for detecting and analyzing chicken behavior contributes in early disease detection, preventing outbreaks [1], ensuring food safety, protecting public health [2], and ensuring the quality of poultry production [3]. Hence, having a reliable system for monitoring chicken behavior is essential for sustainable poultry farming and ensuring the well-being of the birds.

The Food Standards Agency (FSA) reported that 1.35 million chickens die prior to being slaughtered, primarily due to diseases and illnesses [4]. This conclusion was drawn from a 15-month study conducted between 2016 and 2017. Ensuring the health of chickens is achieved by monitoring their behavior. Monitoring chicken behaviors by extracting, and analyzing their trajectories (as shown in Fig. 1), is used as an indicator to ensure their health and well-being. Although the chickens seem fine, sudden death occurs quickly if specific behaviors are displayed, leaving no chance for intervention [5].



(a) Walking trajectory

(b) Eating trajectory

(c) Sleeping trajectory

Fig. 1. Different chicken behavior trajectories.

Different methods for analyzing chicken behavior are utilized (as shown in Fig. 2), one such approach involves using pose estimation as used by Fang et al. [10], where the key points of the chickens are identified and utilized to monitor their behavior. Another method is to analyze the collective behavior of chickens within a farm environment. The manner in which chickens move as a group provides insights into their behaviors. The proposed system (as shown in Fig. 3) primarily concentrates on the third and final approach, which involves tracking the movements of each chicken trajectory individually. The individual actions of the chickens serve as an indicator of their behavior. Fig. 2 shows the previously mentioned methods.

Previous researches that tackled this problem mostly utilized image processing and machine learning to detect and classify chicken behaviors. As illustrated in Table I, Neethira-

TABLE I. Previous Deep Learning-based Approaches for Animal Detection

Algorithm	Application	Dataset Size	Dataset Availability	Accuracy	Reference
Yolov5, Kalman filter	Detecting &Counting	72 Chickens 8 hours x30 FPS	Not available	96%	[6]
Yolov4, Kalman filter	Identify chicken's movement in low light	6 laying hens 1296 frames	Not available	99%	[7]
Yolov5, Detectron2	Quail detection mechanism	5,332 frames	Not available	85.07%,67.15% Average precision	[8]
Yolov5, ResNet18	Trajectory of polar bears	4450 frames	Available	86.4% F1 score	[9]
DeepLabCut and Pre trained ResNet50	Chicken pose estimation and behavior classification	4450 frames	Not available	75% standing,92% preening, 51% walking,96% resting, 62% running,93% eating	[10]
SVM	Prediction of broiler disease	34,280 frames	Not available	97%	[11]
Yolov5,logistic regression ML	Detecting chickens,behaviors	6 hours x30 FPS	Not available	94.7%	Proposed work

jan [6] put forward a system for detecting and counting chickens using Yolov5 and Kalman Filter. Siriani [7] also employed Yolov4 and Kalman filter to monitor chickens' movements in low light conditions. Meanwhile, Fang et al. proposed a system that classifies chicken behavior by analyzing their poses using DeepLabCut [12] and ResNet50 [13].

Table I presents a GAP analysis, which compares the proposed system work with previous studies on animal (chicken) detection and tracking. The table compares the algorithms, applications, data-set sizes, data-set availability, and achieved accuracies used in each previous work. Evaluating the strengths and limitation of each existing approach is achieved by comparing the mentioned factors.

The main contribution of this paper is to investigate the use of chicken's trajectories in order to classify its behavior using classical machine learning models. The proposed system (as shown in Fig. 3) aims to classify three chicken behaviors using their trajectories: walking (as shown in Fig. 1a), eating (as shown in Fig. 1b), and sleeping (as shown in Fig. 1c) (Fig. 4 also shows practical examples of the aforementioned behaviors). This paper utilizes a collected data-set of 90 records split evenly among the three mentioned behaviors. The data-set is to be further explained in the Data Collection section. The proposed model is also evaluated on a benchmarked data-set which is the Human Activity Recognition data-set [14] in order to evaluate its efficiency. Furthermore, a comparison between different classical machine learning models is also conducted in order to determine the highest performing model.

The rest of the paper is organized as follows: Section II discusses the literature review and previous related work, Section III outlines the proposed system's methodology, data collection, and how the system works, Section IV presents the setup and results of the two conducted experiments, Section V discusses the analysis and results produced by the classical machine learning models, Section VI presents the limitation of existing approaches and potential future work ideas, and Section VII states the conclusion of the paper.

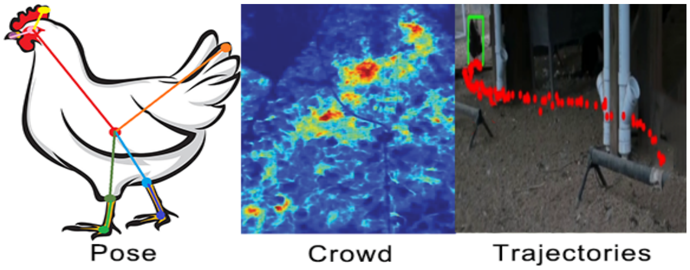


Fig. 2. Methods of chickens behavior classification.

II. Related Work

In recent years, many studies have focused on detecting and tracking animal behavior with the aim of further understanding the animals and detecting their behavior. The aim of these studies is to develop deep learning models for detecting and tracking animal behaviors. In this section, some of the previous work in this area is presented.

Neethirajan proposed a model which detects chickens, counts them, and extracts each chicken's movement from their bounding box [6]. Neethirajan's model detected chickens from various backgrounds using YOLOv5, and then gave each chicken an ID associated with the bounding box. As a result, the model was able to count and track chickens. To detect each chicken's direction, the proposed system stored the centroid of each bounding box. A challenge this model faced was occlusion, where a chicken disappears in one frame and return in the next frame, resulting in the chicken being given a new ID. To solve this challenge, a Kalman filter was used to compute the distance between each centroid and the old one, in order to check if the chicken had been tracked before.

Siriani et al. proposed a model to detect and track chickens in low light [7]. The proposed model used YOLOv4 in order to detect chickens from input frames in addition to returning the bounding box positions to store in CSV files for the tracking process. Kalman filter was used to predict the next position of each chicken depending on it's last positions. This model scored a very high accuracy of 99.9%. However the data-set used was to small in addition to using 10,000 epochs which

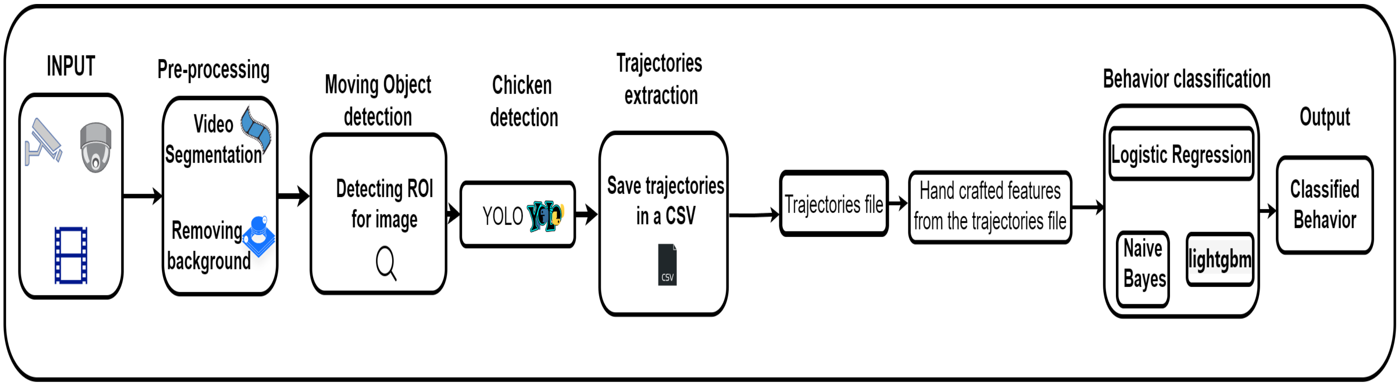


Fig. 3. System architecture.

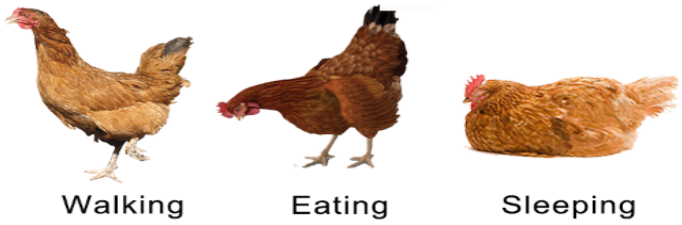


Fig. 4. Different chickens behaviors.

probably led to over-fitting for their model.

Evangelista et al. proposed a model for detecting quails using YOLOv5 and Detectron2 [8]. The proposed system combined YOLOv5 and Detectron2 to create a faster model for training and validation. However, the model failed to attain a high accuracy and detected touching quails as a single quail.

Zuerl et al. proposed a model for monitoring polar bear behavior using YOLOv5 and ResNet18 [9]. The suggested system consisted of four stages: object detection, classification, coordinate mapping, and analysis and user interface. In the first stage, YOLOv5 was used to detect polar bears. In the second stage, the image was cropped to only include the detected animal, and then ResNet18 was used to classify the animal type. After that, the center of one of the edges in the bounding box was used as the position of the polar bear. Following that, the extracted position was evaluated with respect to the map of the zoo to determine its location. The final stage was the user interface, which displayed the bear's location in the form of a heat map or trajectory, the distance covered in a selected time frame, and the bear's activity. Their model failed to achieve high accuracy. Furthermore, the mapping stage of their approach is difficult to implement, as a map of the farm is required.

The previous work mentioned methods primarily focused on detecting animals and determining their trajectories, without considering any behaviors. In contrast, Fang et al. proposed a model for chicken behavior classification, along with the estimation of chicken poses, using DeepLabCut and a pre-trained ResNet50 model [10]. As the first step, they created a basic skeletal map of the stance of broiler chickens. The ResNet-50 was then trained on the data-set, and DeepLabCut was used to predict the chicken's body position. Afterward, the chicken's posture was fed into a Naive Bayes model to determine the behavior to which it belongs. However, the limitations of this approach were the low accuracy scores in detecting running and walking behaviors, as the postures for these two classes are similar and difficult to separate.

Okinda et al. presented a distinct approach from the work discussed earlier [11]. This approach employed a support vector machine (SVM) to predict broiler diseases. Unlike the previous methods, this model detects chickens by analyzing images from the perspective of shape representation. To predict the disease, six models were experimented with, and the SVM scored the highest results. However, the model also missclassified some inputs as incorrect days.

However, to date, there has been no published study on behavior detection using chicken trajectories. This paper presents a novel approach for detecting chickens, extracting their trajectories, and classifying those trajectories according to their respective behaviors.

III. Methodology

The system proposed in this paper consists of six stages: pre-processing, background subtraction and ROI detection, chicken filtering, trajectory extraction, behavior classification, and output Fig. 3. In this section, a comprehensive explanation of each stage is provided.

A. Data Collection

The data-set used in this study was obtained from YouTube videos of chicken barns, totaling 6 hours of video footage. To establish ground truth, the video footage was labeled and trajectories were created with labels such as walking, eating, and sleeping. Each class comprised 30 records, resulting in a total of 90 records. Each record contains the (x, y) positions of the chicken during the recorded clips. The data-set was then split, with 80% used for training and 20% for testing.

For validation purposes, a human activity recognition data-set [15] was used as a benchmark to validate the proposed model. The handcrafted features were calculated from the raw time domain data. The data-set was constructed from 270 records, which contained four behaviors: standing still, walking, jumping, and running, with each class having an equal number of records. The data-set was split in the same manner

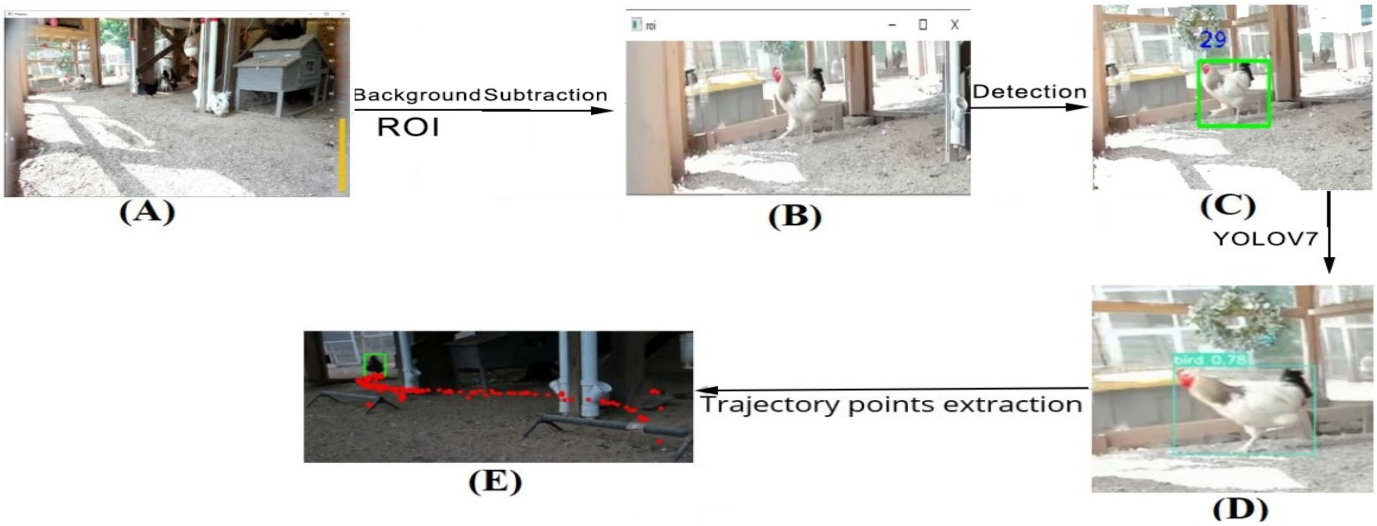


Fig. 5. System stages: (A) Input segmentation, (B) Background subtraction & ROI, (C) Moving object detection, (D) Chicken filtering, (E) Output.

as the constructed data-set, with 80% for training and 20% for testing.

B. Pre-Processing

The proposed system has a pre-processing stage, which requires video segmentation as its first step. The objective is to analyze the content of the video. To achieve the desired results, the video undergoes processing. The fundamental procedure for processing a video is video segmentation [16]. The input videos are segmented into intervals of 5, 10, and 15 seconds for processing. The results showed that the 10-second segmentation achieved the best outcome, as it was observed that chickens typically transition to a different behavior after 10 seconds.

C. Background Subtraction and ROI

After video segmentation, the next step is to apply background subtraction to the videos. Background subtraction is commonly used to isolate objects of interest in a scene by comparing an observed image to an estimated version of the image that does not contain any objects of interest [17], [18]. The purpose of applying background subtraction is to determine the trajectory of the chickens as shown in Fig. 5(B).

The process of applying the Region of Interest (ROI) to a moving object is utilized to eliminate any irrelevant data present in the image. In addition to detecting any moving object as demonstrated in Fig. 5(C).

D. Chicken Filtering

To detect any moving objects. The object detection stage is then performed to determine if the moving object is a chicken. This stage is conducted using YOLO, and if the object detected is a chicken, it is assigned an ID, as shown in Fig. 5(D).
Joseph Redmon and others first introduced YOLO in 2016 [19]. By partitioning an image into a grid and producing predictions for each cell in the grid, the algorithm is made to recognise objects in real-time. The distinctive feature of YOLO is that it only makes one forward pass over the network, enabling real-time image processing. Other object identification methods, on the other hand, call for many forward passes, which makes them slower and less suited for real-time applications. On a number of object detection benchmarks, YOLO has been demonstrated to attain state-of-the-art performance while also being computationally effective. Its widespread application in numerous computer vision tasks is a result of this. In addition to its benefits include quick deep learning network processing times, the capacity to handle larger data-sets, and real-time continuous detection [20], [21].

$$O = \sqrt{(Xnew - Xold)^2 + (Ynew - Yold)^2} \qquad (1)$$

Equation 1: Euclidean Distance Equation

The ID of a chicken changes in the case of occlusion and is assigned a new ID [6]. To overcome this challenge, the Euclidean distance is calculated between the chicken's previous location and its new location [22]. If the output of equation 1 is compared to the positions of all existing IDs and found to be less than 150, the chicken is assigned the same ID and the ID's position is updated.

E. Trajectories Extraction

The first step in trajectories extraction stage is to detect ROI(region of interest) from the image. The ROI is determined by the boundary box added in the previous stage by YOLO.

Next, the location of the chickens is assumed to be at the right edge of the bounding box, and a trajectory point is added every 10 frames. As previously mentioned, processing every frame is a challenging task, and thus trajectory extraction is only performed once every 10 frames [23].Finally, the trajectories are saved in a CSV (Comma-Separated Values) file to calculate features from it.

$$A = \frac{1}{n}\sum_{i=1}^{n} a_i = \frac{a_1 + a_2 + \cdots + a_n}{n} \tag{2}$$

Equation 2: Mean equation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{3}$$

Equation 3: Standard deviation equation

$$M = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4}$$

Equation 4: Median equation

F. Hand Crafted features

Since the trajectories are considered time series data [24], several time series features are calculated, such as the mean, median, standard deviation, minimum, and maximum. Mean is used to calculate the average distance covered as shown in equation 2 where a1, a2, ... and is calculated using the euclidean distance mentioned in equation 1. Median is used to extract the center element from a list of numbers using equation 4. In this case median is used to achieve the medium distance from all distances calculated. Standard deviation is used to determine how close the values are to the average (mean) value. The standard deviation is calculated using the equation 3.

$$DC = \sum_{i=1}^{n} \sqrt{(Xnew - Xold)^2 + (Ynew - Yold)^2} \tag{5}$$

Equation 5: Distance Covered equation

$$\begin{cases} SS+ = 1 & \text{if } \sqrt{(Xnew - Xold)^2 + (Ynew - Yold)^2} <= 50 \\ SS+ = 0 & \text{otherwise} \end{cases} \tag{6}$$

Equation 6: Standing still Equation

In addition, some original features were calculated, such as standing still, distance covered, maximum distance covered, and starting-ending. The standing still feature was calculated by subtracting every two consecutive trajectory points, and if the result was less than 50, the chicken is considered to be standing still as shown in equation 6. The distance covered feature was calculated by summing all of the trajectory points as presented in equation 5. The maximum distance covered feature was calculated by determining the maximum difference between each of the consecutive trajectory points. Finally, the starting-ending feature was calculated by subtracting the first trajectory point from the last trajectory point.

After combining and evaluating these features, the best results were achieved by using the following set of features: standing still, distance covered, mean, maximum distance covered, starting-ending position, and label.

G. Behavior Classification

In order to classify the behaviors, three classical machine learning algorithms were used: Logistic Regression, Naive Bayes, and LightGBM.

Classification is the main application of logistic regression. Unlike linear regression, logistic regression data points are not arranged in straight lines. Each group represents a category, and each data point belongs to a specific category. The objective of logistic regression is to find the classification boundary line, which is represented by the regression equation [25], [26]. Because of its distinctive way of expressing uncertain knowledge, its extensive capacity for expressing probability, and its incremental learning features for integrating prior knowledge, the Bayesian technique has quickly become one of the most alluring focal points in many methods. One of the original Bayesian classification algorithms, the Naive Bayesian Classification Algorithm, has a straightforward structure and great processing efficiency. A Naive Bayes classifier has the advantage of only needing to estimate relevant parameters, such as the mean and variance of variables, based on a small sample of training data. The assumption of independent variables means that only the procedure for estimating each variable is required, and that the entire covariance matrix is unnecessary [27], [28].

LightGBM is a new model that is based on the gradient learning framework and decision trees, and implements the concept of boosting. It differs from the XGBoost model in its use of histogram-based methods to accelerate training, conserve memory, and achieve a leaf wise growth approach with depth limitations. The histogram method involves discretizing continuous floating-point eigenvalues into k bins, which results in a histogram with a width of k. This algorithm preserves the discretized feature values using 8-bit integers, reducing memory consumption without sacrificing accuracy, without the need for additional storage of pre-sorted results [29], [30].

H. Output

After processing the input, the extracted features are fed into the model to classify its behavior.

IV. Analysis and Results

Two experiments were performed using the same approach that was previously explained. Input videos were divided into 10-second segments. Experiment 1, which focused on chicken behavior, had a sample size of 12 hours footage from a chicken barn, where the selected behaviors were cropped in order to conduct the experiment. Experiment 2 involved the Human Activity Recognition in order to check the efficiency of the proposed system, using the HAR data-set, both experiments also used the same hand crafted features. After pre-processing the segmented videos as shown in Fig. 3, and the trajectories of the chicken were extracted, they were passed to the classical machine learning models in order to determine the highest performing model.

A. Experiment 1

Experiment 1 aims to classify the three previously mentioned chicken behaviors, while also comparing between different classical machine learning models.

1) Setup: The first experiment uses the collected chicken data-set to test the efficiency of the proposed system. It was carried out using 90 records with 30 records allocated for each behavior and split 80-20 between training and testing, respectively. All classical machine learning models were trained and tested on the same data-set, the aforementioned models which are Logistic regression, Naive Bayes, and Light Gradient Boosting Machine (as shown in Fig. 3), were selected in order to determine the best performing model in this particular research.

2) Results: The performance of three different machine learning algorithms, namely Naive Bayes, Logistic Regression, and Light Gradient Boosting Machine (LightGBM), were evaluated in this study for the classification of chicken behavior. The results showed that LightGBM outperformed both Naive Bayes and Logistic Regression in terms of accuracy. Results are shown in Table II.

B. Experiment 2

The second experiment was conducted in order to have the proposed system evaluated on a public and benchmarked data-set.

1) Setup: The second experiment makes use of the Human Activity Recognition data-set. It was conducted using 270 records divided evenly between all 4 behaviors which are StandingStill, Walking, Jumping and Running. The records assigned to each behavior was also split 80-20 for training and testing respectively.

2) Results: In Experiment 2, the performance of the same three classical machine learning models were evaluated: Logistic Regression, Naive Bayes, and LightGBM. The results, as shown in Table III, indicate that Logistic Regression outperformed both Naive Bayes and LightGBM.

TABLE II. Chicken Dataset Results

Model	Accuracy
LightGBM	94.7%
Naive Bayes	91.3%
Logistic regression	89.8%

TABLE III. Comparing Results with Previous Work

Refrence	Accuracy
[31]	98.99%
Logistic regression	98.4%
LightGBM	93.9%
Naive Bayes	90.1%

V. Discussion

A. Experiment 1

LightGBM performed better than both Naive Bayes and Logistic regression (as shown in Table II) in experiment 1 which was conducted to assess the performance of the proposed system on classifying chicken behavior, this is attributed to several factors. Firstly, LightGBM is a gradient boosting algorithm that builds decision trees sequentially to correct the errors made by previous trees and then work on the leaves of these trees, where each node in the tree is a hand-crafted feature on its own. This results in a more robust model that captures complex and non-linear relationships between the features and the target. Secondly, LightGBM has a tree-pruning mechanism that helps to prevent over-fitting on small data-sets. By removing branches with low information gain, LightGBM is able to reduce the size of the tree and avoid over-fitting, leading to better generalization on unseen data.

On the other hand, Naive Bayes and Logistic Regression are linear models that assume a linear relationship between the features and target, which is not true in this particular experiment, and results in underfitting when the relationship between the features and the target label is more complex. Naive Bayes classifier also considers each of these features to contribute individually in the predicting process, which loses information about how the hand-crafted features are used together to identify the behavior.

B. Experiment 2

The results of Experiment 2 (as shown in Table III) show that, unlike Experiment 1, Logistic regression scored higher accuracy than both LightGBM and Naive Bayes in the classification of human activities. This improved performance of Logistic Regression compared to Experiment 1 is attributed to the larger data-set size in Experiment 2. In Experiment 1, the small data-set size have resulted in Logistic Regression being underfit, however, with the increased data-set size in Experiment 2, Logistic Regression was able to show higher accuracy. Logistic Regression also robust to outliers and has the ability handle real-world noisy data-sets, which is why it was suitable in this experiment.

The accuracy of both Naive Bayes and LightGBM models stayed around the same percentage (as shown in both Table II, and Table III), due to the same reasons that were mentioned in the discussion section for experiment 1.

VI. Limitations and Future Work

In this section, the limitations encountered during this research will be outlined, along with potential areas for future exploration for interested researchers.

The paper highlights the potential of using a computer-vision system and machine learning for chicken behavior analysis and surveillance in poultry farms. These technologies can provide valuable insights and serve as useful tools for farmers and researchers. However, there are also several limitations and challenges associated with this approach, which should be taken into consideration. Chickens overlapping: Chickens often move around in groups and can block each others movements (as illustrated in Fig. 6), making it challenging for computer vision algorithms to track and detect individual behaviors. In addition, occlusion can occur when one chicken is partially or completely obstructed from view by another chicken or object in the environment.

Chicken behaviors variability: chickens exhibit a wide range of behaviors, these different behaviors look very similar to each other (as shown in Fig. 1), which further complicates their classification, some of these behaviors are also not feasible to classify using only their trajectories, which means that a future work combining the proposed system with a pose estimation-based system is needed.

Future research to broaden the scope of this study is including the classification of abnormal behaviors observed in poultry farms. Additionally, other machine learning algorithms, larger data-sets, and more complex behavior classification tasks to be explored to further enhance the accuracy and practical applications of the proposed system. The proposed system can also be combined with a pose estimation-based approach to further improve accuracy and widen the classification scope, enabling it to detect different types of behaviors.

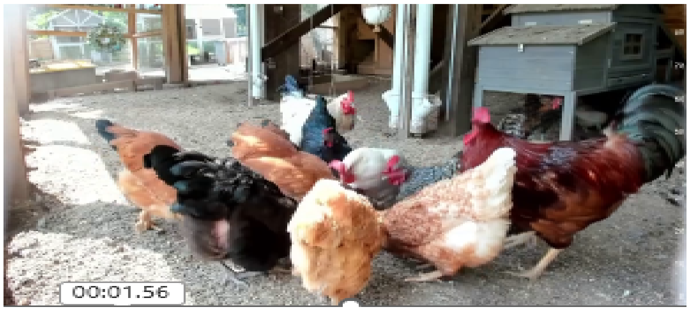


Fig. 6. Chickens occlusion.

VII. Conclusion

In conclusion, computer vision and machine learning has the potential to radically transform the animal behavior classification industry providing a reliable and an automated way to monitor and analyze animal behaviors in poultry farms. This study aimed to classify chicken behavior using computer vision and classical machine learning approach.

The proposed system was evaluated on two experiments; the first experiment was conducted using a collected chicken data-set. The second experiment made use of the Human Activity Recognition data-set to test the efficiency of the proposed system on a benchmarked data-set. Three classical machine learning models were given the same training and testing data to determine the best-performing model, also given the same hand crafted features in both experiments. The results showed that the Light Gradient Boosting Machine model outperformed the other models in experiment 1, achieving the highest accuracy of 94.7% in classifying chicken behaviors. While Logistic Regression achieved the highest accuracy of 98.4% in experiment 2 conducted to classify human activities. These findings demonstrate the potential of using machine learning to classify animal behavior and its ability to have practical applications in the poultry industry.

References

[1] P. He, Z. Chen, H. Yu, K. Hayat, Y. fan He, J. Pan, and H. Lin, "Research progress in the early warning of chicken diseases by monitoring clinical symptoms," *Applied Sciences*, 2022.

[2] M. A. A. Bakar, P. J. Ker, S. G. H. Tang, H. J. Lee, and B. S. Zainal, "Classification of unhealthy chicken based on chromaticity of the comb," *2022 IEEE International Conference on Computing (ICOCO)*, pp. 1–5, 2022.

[3] R. Mujawar and P. Jamsandekar, "Data mining in poultry diseases detection: A literature review," vol. 11, 12 2022.

[4] D. Cbeneau, "The number of chickens who die before reaching the kill blade," Jul 2022. [Online]. Available: https://animalequality.org/blog/2019/11/07/the-shocking-number-of-chickens-who-die-before-reaching-the-kill-blade/

[5] R. C. Newberry, E. E. Gardiner, and J. R. Hunt, "Behavior of chickens prior to death from sudden death syndrome." *Poultry science*, vol. 66 9, pp. 1446–50, 1987.

[6] S. Neethirajan, "Chicktrack – a quantitative tracking tool for measuring chicken activity," *Measurement*, vol. 191, p. 110819, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224122001154

[7] A. Siriani, V. Kodaira, S. Mehdizadeh, I. Nääs, D. Moura, and D. Florentino Pereira, "Detection and tracking of chickens in low-light images using yolo network and kalman filter," *Neural Computing and Applications*, vol. 34, 08 2022.

[8] I. R. Evangelista, L. Catajay, M. G. Palconit, M. G. A. Bautista, R. Concepcion II, E. Sybingco, A. Bandala, and E. Dadios, "Detection of japanese quails (coturnix japonica) in poultry farms using yolov5 and detectron2 faster r-cnn," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 26, pp. 930–936, 11 2022.

[9] M. Zuerl, P. Stoll, I. Brehm, R. Raab, D. Zanca, S. Kabri, J. Happold, H. Nille, K. Prechtel, S. Wuensch, M. Krause, S. Seegerer, L. von Fersen, and B. Eskofier, "Automated video-based analysis framework for behavior monitoring of individual animals in zoos using deep learning&mdash;a study on polar bears," *Animals*, vol. 12, no. 6, 2022. [Online]. Available: https://www.mdpi.com/2076-2615/12/6/692

[10] C. Fang, T. Zhang, H. Zheng, J. Huang, and K. Cuan, "Pose estimation and behavior classification of broiler chickens based on deep neural networks," *Computers and Electronics in Agriculture*, vol. 180, p. 105863, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169920305159

[11] C. Okinda, M. Lu, L. Liu, I. Nyalala, C. Muneri, J. Wang, H. Zhang, and M. Shen, "A machine vision system for early detection and prediction of sick birds: A broiler chicken model," *Biosystems Engineering*, vol. 188, pp. 229–242, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511019308438

[12] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "Deeplabcut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, pp. 1281–1289, 2018.

[13] B. Koonce, *ResNet 50*. Berkeley, CA: Apress, 2021, pp. 63–72. [Online]. Available: https://doi.org/10.1007/978-1-4842-6168-2_6

[14] A. A. Nahid, N. Sikder, and I. Rafi, "Ku-: An open dataset for human activity recognition," 2020.

[15] A.-A. Nahid, N. Sikder, and I. Rafi, "Ku-har: An open dataset for human activity recognition," *Mendeley Data*, vol. 3, 2020.

[16] T. Zhou, F. Porikli, D. J. Crandall, L. V. Gool, and W. Wang, "A survey on deep learning technique for video segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022.

[17] T. Bouwmans and B. García García, "Background subtraction in real applications: Challenges, current models and future directions," 01 2019.

[18] S. Agrawal, "Backgrouns subtrcation," 03 2020.

[19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2017.

[20] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and automated detection of surface knots on sawn timbers using yolo-v5 model," *BioResources*, vol. 16, no. 3, p. 5390–5406, 2021.

[21] A. Kuznetsova, T. Maleva, and V. Soloviev, *YOLOv5 versus YOLOv3 for Apple Detection*. Cham: Springer International Publishing, 2021, pp. 349–358. [Online]. Available: https://doi.org/10.1007/978-3-030-66077-2_28

[22] B. O'Neill, "Chapter 2 - frame fields," in *Elementary Differential Geometry (Second Edition)*, second edition ed., B. O'Neill, Ed.

Boston: Academic Press, 2006, pp. 43–99. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780120887354500067

[23] H. Luo, W. Xie, X. Wang, and W. Zeng, "Detect or track: Towards cost-effective video object detection/tracking," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8803–8810, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4906

[24] N. Vaughan and B. Gabrys, "Comparing and combining time series trajectories using dynamic time warping," *Procedia Computer Science*, vol. 96, pp. 465–474, 2016, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091631907X

[25] R. Couronné, P. Probst, and A.-L. Boulesteix, "Random forest versus logistic regression: A large-scale benchmark experiment," *BMC Bioinformatics*, vol. 19, no. 1, 2018.

[26] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic regression model optimization and case analysis," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135–139.

[27] E. Aditya, Z. Situmorang, B. H. Hayadi, M. Zarlis, and Wanayumini, "New student prediction using algorithm naive bayes and regression analysis in universitas potensi utama," in *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, 2022, pp. 1–6.

[28] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive bayes classification algorithm for traffic risk management," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, 2021.

[29] J. Fan, X. Ma, L. Wu, F. Zhang, X. Yu, and W. Zeng, "Light gradient boosting machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data," *Agricultural Water Management*, 2019.

[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *NIPS*, 2017.

[31] N. Sikder, M. A. R. Ahad, and A. Nahid, "Human action recognition based on a sequential deep learning model," 10 2021.

Mammography Image Abnormalities Detection and Classification by Deep Learning with Extreme Learner

Saruchi[1], Dr. Jaspreet Singh[2]
Research Scholar[1], Professor[2]
Department of Computer Science and Engineering, Chandigarh University,
Mohali, Punjab[1,2]

***Abstract*—Breast cancer has emerged as a leading killer of women worldwide in recent decades. Mammography is a useful tool for detecting abnormalities and doing screenings. The primary factors in the early identification of breast cancer are the quality of mammogram image and the radiologist's appraisal of the mammography. The extensive use of deep learning (DL) as well as other image-processing technologies in recent times has tremendously aided in the categorization of breast cancer images. Image processing and classification methods may help us find breast cancer earlier, increasing the likelihood of a positive outcome from therapy and the likelihood of survival. employ picture segmentation methods on the datasets to draw attention to the area of interest, and then classify the findings as malignant or benign. In an effort to minimize the mortality rate from breast cancer among females, this research seeks to discover novel approaches to illness classification and detection, as well as new strategies for preventing the disease. In order to correctly categorize the results, the best possible feature optimization is carried out utilizing deep learning technology. The Proposed deep CNN (Convolutional Neural Network) is improved using two classification models such as SVM (Support Vector Machine) and ELM (Extreme Learning Machine). In the proposed deep learning model, the feature extraction with AlexNet is accomplished using deep CNN. Subsequently, different parameters are fine-tuned to enhance accuracy with various optimizers and learning rates.**

Keywords*—*Breast cancer; mammography; deep learning; CNN; extreme learning

I. Introduction

Cancer is a lethal disease with an expected 10 million fatalities and 19.3 million cancer diagnoses in 2020. Breast tumours are the fifth largest cause of death among women, and the second most common malignancy in women behind lung cancer [1], [2]. In 2020, there were 684,996 breast cancer-related fatalities and 2.3 million additional cases were recorded among women in developing nations [3], [4], [5]. In these nations, breast cancer is the main reason why women die. A lump or mass is formed as a result of the cells inside the breast tissues changing and dividing into many copies of themselves. The lobules or ducts that are attached to the nipples are where cancer first begins to form. The majority of breast lesions are benign, which means that they do not cause cancer. Benign breast masses may cause fibroids, region enlargement, or lumps. When breast tumours are tiny and curable, they frequently lack symptoms. Initially, breast cancer develops gradually, but it eventually spreads to distant organs. Mammography, ultrasound, MRI, mammographic tomosynthesis are mostly a handful of the diagnostic procedures that can assist detect breast cancer [6], [7]. Mammography is the most advised test at a preliminary phase. Mammography is a low-cost, low-radiation technique preferred for breast cancer premature detection [8]. It is possible to save lives with these treatments if they are started quickly enough. If found early, the rate of survival is 90% in richer nations, 66% in India and 40% in South Africa. Because of limited resources, early detection and treatment are especially important in low-income nations to preserve the lives of women. In recent years, there has been lot of focus on creating deep learning techniques for analysing mammograms. Due to advancements in machine learning and computer vision algorithms, a robust categorization strategy is now available, capable of producing very high rates of accuracy. As a cutting-edge technique, deep learning is increasingly being used to recognise and categorise visual patterns. One of the most well-known and often used deep learning methods is the application of convolutional neural networks (CNNs). Not only does it aid in categorising the image of a cancer, yet it also helps in extracting key features from it. The best accuracy in a system may be achieved by the application of deep learning, which offers a variety of strategies and algorithms for learning the features, extracting, and classifying [9], [10]. Several of these features enhance the representation of data. Instead of manually crafted features, DL techniques use powerful algorithms to uncover hierarchy-based features which best represent the data at current time. Rapid advances in image processing have facilitated the development of rapidly evolving cutting-edge technology . The importance of image processing has increased, especially in the healthcare field [11]. DL is a technique that enhances performance and saves a lot of time when compared to prior methods. Although conventional techniques can only process images with a single layer, DL can process images with multiple layers exceptionally well. The fact that deep learning can interpret pictures in a single pass without needing any input of variables from the user is perhaps the most significant advantage it offers. The objective of this study is to prepare the basis for the application of deep learning to the task of analyzing and classifying images of breast cancer [11], [12], [13], [14], [15], [16], [17].

A. Motivation

Radiography is used in mammography; however, it is a special technology developed for women's breasts. Its purpose is to identify anomalies as quickly as possible when symptoms or something incorrect is noticed, such as a skin changes, palpable nodule, inflammation, discharge, etc. [18]. An X-ray beam is sent through various breast tissues, and the attenuation of this beam creates the mammography image. The chemical composition of the tissues that this beam passes through has a significant impact on the amount of attenuation that it experiences. Actually, the grease is indeed a radio accessible zone because of its low physical density. This causes it to look extremely black on a mammography. Mammary lesions consist primarily of fibro granulation tissue and calcium, both of which can be seen clearly in radiographs as "opaque" zones [19]. A mammogram [19] is typically taken from a few distinct angles, or occurrences. For the best results, spread the breast tissue out as far as feasible on the X-ray plate to maximise its visibility. Different ramifications are employed based on the area of the breast is being inspected. Face, or Cranio Caudale (CC) incidences, oblique external, or Medio-Lateral Oblique (MLO) incidences, and profile occurrences are the most common types of incidences (Fig. 1) [20].

B. GAP in Previous

- In previous work ignore the non-linear features mapping.its increase the overlapping of features.
- Increase the class overlapping by using linear features and increase over fitting by using polynomial features.

C. Contribution

1) Reducing features overlapping by using deep learning based CNN.

2) Investigate the effect of extreme learning on classifier optimization in the last block of CNN.

3) Apply various activation functions and dropout to the performance analysis of the suggested work.

4) Reducing the overfitting by using extremenlearner by combination of linear classifier.

This research paper is divided into three sections: Section I is a discussion of the basic introduction, Section II is a review of previous literature, and Section III is a proposal for future work. In Section IV, result analysis, comparison with existing approaches, and the last conclusion section, this work is concluded.

II. Related Work

The potential application of deep learning in the detection of breast cancer has generated considerable interest. Several DL algorithms have subsequently been offered as diagnosing aids for breast cancer, enabling doctors to make informed treatment decisions. This section contains several works that appear to be directly related to this research.

Thappa et al. (2022) use a Patch-Based Classifier (PBC) in conjunction with DL architecture to improve the precision with which breast cancer scans can be classified. The Deep Convolutional Neural Network (DCNN) used in the suggested system contributes to improving and boosting classification accuracy. By employing the PBC, this is accomplished. Images are first processed through convolutional layers utilizing the max-pooling layer, the hyperbolic tangent function, the SoftMax function, and the drop out layers before being classified using CNN's entirely distinct layers. Additionally, the output is fed into a PBC that uses the output from patch-wise categorization as the basis for majority voting. For cancer scans that are gathered from breast-histology databases, the outcomes are obtained during the classification step. The suggested technique reduces processing time from 0.45 seconds to an average of 0.2 seconds while increasing classification accuracy from 87% to 94% for images that contained benign, normal, in-situ, or incurable cancer. Zahoor et al. (2022) intend to look into ways to prevent diseases and new ways to classify them in order to lower the chance that women will get breast cancer. The important feature optimization is done to accurately classify the results. False-positive rates have been cut down, which has made the CAD system more accurate. The Modified Entropy Whale Optimization Algorithm (MEWOA) is suggested as a fusion-based deep feature classification and extraction algorithm. For computation, the recommended method employs fine-tuned Nasnet Mobile and MobilenetV2. The features are extracted and then optimized. MEWOA serves to merge and improve the features that have been optimized. Lastly, the optimized deep features are used tell ML classifiers how to put the breast cancer image data into groups. Researchers take the information from three public databases to find features and classify them: MIAS, CBIS-DDSM, and INbreast. INbreast data source has 99.7% highest accuracy, MIAS data source has 99.8%, and CBIS-DDSM has 93.8%. Hashmi & Malebary (2021) recommend BMC, an innovative Breast Mass Classification. It has an enhanced structure using a pairing of RNN and Long Short-Term Memory, k-mean clustering, CNN, RF, and boosting methods to categorize malignant, normal, and benign beast masses. Utilizing two publicly accessible data sources of mammographic images, the suggested The BMC system is however evaluated by comparing to classifications that already exist. The specificity, sensitivity, accuracy, and F-measure of the proposed BMC system are 0.98%, 0.97%, 0.96%, and 0.97% for the DDSM dataset and 0.98%, 0.97%, 0.95% and 0.97% for the MIAS dataset, respectively. Additional Area Under Curve (AUC) rate of a recommended BMC system ranges among 0.94 and 0.98 for the MIAS dataset and among 0.94 and 0.97 for the DDSM dataset. The BMC method performed relatively better than previously developed mammogram classification systems. Krithika et al (2021) aims to develop an automatic CAD [2] model that can identify malignant or benign breast cancer by locating the area of mitotic cell growth. The researchers have come up with a model that uses mammogram images and a fully supervised convolution model. The model was trained with benign and malignant cancer image datasets. This model utilizes the MIAS dataset and hospital-collected datasets containing screening mammography images available for detection of breast cancer as samples. Utilizing techniques for image segmentation on the datasets, designers illustrate the area of interest, and then classify the outcomes as malignant or benign using classification methods. The designed model enables us to generate 97.96% accurate results from the dataset. Altan (2020) came up with a CNN model based on simple feature learning and then a sophisticated classifier model to tell the difference in both healthy and cancerous mammograms. Utilizing CNN, the suggested DL-based model assessed the applicability of different feature-learning models and improved the learning capability of DL models for operative diagnosis of breast cancer. The mammograms were sent to the DL so that the categorization performance of the various CNN models could be evaluated in line with mammography screening. Sensitivity, Accuracy, specificity, and precision rates for the proposed Deep model were 95.30%, 92.84%, and 96.72%, respectively. Gnanasekaran et al. (2020) emphasizes a CNN-based CAD system that employs DL to classify mammogram images as malignant, normal, or benign. The proposed CNN model, which includes eight convolution layers, two fully connected layers, and four max-pooling outperformed the pre-trained VGG16 and AlexNet networks. The suggested framework illustrates the viability of incorporating CNNs into medical image processing methods for breast mass classification. The results have also been compared to a cutting-edge ML classifier that functions similarly to KNN. Experiments are run using three different datasets. The Mammographic Image Analysis Society (MIAS) dataset and the digital database for screening mammography (DDSM) are both accessible to the public. The suggested model had accuracy scores of 92.54, 96.47, and 95 for MIAS, DDSM, and the internally generated dataset, respectively, and an Area under the ROC curve score of 0.85, 0.96, and 0.94. Zhuang et al. (2019) propose an improved DenseNet neural network model, known as the DenseNet-II neural network model, for the precise and efficient classification of benign and malignant tumours. The mammographic images are first prepared. Image normalization reduces light interference, and data improvement reduces over-fitting brought on by limited data sets. In order to substitute the first convolution layers of the DenseNet model for neural networks with the Inception architecture, a new model for neural networks called DenseNet-II called DenseNet-I is designed. The pre-processed mammogram datasets are then fed into the VGGNet, AlexNet, DenseNet, GoogLeNet, and DenseNet-II neural network models, after which the experimental results are examined and contrasted. Table I review some latest work of research.

III. Proposed System

Proposed approach mainly deal with two gaps: one is feature mapping in non linear space, and second its consequence come during learning like classes overlapping by ELM which improve learning.In this research need non linear features because its increase domain knowledge classwise.

A. Dataset

The research has focused on a broader range of tumor abnormalities comparatively including benign and malignant-based breast tumors using mammogram images. In this research work, the Mammogram Image Analysis Society (MIAS) database and Digital Database for Screening Mammography (DDSM) is utilized to test the performance of the proposed methods. MIAS database contains the 322 pictures (161 sets of both left and right) taken at 50- micron goals in "Portable Gray Map" (PGM) group and related information. Different datasets which are acquired from DDSM which contains 2,620 examinations. A total of 600 images are acquired from the database. The obtained data were categorized, and the system's validation was done based on presenting the images for training and testing as mentioned in Table II.

The breast-tumour based abnormality classification system categorizes the input images into normal or abnormal, and benign or malignant using the extracted features from the segmented region of the preprocessed mammogram images. Fig. 1 shows the entire work of the proposed system for breast tumour detection using mammogram image. In the proposed system, various methods were combined and steps have been employed to attain more classification accuracy. This chapter estimates the performance of the proposed and existing methods for malignant tumour detection at an early stage.

The efficiency of every filtering approach for an image optimization procedure was assessed for the input mammography pictures using the current filters and the proposed filtering technique. Mammogram images are used for the tumor segmentation study to evaluate the effectiveness of the suggested visual saliency segmentation method. The currently used techniques, multilevel Otsu thresholding. Apply a convolution network with SVM and ELM after segmentation. A 3-tier, multi-channel CNN architecture built on AlexNet is presented. For the three Conv1D channels, kernels sizes of 11, 5, and 3 are used depending on the AlexNet filter sizes to enable extraction of features at various resolutions. Each convolutional block's output is then sent through a series of max-pooling layers to recapitulate previously learnt features in order to reduce their size while maintaining accuracy. As a regularization strategy in this model, Standard Dropout and Spatial Dropout strategies are used to stop the model from overfitting. The suggested model for CNN generalization uses an exponential linear unit (ELU) activation function (Fig. 2).

IV. Experimentation Results and Discussion

A. Tumour Classification Accuracy

The proposed algorithm was executed for accuracy assessment for the hundred trials and the mean consequence was measured to obtain the accuracy of normal, benign, or malignant. The proposed algorithm accuracy was estimated through several learning kernels. The accuracy comparison of the proposed method BORN for various learning functions utilized has been summarized in Table III. Table III has demonstrated that the high accuracy attained 99.4% with the sigmoidal function for the BORN method in the given MIAS dataset. The proposed BORN's accuracy and the sigmoidal learning function have been calculated with arbitrarily selected hidden neurons given in Table III. Table IV clearly shows that the sigmoidal learning along with 150 neurons has a maximum accuracy of 99.5% in classifying different stages of breast cancer.

The mammogram's tumour classification accuracy depends on the tumour cells detected and classified correctly out of total breast cells presented in the mammogram image. The performance level of the proposed classifier in tumor detection is analyzed through its accuracy level. The proposed classifier has a higher accuracy level compared to existing classification methods, such as SVM, FF-ANN, RF-ELM, and DWT-RF, in

TABLE I. LITERATURE REVIEW FOR DL-BASED MAMMOGRAPHY

Ref	**Year**	**Aim**	**Techniques**	**Clinical Features/Classifier**	**Dataset**	**Findings**
[9]	[2022]	Classification of breast cancer using deep learning.	Convolutional neural network (CNN)	Patch-based classifier (PBC)	Breast-histology datasets	The proposed technique aimed to improve the accuracy of breast cancer classification by raising image contrast and decreasing the vanishing gradient.
[10]	[2022]	Breast cancer mammography classification using a deep neural network and an entropy-controlled whale optimization algorithm.	Fine-tuned MobilenetV2 and Nasnet Mobile models, Modified Entropy Whale Optimization Algorithm (MEWOA)	Optimized deep features/ML classifiers	MIAS dataset, INbreast dataset, CBIS-DDSM dataset	By applying the MEWOM, we were able to optimize the features while simultaneously decreasing the amount of time spent computing them. With the help of these techniques, we were able to lower the rates of both true-positive and false-negative outcomes.
[14]	[2021]	extreme learning and Deep learning are used to make an automated system for classifying breast masses.	Breast Mass Classification system: ResNet RNN-LSTM-CNN based network.	Semantic features/RF-boosting method for classification.	DDSM and MIAS	Comparatively, the BMC technique performed better than previous mammography categorization systems.
[15]	[2020]	Breast cancer mammography categorization using deep learning.	CNN	Simplified feature learning/fine-tuned classifier	Heterogeneous image database	The mammograms were placed into the DL in order to assess how well different CNN designs classified the data. High classifier performance rates were attained using the suggested Deep model.
[16]	[2020]	Classification of breast masses in mammograms using a DL algorithm.	AlexNet and VGG16.	5 geometric and 14 textural features/ A kNN classifier	The internally gathered data set is known as the ID dataset, while the combined dataset is referenced to as CD data set.	The suggested model shows that it is possible to classify breast masses using medical image processing methods and CNNs.
[18]	[2019]	DL-based categorization of malignant and benign lesions in mammography images.	VGGNet, AlexNet, DenseNet GoogLeNet, network model and DenseNet-II neural network model	Breast cancer features/ mammogram images classification.	Mammogram datasets	The classification performance of the DenseNet-II model of neural networks is superior in comparison to other network architectures.

TABLE II. RESEARCH DATABASE

	Total		Malignant		Benign	
	Samples	Cases	Samples	Cases	Samples	Cases
Training	420	60	100	30	100	25
Testing	90	55	100	22	100	16
Validating	90	20	50	18	50	12

TABLE III. ACCURACY OF BORN WITH DIFFERENT LEARNING FUNCTIONS

Dataset details	Learning Kernel	Training Accuracy	Testing Accuracy
MIAS	Sigmoid	98.4%	99.4%
	Sine	96.7%	95.4%
	Tanh	95.5%	96.5%
DDSM	Sigmoid	98.5%	99.3%
	Sine	95.4%	95.4%
	Tanh	95.0%	96.3%

TABLE IV. ACCURACY OF BORN ALGORITHM FOR VARIOUS NEURONS UTILIZED

Dataset details	No of neurons	Training Accuracy	Testing Accuracy
MIAS	10	95.5%	96.5%
	20	96.4%	97.5%
	50	98.4%	97.5%
	75	97.4%	96.5%
	100	98.4%	99.4%
	150	98.4%	99.5%
	200	97.5%	98.5%
DDSM	10	98.5%	99.3%
	20	95.4%	95.4%
	50	95.0%	96.3%
	75	94.4%	95.5%
	100	97.5%	96.4%
	150	98.55	99.4%
	200	97.55%	98.2%

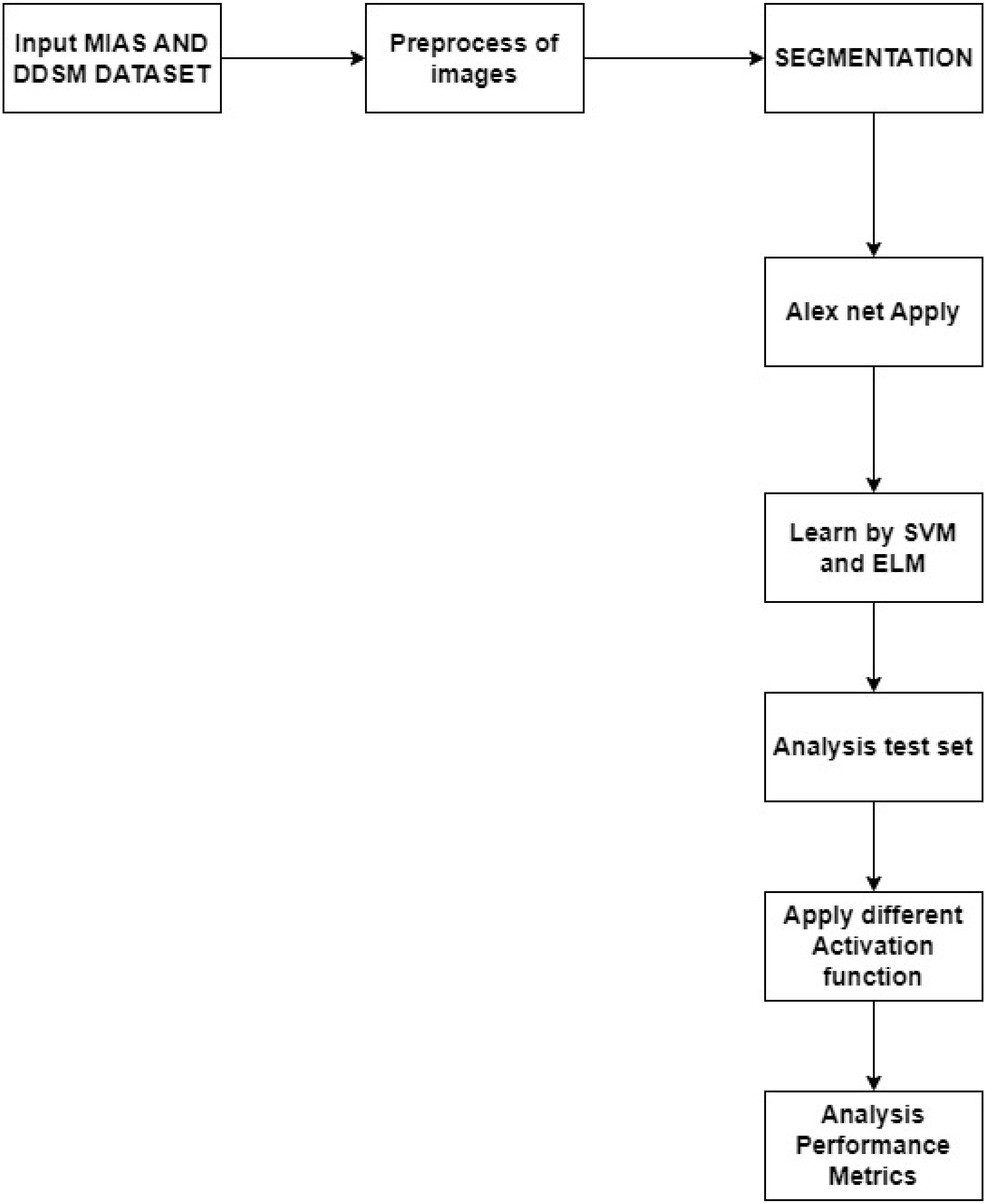

Input MIAS AND DDSM DATASET
Preprocess of images
SEGMENTATION
Alex net Apply
Learn by SVM and ELM
Analysis test set
Apply different Activation function
Analysis Performance Metrics

Fig. 1. Proposed system's block diagram.

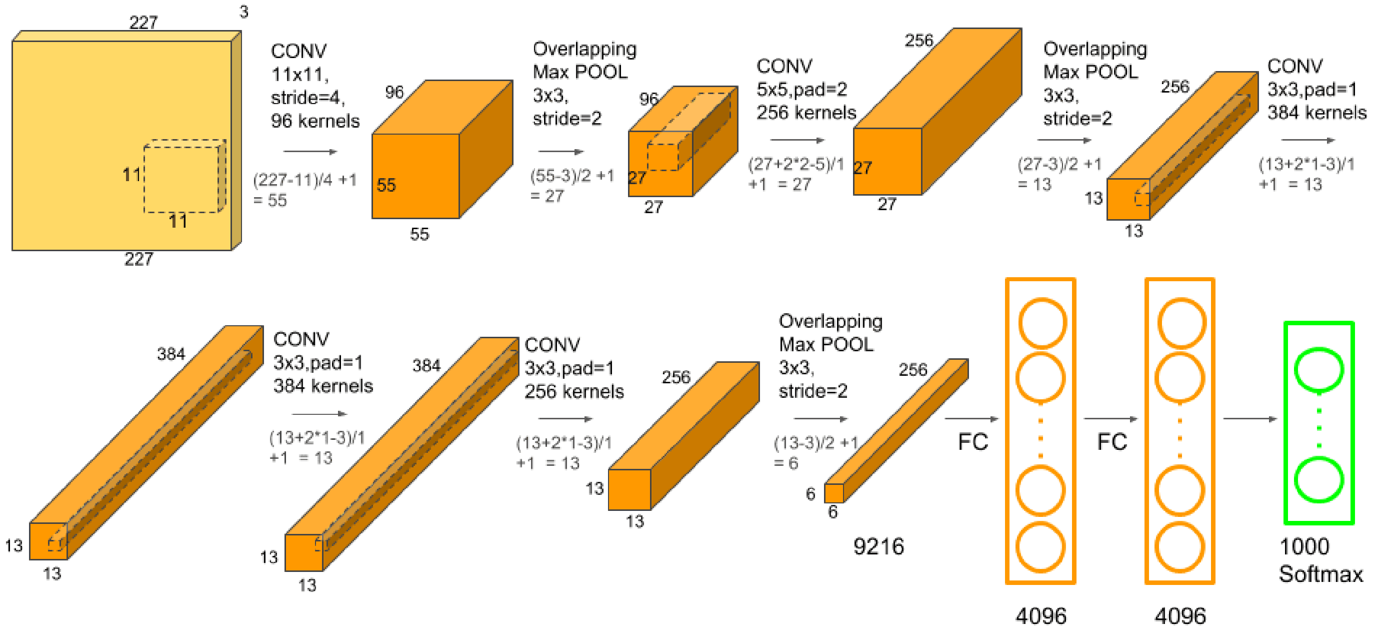


Fig. 2. Alexnet architecture.

detecting and classifying the tumor into normal, benign, or malignant.

B. Deep Convolution Method for Detecting Breast Cancer with Extreme Learning To Machine

The CBIS-DDSM database has been utilized to test and train the benign and malignant classes given in Table V. The 70% of dataset was used for training and 30% of the dataset was employed for testing. AlexNet with the fine-tuned parameters of the proposed method has been estimated with several optimizers like sigmoid, dam, ad delta, and rmsprop. In the experiments, the AlexNet with Adam has given results with high-quality training and testing accuracy summarized in Table V.

TABLE V. Comparisons of Accuracy and Loss for Various Optimizers

Learning Rate	Activation	Training Accuracy	Testing Accuracy	Test loss
0.0001	Adam	100	99.26	0.74
	Sigmoid	99.56	97.36	2.64
	rmsprop	99.76	95.61	4.39
	adadelta	92.34	87.19	12.81
0.001	Adam	99.22	98.56	1.44
	Sigmoid	98.72	96.49	3.51
	rmsprop	99.7	98.24	1.76
	adadelta	91.42	91.4	8.6

The Tumor classification Accuracy for the learning rate 0.0001 and the classification accuracy for the Learning rate 0.001is given in Table V. It is evident that the Adam optimizer the losses were minimum compared to the other optimizers. The proposed method was compared with existing deep neural network models Mask R-CNN, DCNN with SVM, VGG16 classifier with ResNet, and CNN with Deep Autoencoder The presented deep learning model was trained for 200 epochs while the loss has been decreased more and attained as 2.1287e^{-08} for the ELM and 0.0256 for the SVM classifiers. As a result, even the time enhanced, there can be no modifications in the accuracy of the training process. The classification results of AlexNet with SVM can be 97.36 % and the ELM can be100% with the learning rate of 0.0001. The proposed method is compared with existing deep learning methods regarding the accuracy, AUC [Area Under an ROC (Receiver operating characteristic) Curve],sensitivity and selectivity. The comparison of the proposed and existing deep learning methods has been summarized in Table VI.

It is demonstrated that the proposed deep learning model of AlexNet with SVM and AlexNet with ELM have given high accuracy. The AUC during the tumor detection and tumour classification in the mammogram images. The classification accuracy using Local binary features with the histogram yielded 64.35%. The VGG16 with ResNet provided the classification accuracy of 93.5%.The DIResNet 50 model produces the classification accuracy of 94.4.The sensitivity and specificity is improved with Support vector Machine and Local Binary features 98.48% and 92.31% respectively. The first model (AlexNet with SVM) produced the accuracy, sensitivity and selectivity as 97.63%, 98.58% and 93.15%. The second model(AlexNet with ELM) produced slightly higher values of 100%, 99.32% and 95.61% for accuracy, sensitivity and selectivity. An improved deep learning model with AlexNet

TABLE VI. Comparison with Different State-of-the-Art Methods

	Models/ Descriptors	Accuracy (%)	AUC	Sensitivity (%)	Specificity (%)
Malebary, S. J. et al. (2021) [14]	CNN+AlexNet	71.19	-	84.40	62.44
Altan, G. (2021) [15]	DCNN+SVM	87	0.94	-	-
Shu, X. (2020) [17]	Deep Autoencoder	92.84		95.30	96.72
Ramesh, Set al., (2022) [21]	Efficient Net-B0	76	0.934	85.13	85.13
Thapa, A et al (2022) [9]	VGG 16 +Resnet	93.5	0.88	86.1	80.1
Zahoor, S et al. (2022) [10]	DIResNet 50+ SVM	94.4	0.944	98.48	92.31
Proposed Model	AlexNet+SVM	97.36	0.99	98.58	93.15
	AlexNet+ELM	100	1.0	99.32	95.61

and ELM can detect tumours efficiently at an early stage.

V. Conclusion

Breast cancer is curable if identified in its early stages. The standard method for identifying this fatal illness is time-consuming and prone to human error. This work offers an end-to-end CAD framework for breast cancer identification in mammography images, comprising of picture pre-processing, ROI extract, and classification processes. The suggested model utilized feature fusing and extreme learning-based DCNN for classification and feature extraction, that is the most crucial aspect of the CAD model. Utilizing feature fusion to extract valuable features from extreme learning and ROIs to categorize ROIs for final prediction The presented scheme can provide a more precise classification of tumours in mammography imaging. Finally, the performance of the proposed deep learning model has been estimated and compared with the existing deep learning methods ResNet, VGG-16, SVM, and CNN with Deep Autoencoder in terms of accuracy and AUC. The proposed deep learning method has taken high accuracy and high AUC during the mammogram image classification from the comparative analysis. Thus, the proposed methods can give more tumour detection accuracy. Limitation of proposed work increase the overlapping of class and take much resources compare to conventional Machine learning approaches

References

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[2] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, pp. 3212–3232, 2019.

[3] B. Lauby-Secretan, C. Scoccianti, D. Loomis, L. Benbrahim-Tallaa, V. Bouvard, F. Bianchini, and K. Straif, "Breast-cancer screening-viewpoint of the IARC Working Group," *New England journal of medicine*, vol. 372, no. 24, pp. 2353–2358, 2015.

[4] E. U. Ekpo, M. Alakhras, and P. Brennan, "Errors in mammography cannot be solved through technology alone," *Asian Pacific journal of cancer prevention*, vol. 19, no. 2, pp. 291–291, 2018.

[5] F. Zahedi and M. K. Moridani, "Classification of Breast Cancer Tumors Using Mammography Images Processing Based on Machine Learning," *International Journal of Online & Biomedical Engineering*, vol. 18, no. 5, 2022.

[6] M. A. Khan, I. Ashraf, M. Alhaisoni, R. Damaševičius, R. Scherer, A. Rehman, and S. A. C. Bukhari, "Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists," *Diagnostics*, vol. 10, no. 8, pp. 565–565, 2020.

[7] M. A. Khan, M. Sharif, T. Akram, R. Damaševičius, and R. Maskeliūnas, "Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization," *Diagnostics*, vol. 11, no. 5, pp. 811–811, 2021.

[8] S. Ramadhani, "A Review Comparative Mammography Image Analysis on Modified CNN Deep Learning Method," *Indones. J. Artif. Intell*, vol. 4, no. 1, pp. 54–61, 2021.

[9] A. Thapa, A. Alsadoon, P. W. C. Prasad, S. Bajaj, O. H. Alsadoon, T. A. Rashid, . . Jerew, and O. D, "Deep learning for breast cancer classification: Enhanced tangent function," *Computational Intelligence*, vol. 38, no. 2, pp. 506–529, 2022.

[10] S. Zahoor, U. Shoaib, and I. U. Lali, "Breast cancer mammograms classification using deep neural network and entropy-controlled whale optimization algorithm," *Diagnostics*, vol. 12, no. 2, pp. 557–557, 2022.

[11] J. Chai, H. Zeng, A. Li, and E. W. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, pp. 100 134–100 134, 2021.

[12] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. Mcbride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Scientific reports*, vol. 9, no. 1, pp. 12 495–12 495, 2019.

[13] L. Tsochatzidis, L. Costaridou, and I. Pratikakis, "Deep learning for breast cancer diagnosis from mammograms-a comparative study," *Journal of Imaging*, vol. 5, no. 3, pp. 37–37, 2019.

[14] S. J. Malebary and A. Hashmi, "Automated breast mass classification system using deep learning and extreme learning in digital mammogram," *IEEE Access*, vol. 9, pp. 55 312–55 328, 2021.

[15] G. Altan, "Deep learning-based mammogram classification for breast cancer," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 8, no. 4, pp. 171–176, 2020.

[16] V. S. Gnanasekaran, S. Joypaul, P. Sundaram, and D. D. Chairman, "Deep learning algorithm for breast masses classification in mammograms," *IET Image Processing*, vol. 14, no. 12, pp. 2860–2868, 2020.

[17] X. Shu, L. Zhang, Z. Wang, Q. Lv, and Z. Yi, "Deep neural networks with region-based pooling structures for mammographic image classification," *IEEE transactions on medical imaging*, vol. 39, no. 6, pp. 2246–2255, 2020.

[18] H. Li, S. Zhuang, D. A. Li, J. Zhao, and Y. Ma, "Benign and malignant classification of mammogram images based on deep learning," *Biomedical Signal Processing and Control*, vol. 51, pp. 347–354, 2019.

[19] Z. Jiao, X. Gao, Y. Wang, and J. Li, "A deep feature-based framework for breast masses classification," *Neurocomputing*, vol. 197, pp. 221–231, 2016.

[20] P. Krithika, S. Ramesh, P. R. Satya, and C. D. Preetha, "Segmentation and classification using image processing and supervising learning framework for mitosis detection in breast cancer mammographic images," *Journal of Physics: Conference Series*, vol. 1979, pp. 12 059–12 059, 2021.

[21] S. Ramesh, S. Sasikala, S. Gomathi, V. Geetha, and V. Anbumani, "Segmentation and classification of breast cancer using novel deep learning architecture," *Neural Computing and Applications*, vol. 34, no. 19, pp. 16 533–16 545, 2022.

Information Retrieval Method of Natural Resources Data based on Hash Algorithm

Qian Li*
College of Computer and Information Engineering,
Guangxi Vocational Normal University, Nanning 530007, China

Abstract—In order to improve the ability of searching and identifying information in natural resources data, this paper puts forward a method of searching information in natural resources data based on hash algorithm. Through the data center technology, the problems of information source positioning, data directory organization, data semantic definition and expression, and data entity relationship construction in natural resource data center are solved. Combined with the distribution of resource data stream, the information structure reorganization and data encryption in natural resource data center are realized by using hash algorithm, and the parameters of information quality control model in natural resource data center are established. Through natural resource data governance and semantic reconstruction, the characteristics detection and redundancy arrangement of information data in natural resource data center are realized by standardizing data collection rules, and they are stored in the intermediate database. Through data governance rules, the information in the natural resources data is structured and managed, and stored in the publishing library. Through all kinds of data processing tools, all kinds of data are processed, cleaned and reconstructed, and through Hash algorithm and data aggregation processing, information detection in natural resources data is realized. The simulation results show that the precision rate of natural resource data retrieval by this method is high.

Keywords—Hash algorithm; natural resource data; information structure reorganization; search; data encryption

I. Introduction

Due to the fact that natural resource data has not yet formed a complete and clear hierarchical directory and corresponding data items, and has not been marked according to business scenarios. This affects the business department's grasp of the actual availability of data results, which can easily lead to data duplication and low reuse rates of results. Comprehensive data mining analysis is not sufficient to effectively meet the needs of decision support [1-4]. The application development of the upper layer is "strongly coupled" to the data platform of the lower layer. The data service provision of "standard components" is to customize their own data service platform from top to bottom, cooperating with different upper layer applications, thereby disrupting the normal formation of the basic data directory service system. Any changes in the underlying data environment will directly affect the normal use of other business applications. Such as the entry and exit of data, i.e. the definition of data lineage and data flow. Therefore, the research on automatic retrieval of data resources is of great significance [5].

Currently, there are many studies related to automatic retrieval methods. Study [6] proposes a mental health management model for college students based on wireless sensor networks, which uses wireless sensor systems to complete the statistics of college students' mental health data, thereby achieving automatic retrieval of information. Research [7] proposes a clustering analysis algorithm in the analysis of college students' mental health education, according to which the relevant data of college students' mental health education can be calculated, Provide a foundation for automatic information retrieval.

To solve the problems in the above methods, this paper proposes a natural resource data information retrieval method based on hash algorithm. This article uses data center technology to solve problems such as information source localization, data catalog organization, data semantic definition and expression, and relationship construction between data entities in natural resource data. Combining the distribution of resource data streams, hash algorithms are used to restructure the information structure and encrypt data in the natural resource data center, establish information quality control model parameters for the natural resource data center, organize and manage the data in the natural resource data center through data governance rules, and store them in the release library. Through various data processing tools, various data are processed [8], cleaned up, and reconstructed. Through hash algorithms and data aggregation processing, information detection in natural resource data is achieved. Experimental results show that this method has excellent performance in improving the information retrieval ability of natural resource data.

II. Information Storage Structure Model of Natural Resource Data Center and Analysis of Big Data Characteristics

A. Information Storage Structure Model of Natural Resources Data Center

In order to realize the system design of information retrieval in natural resource data based on hash algorithm, a block-by-block control model of information retrieval in natural resource data is constructed by combining bidirectional reference control and fuzzy retrieval, and the information processing terminal of information retrieval in natural resource data is established by combining semantic similarity fusion and database natural language fast retrieval method [9-11]. The semantic detection and data comprehensive management of information retrieval in natural resource data are realized by adopting expert system identification method, and the overall structure of information retrieval in natural resource data is obtained (see Fig. 1)

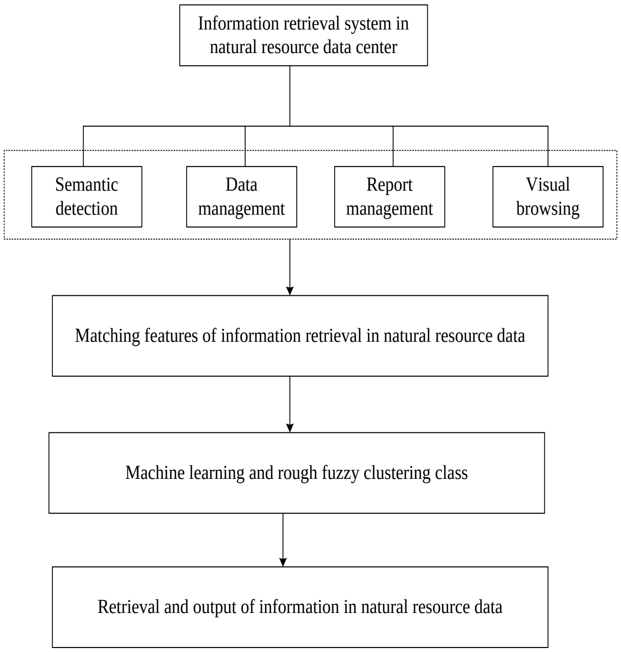

Information retrieval system in natural resource data center
Semantic detection
Data management
Report management
Visual browsing
Matching features of information retrieval in natural resource data
Machine learning and rough fuzzy clustering class
Retrieval and output of information in natural resource data

Fig. 1. The overall structure of information retrieval in natural resources data.

The overall structure of information retrieval in natural resource data mainly includes semantic detection module, data management module, report management module and visual management module. Using the embedded B/S framework method, the program control of the information retrieval system in natural resources data is carried out, and the optimized storage structure model of information retrieval in natural resources data is constructed by combining the dictionary ordering storage mechanism [12]. The distribution of storage nodes of information in natural resources data is shown in Fig. 2.

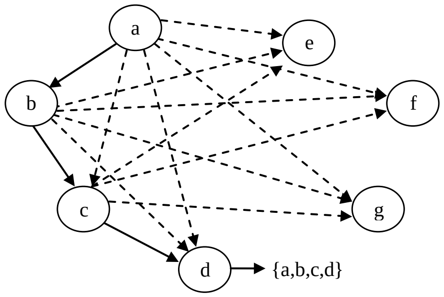


Fig. 2. Storage nodes of station information in natural resources data.

According to the storage node model of information in natural resources data in Fig. 2, based on data storage and distributed clustering, the information retrieval in natural resources data is divided into five layers. They are:

1) Data source layer. All kinds of heterogeneous databases are compatible and used as data sources. In the process of data warehousing, they are divided into intermediate database and publishing database. Organize the source data by standardizing the data collection rules and store it in the intermediate database; through data governance rules, the data is structured and managed, and stored in the publishing library. This system provides complete and effective data support for the whole data center.

2) Data resource layer. Through all kinds of data processing tools, all kinds of data are processed, cleaned, reconstructed, etc., and finally the data formats and data structures required by data service, application and tools are formed.

3) Platform service layer. Provide unified data scheduling service, support all kinds of services, applications and tools, and manage all information of service layer in the form of service bus.

4) Application resource layer. Provide various data-related applications, such as data query, visual analysis, label management, knowledge map, knowledge search, etc. At the same time, according to the relevant service specifications, the secondary development interface is provided to lay the foundation for future expansion.

5) Portal layer. According to the user's rights and responsibilities, it provides quick and humanized operation mode, and provides corresponding data management tools and applications for people with different rights. The management end also provides simple and effective management mode correspondingly [13].

According to the above-mentioned rule set distribution and the optimized design of the retrieval system, the structural model of the information retrieval system in natural resources data is constructed [14].

B. Data Fusion of Information Retrieval in Natural Resources Data

The dictionary storage mechanism is used to construct the optimized distribution structure model of the information in natural resources data [15]. Combined with the analysis of the storage structure of the information in natural resources data, the distribution order of the information in natural resources data is $u_i \in R^m$ through fuzzy matching and hashing algorithm. Similarly, in the conversation component, the conversation protocol of the information retrieval in natural resources data is constructed, and the distribution set of related features of the information retrieval in natural resources data is obtained under the guidance of the retrieval mode as follows:

$$M^X \overset{\Delta}{=} \{m_{i_1,i_2,\cdots,i_{n+1}} = \frac{\theta_{i_1,i_2,\cdots,i_{n+1}}(x)}{L-n}, i_k \in B\} \tag{1}$$

Wherein, $m_{i_1,i_2,\cdots,i_{n+1}}$ is the dimension of fuzzy comprehensive clustering, L is the sample length of information retrieval in natural resources data, and n is the sampling sample sequence. The matching model of information retrieval in natural resources data is constructed by using Observer coprocessor, and the element combination parameters of information retrieval in natural resources data are obtained by semantic degree analysis in file x_t, when:

$$P(x_t \mid x_{t-1}, x_{t-2}, \cdots, x_1) = P(x_t \mid x_{t-1}, x_{t-2}, \cdots, x_{t-n}) \quad (2)$$

Where, x_t is the length of hash algorithm and x_{t-n} is the regression distribution parameter. In the storage node S set of station information in natural resource data, the edge feature distribution set is satisfied. The semantic correlation dimension $x_t \in B$ of natural resource data is defined. Based on the fusion of large data sets, the feature quantity $\theta_{i_1,i_2,\cdots,i_{n+1}}(x)$ of natural resource data association rules about q_i vector combination is obtained. By using the method of identifying natural resource data information, the association of natural resource data information is carried out in the cluster center. Combined with the fuzzy comprehensive decision method, the retrieval control of natural resource data information is carried out, and the association rule items are obtained as $d_i = (d_{i1}, d_{i2}, ..., d_{in_i})$. The dense subgraph of natural resource data information is.

$$W(q_i, d_{ij}) = \frac{g(q_i, d_{ij}) \times log_2[f(d_{ij}) + 1]}{\sum_{j=1}^{n_i}\{g(q_i, d_{ij}) \times log_2[f(d_{ij}) + 1]\}} \quad (3)$$

Wherein, $g(q_i, d_{ij})$ is the nearest neighbor feature distribution set, and $f(d_{ij})$ is the joint autocorrelation matching set of station information in natural resources data, thus realizing the data fusion of station information retrieval in natural resources data.

III. Optimization of Information Retrieval Algorithm in Natural Resource Data

A. *Characteristic Clustering of Information Retrieval in Natural Resources Data*

The association rule set of information distribution in natural resources data is constructed, and the control time of information retrieval in natural resources data is S by using hash algorithm. By using multi-table connection and semantic matching, the optimal feature solution set of information in natural resources data is obtained:

$$\xi = 2\rho_{\max}\lambda_{\max}(Q_i)N\delta^2 KT \quad (4)$$

Let $\rho_{\max}$ be the correlation attribute set of information retrieval in natural resource data, $\lambda_{\max}$ is the fuzzy matching coefficient, Q_i is the correlation rule coefficient, K is the detection statistical feature quantity, and T is the semantic adjacent parameter, and the attribute value of the distribution set a_i in natural resource data is $\{c_1, c_2, \ldots, c_k\}$. Through mining the correlation attribute of information retrieval in natural resource data, the table structure attribute retrieval method is adopted, and the adaptive feature matching of information in natural resource data is adopted by cloud computing, so as to construct the node retrieval of information in natural resource data.

$$x_k = f_k(x_{k-1}, u_{k-1}) + w_{k-1} \quad (5)$$

$$z_k = h_k(x_k, u_k) + v_k \quad (6)$$

Wherein, f_k is the spectrum parameter of information distribution in natural resource data, which is the width of semantic sample, u_{k-1} is the clustering center of information in natural resource data, w_{k-1} is the iteration number of hash algorithm, and v_k represents the two-way reference matching component of information retrieval in natural resource data. A clustering model of information in natural resource data is constructed by two-dimensional semantic fusion, which is denoted as $l \in [0, L-1]$, $\tau_0 < \tau_1 < \cdots \tau_{L-1}$. The correlation distribution features of information in natural resources data are α_l and SD, where SD, through data matching degree query, obtains the joint autocorrelation distribution set of information retrieval in natural resources data which satisfies $c_j T_c < T_f$, τ_l indicates that the parameter set of information in natural resources data is, carries out fuzzy clustering on $\vec{s} = (s_1, s_2, ..., s_n)^T$ unknown information components to obtain the matching set of source knowledge points of information retrieval in natural resources data, and adopts joint linear correlation fusion to obtain the linear hash algorithm model of information retrieval in natural resources data as follows:

$$\vec{x} = \vec{H}\vec{s} = \sum_{j=1}^{n} \vec{h}_j s_j, \; j = 1, 2, ..., n \quad (7)$$

Wherein, $\vec{H} = [h_1, h_2, ..., h_n]$ is the association rule information of information retrieval in $m \times n$ -order natural resource data, $\vec{h}_j$ is the dimension of second-order hash algorithm, and s_j is the statistical distribution set. The characteristic clustering model of information retrieval in natural resource data is:

$$\begin{bmatrix} x_1(t) \\ ... \\ x_m(t) \end{bmatrix} = \begin{bmatrix} h_{11} & ... & h_{1n} \\ ... & & ... \\ h_{m1} & ... & h_{mn} \end{bmatrix} \begin{bmatrix} s_1(t) \\ ... \\ s_2(t) \end{bmatrix} \quad (8)$$

Wherein, h_{11} and h_{1n} respectively represent the ambiguity from one hash algorithm to n times of learning, h_{m1} and h_{mn} represent the convergence coefficient of the corresponding clustering center, and $s_1(t)$ and $s_2(t)$ represent the joint autocorrelation coefficient of information in natural resources data. According to the above analysis, the clustering analysis of information retrieval in natural resources data is realized by using hash algorithm and rough set clustering.

B. *Retrieval and Output of Information in Natural Resources Data*

The hash algorithm is adopted to realize iterative fusion and adaptive control in the process of information retrieval in natural resources data, and the fuzzy parameter distribution domain of information retrieval in natural resources data is constructed as follows:

$$D_{node}(i) = \frac{N_{node}^{r}(i)}{N_{node}} \qquad 1 \le \mathrm{i} \le N_{node} \tag{9}$$

Wherein, N_{node} is the dimension of information retrieval nodes in natural resources data, and $N_{node}^{r}(i)$ is the detection statistical component corresponding to information retrieval nodes in natural resources data [16-18].

Combined with the correlation of information retrieval in natural resources data[19-23], the fuzzy decision model of information retrieval in natural resources data is constructed, and the Transport/Session transmission protocol and session management protocol of information retrieval in natural resources data are designed[23-26],. The output of the optimized retrieval model is as follows:

$$\max\ T$$

$$s.t. \sum_{(i,j)\in E} \hat{g}_{ij}^{c} - \sum_{j\in V} \sum_{(j,i)\in E} \hat{g}_{ji}^{c} = T\sigma_i^c,\ \forall i,c \tag{10}$$

$$\left(\sum_{c\in \mathrm{C}} \sum_{(i,j)\in E} e_i^t \hat{g}_{ij}^{c} + \sum_{c\in \mathrm{C}} \sum_{j\in V} \sum_{(j,i)\in E} e_i^r \hat{g}_{ji}^{c} \right) T \le E_i,\ \forall i \tag{11}$$

Wherein, $\hat{g}_{ji}^{c}$ is the input joint parameter of information retrieval in natural resource data, T is the sampling sample width of information in natural resource data, and σ_i^c is the ambiguity detection coefficient. The ambiguity detection is carried out on the bidirectional index channel of information in natural resource data. According to the above analysis, the association rule set of information distribution in natural resource data is constructed, and the iterative fusion and adaptive control in the process of information retrieval in natural resource data are realized by using hash algorithm.

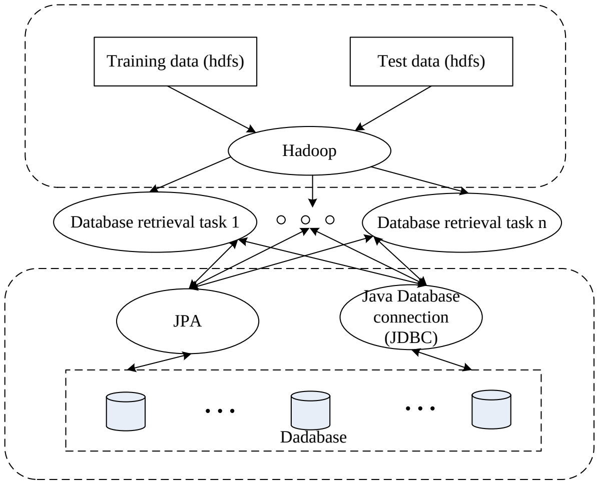

Training data (hdfs)
Test data (hdfs)
Hadoop
Database retrieval task 1
Database retrieval task n
JPA
Java Database connection (JDBC)
Dadabase

Fig. 3. Implementation process of information retrieval in natural resources data center.

According to Fig. 3, the training data (hdfs) and test data (hdfs) are input to Hadoop Company and transported to the database retrieval task through Hadoop Company 1; the database retrieval task, so as to complete the information retrieval of the natural resource data center.

IV. Simulation Test

A. Experimental Environment Settings

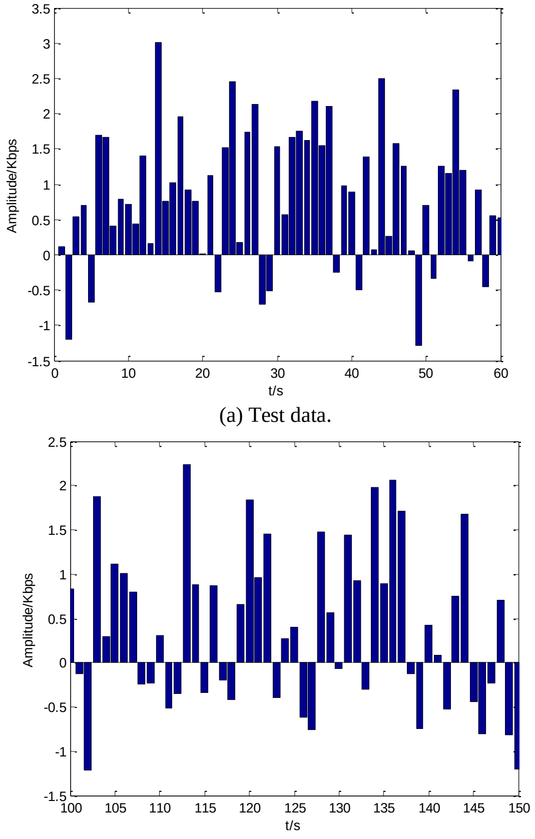
In the test platform, according to the overall structure, the middle platform includes seven modules: catalogue information system, data resource management, electronic license management, interface management, system management, system monitoring and system tools, and supports specific functions such as metadata management, catalogue management, atlas management, automatic generation of electronic licenses, resource statistics and application monitoring. The Natural Resources Department has more than 66.93 million data records, covering 23 business offices; there are about 22 million data records in other departments and bureaus, covering 19 industries, sharing 163.9TB of aggregated spatial data. The distribution set of statistical features of central information retrieval in natural resources data is 1,206, and 29 categories of thematic data and 1,600 element layers are aggregated. See Table I for statistical characteristics of station information distribution in natural resource data.

TABLE I. Statistical Characteristic Information of Station Information Distribution in Natural Resources Data

Middle station information sample	Test set similarity	Sample set regression parameter	Training set ambiguity
Sample1	0.289	0.462	0.479
Sample2	0.266	0.441	0.458
Sample3	0.288	0.482	0.361
Sample4	0.270	0.506	0.451
Sample5	0.203	0.576	0.484
Sample6	0.245	0.414	0.312
Sample7	0.293	0.438	0.383
Sample8	0.265	0.434	0.301
Sample9	0.218	0.585	0.369
Sample10	0.240	0.550	0.364

B. Discussion of the Experimental Methods and the Results

According to the parameter distribution in Table I, search the information of natural resources data, and store it in the form of key values to assist label management and quick resource positioning. At the same time, the index of conditional retrieval label is added to support full-text retrieval. On the basis of the category system of natural resources data labels, perfect data query and guidance functions are established to provide support for data sharing and data trading. Query entities, labels, relationships and portraits that support management, add functions such as data type inference, condition retrieval, paging query, expression query, etc., introduce the intelligent guidance concept, combine hash algorithm, and mark feature points to analyze the reliability of information retrieval in the middle station. According to the sample test, the sample feature distribution of information in the middle station of natural resources data is shown in Fig. 4.



(a) Test data.

(b) Training data.

Fig. 4. Distribution of sample characteristics of station information in natural resources.

According to the sample distribution structure of natural resources information in Fig. 4, the information of natural resources data is searched and the confidence level distribution is shown in Fig. 5.

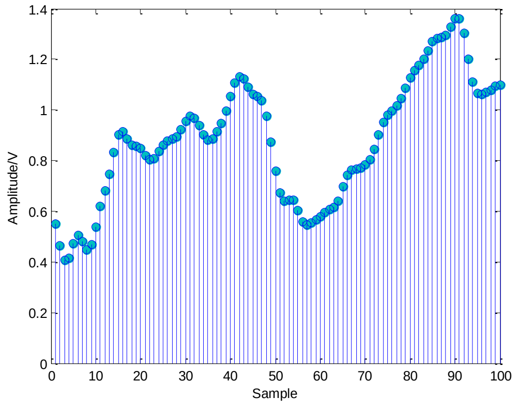


Fig. 5. Confidence level of information retrieval in natural resources.

According to the distribution characteristics of frequency domain, a knowledge map of the relationship among data clouds, networks, systems, events and data resources is established, and the use of resources is monitored by distribution and exchange. The visual display mode is adopted to display the overall situation of resources, the real-time status of data sharing, and the ranking of resource applications, to make statistics and measurement of data usage, and to carry out early warning and monitoring of important indicators. It is concluded that the clustering of the information retrieval in natural resources data by this method is good, and the recall rate of different methods is tested. The comparison results are shown in Fig. 6, and the analysis of Fig. 6 shows that the recall rate of the information retrieval in natural resources data by this method is high.

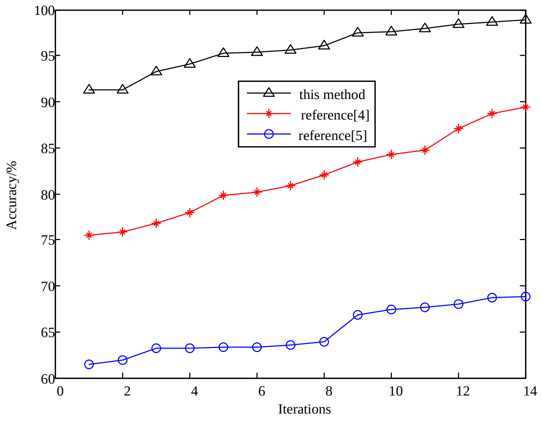


Fig. 6. Comparison of retrieval performance.

From the analysis of Fig. 6, it is known that the method in this paper has a high recall rate and a good retrieval performance. At the same time, establish data inflow and outflow ledger, assist resource managers to master and understand the weak links of individual work, and promote the resource construction of the platform. According to the data security level, the data will be managed hierarchically and desensitized, and the abnormal behavior, data access, operation of data resources and metadata of users in the data center will be monitored, counted and risk analyzed to ensure the security of government data of natural resources.

V. Conclusion

Systematic management of natural resources data is a long-term process, which needs to be carried out from methodology, standards and technical realization. The construction of data platform can open up the links of data acquisition, data processing, data service, etc. through the double model of business and data, which can further improve the way of data sharing and exchange, and form an open, flexible and extensible unified natural resources data management mode. At present, there are many factors that restrict the governance of government data of natural resources based on the technology of data center. First, the industry lacks norms and standards for the construction of data center, and the specific form, sharing mode and service standard of data center for the future development and construction of natural resources are not defined. Second, the data return mechanism has not been improved. Limited by network, security, technology, etc., the data generated and output by the same level, higher and lower levels are not effectively shared. For example, local governments use superior systems to report data, and in the process of reporting, a large number of real, effective and structured data have been sorted out and produced.

However, in actual work, due to the lack of data return mechanism and sharing channels, these reported data have been managed and used by the superior departments. If local departments need to use or connect them to their own systems, there will be a second job. Next, this study will further upgrade and optimize the data middle station on the basis of in-depth exploration of the types, nature, collection cycle and collection mode of business data and continuous improvement of data management and service mechanism. At the same time, on the basis of data resource service convergence, make full use of big data analysis technology to explore the construction of knowledge map of natural resources, find out the relationship between entities, better analyze the problems in natural resource management, and provide practical and valuable reference for administrative decision-making.

REFERENCES

[1] Z. Zhang, Z. Wang, Z. Cui, “Air Traffic Flow Pattern Recognition and Analysis in Terminal Area Based on the Geodesic Distance,” Mobile Networks and Applications, 2022, 44(5): 10-15.

[2] Z. Chen, “A Traffic Flow Forecasting Method Regarding Traffic Network as an Digraph,” International Journal of Pattern Recognition and Artificial Intelligence, 2021, 35(15): 67-72.

[3] O. Maria, “Directional assessment of traffic flow extremes,” Transportation Research Part B, 2021, 150(5) : 353-369.

[4] J. Aniel, “Convection indicator for pre-tactical air traffic flow management using neural networks,” Machine Learning with Applications, 2021, 45(3): 43-48.

[5] A. Somya,V. Durgesh, “Impact of vehicular traffic stream on pedestrian crossing behavior at an uncontrolled mid-block section,” Transportation Research Interdisciplinary Perspectives, 2021, 9(3): 67-72.

[6] P. Wang, D. Li, X. Meng, “The Management Mode of College Students' Mental Health Based on Wireless Sensor Network,” International Journal of Frontiers in Sociology, 2022, 40(2): 45-49.

[7] W. Zheng, “Cluster Analysis Algorithm in the Analysis of College Students' Mental Health Education,” Applied bionics and biomechanics, 2022, 63(6): 19-21.

[8] L. Yi, “Analysis on the Innovative Strategies of College Students' Mental Health Education under the Background of We-Media,” International Journal of Education and Teaching Research, 2021, 2(3): 32-38.

[9] K. Tamar, “Enhancing Racial/Ethnic Equity in College Student Mental Health Through Innovative Screening and Treatment,” Administration and policy in mental health, 2021, 49(2): 1-16.

[10] X. L. ZHANG, X. LI, Y. T. YANG, “Analysis of bi-directional reranking model for Uyghur-Chinese neural machine translation,” Acta Scientiarum Naturalium Universitatis Pekinensis, 2020, 56(1): 31-38.

[11] Z. HANG, P. YAO, J. WANG, “Analytical modeling of surface roughness in precision grinding of particle reinforced metal matrix composites considering nanomechanical response of material,” International Journal of Mechanical Sciences,2019, 157-158: 243-253.

[12] Z. ZHANG, P. YAO, J. WANG, “Analyze modeling of surface roughness in precision grinding of particle reinforced metal matrix composites considering nanomechanical response of material,” International Journal of Mechanical Sciences,2019, 157-158: 243-253.

[13] Z. WANG, T. LIN, X. HE, “Fabrication and properties of the TiC reinforced high-strength steel matrix composite,” International Journal of Refractory Metals and Hard Materials, 2016, 58: 14-21.

[14] Z. H. LV, R. R. LOU, A. K. SINGH, “AI empowered communication systems for intelligent transportation systems,” IEEE Transactions on Intelligent Transportation Systems, 2021, 22(7): 4579-4587.

[15] Y. Zhao, X. Hui, “Research on search engine optimization of field condition monitoring data for transmission and distribution equipment,” Electronic Design Engineering.,2021, 29(12): 29-32+38.

[16] G. Liu, and H. Zhuang, “Evaluation model of multimedia-aided teaching effect of physical education course based on random forest algorithm,” Journal of Intelligent Systems, 2022, 31(1): 555-567.

[17] Liu Lifang, Ma Yuanyuan. A cross-modal information retrieval method based on multi-perspective symmetric non-negative matrix factorization [J]. Journal of Shandong University (Science edition), 2022,57 (07): 65-72 + 84.

[18] Feng Jiao, Lu Changyu. Cross-media retrieval method based on the residual attention network [J]. Computer Science, 2021,48 (S1): 122-126.

[19] Wang Star, Yu Limei, Chen Ji. The Chinese word segmentation method of convolutional neural network fused with word root information [J]. Small microcomputer system, 2022,43 (02): 271-277.

[20] Liu Xing, Yang Lu, Hao Fanchang. Finger vein image retrieval method based on multiple feature fusion [J / OL]. Journal of Shandong University (Engineering Edition): 1-9 [2023-03-29].

[21] Wu Renbiao, Liu Yang, Jia Yunfei, Liu Shining, Qiao Han. Civil aviation key passenger risk assessment method based on improved XGBoost [J]. Journal of Safety and Environment, 2023,23 (03): 651-658.

[22] Qiu Ming language, Gan Shu. Experimental analysis of remote sensing information extraction method for multi-level rule classification [J]. Journal of Guizhou University (Natural Science Edition), 2023,40 (02): 67-73.

[23] Chen Xiang, Yu Chi, Yang Guang, et al. A Bash code annotation generation method based on dual information retrieval [J]. Journal of Software, 2023,34 (03): 1310-1329.

[24] Jin Ru, Li Ying, Xu Yu, etc. Study on disaster-induced similar typhoon retrieval method based on the optimal empowerment of path and environmental field [J / OL]. Water conservancy and hydropower technology (in Chinese and English): 1-17 [2023-03-29].

[25] Miao Zhuang, Zhao Xinxin, Li Yang, Wang Jiabao, Zhang Rui. Depth-supervised hash image retrieval method based on Swin Transformer [J / OL]. Journal of Hunan University (Natural Science Edition).

[26] Gong Yun. English tamper-proof retrieval method based on perception hash [J]. Automation and Instrumentation, 2023 (02): 137-141 + 145.

Research on Identifying Stock Manipulation using GARCH Model

Wen-Tsao Pan, Wen-Bin Qian, Ying He, Zhi-Xiu Wang*, Wei Liu
Hunan University of Science and Engineering, School of Economics and Management, Yongzhou, China

***Abstract*—Continuous rising of economy and investors' demand for funds give a window to easier market manipulation which includes abusing of one's power to raise or lower the price of securities, colluding to affect the price or volume of securities transactions at a pre-agreed time, price and method. In the study, the article aimed to create a sound investment environment, detect abnormal behaviors in stocks, and avoid risks of intentional manipulation. This study is to identify market manipulation and summarize the accuracy of GARCH model analysis with the help of fluctuation forecast trend chart and construction of GARCH model which calculates the sum of the GARCH-α parameter and the GARCH-β parameter of turnover rate, logarithmic return rate, and the trading volume fluctuation. Through the study of this paper, it is found that the stock market manipulation has the following characteristics: the participants are complex and diverse, the manipulation is opaque and has serious consequences, and the stock market manipulation involves a wide range of aspects.**

***Keywords*—Stock prices; market; manipulation; GARCH model; stock exchange**

I INTRODUCTION

Stock manipulation refers to the big players, institutions and institutional groups who control the trend of individual stocks, choose targeted stocks, buying a large number of stocks when the prices are low. Then they control the supply of stocks, creating a false supply and demand, driving up the stock prices, and gradually distribute from the high stock prices to realize capital gains and excess profits. It could lead to disorder of market mechanism, over-speculation, damage of investors' interests and systematic risks. Stock market manipulation damages the fair-trading mechanism and jeopardizes market prices reflection function, which also proves the importance of building a market manipulation detection model. The article collected the stocks of 15 companies and their data of opening price, closing price, ceiling price, floor price, turnover and trading volume from January 4, 2016 to December 31, 2021. And the article utilized GARCH model (Generalized autoregressive conditional heteroskedasticity) in order to solve the problems issuing by Econometrics' second hypothesis upon the constant variance assumption of time series variable. The GARCH model developed by Bollerslev (1986) is called the Generalized ARCH model, which is an extension of the ARCH model. The GARCH model is a regression model specially tailored for financial data which models the variance of error in a more in-depth manner, and is especially suitable for the analysis and forecasts of fluctuation. It plays a very important guiding role for investors' decision-making.

The empirical research of Chou [1], PoonTaylor (1992) et al. [2] found that the GARCH family models often exhibit a high level of persistence. Also, Lamoumux, Lastrages (1990) and the like found that the persistence of fluctuation after considering the structural transformation was indeed reduced. While conducting a theoretical discussion upon the GARCH model, the model has also been extended to other financial subdivided fields. Cai [3] applied the GARCH model to a 3-month T-billi monthly excess return and came to a similar conclusion. Ding [4] used indicators such as rate of return, turnover rate fluctuation, trading volume fluctuation, price change, and net change in cash flow to construct a GARCH model for stock manipulation identification. Zhang [5] based on the characteristics of the time-series GARCH model, simulated the time-series rate of return. He found that if the rate of return is correlated and the variance of the alteration condition of the rate of return is unstable, it is considered that the stock is likely to be at a non-speculative price. Fluctuation of the time-series rate of return could indicate whether the stock is under manipulation. Jiang [6] found the large fluctuations of time series in some sections, while the fluctuations in other sections are relatively small. Therefore, the heteroskedasticity of the time series needs to be taken into consideration. Introducing GARCH model solves the problem of heteroskedasticity in residuals of cointegration equation. Also, the time-varying standard deviation is introduced to more accurately reflect the real changes of fluctuation, and thereby optimizes the trade signals. Guo [7] stated that China Stock Markets has been in operation for 20 years. Index return of China Stock Markets encompasses heteroskedasticity and leptokurtic with fat-tail in financial time series. With purposes of deeply analyzing the nature of the index return, the GARCH model is used for the Market index for empirical analysis. Geng and Liu proposed that the GARCH model is effective to describe the fluctuation of financial data [8]. It is the most commonly used and most convenient time-series heteroscedastic fitting model. Building up GARCH model and later predicting the rate of return and stock prices, could produce results that provide some references for investors' decision making.

Wang and Wu [9] used the GARCH model to conduct empirical research on the fluctuation and rate of return of the CSI 300 Index. They also achieved good results in accomplishment of fitting and prediction of the time series. Nie and Hu concluded the applications of the GARCH model in 10 fields [10], namely, finance, macroeconomic management and sustainable development, investment, securities, mathematics, market research and information, trade economy, industrial economy, agricultural economy industry, oil and natural gas industry, among which, the GARCH model accounts for the largest proportion in researches of the financial field, and further guides the research on realistic financial issues. Li and Xin [11] pointed out that the GARCH model can not only describe the heteroskedasticity of the time-series rate of return, but also, together with other financial analysis theories, make a more important contribution to the study of practical financial problems. Pan et al. used the GARCH model to fit a line to dynamic data of gold prices [12]. And the outcome which has practical significance for the future prediction and risk control of the market is remarkable.

In Liu’s (2004) study [13], two types of models, GARCH model and asymmetric GARCH model, were used to study actual data. He found that Shanghai stock market doesn’t have the characteristics of high risk and high return, but the asymmetric GARCH model shows that when the risk is higher than a certain level, there is a positive correlation between the rate of return and the risks. According to the dataset 1 and the penalty bulletins, it is possible to analyze the market manipulation, identifying its characteristics and judge its existence. Also, it can better assist the CSRC (China Securities Regulatory Commission) to supervise, creating a good investment environment, and maintain market orders. So, and TSE use the GARCH model to discover the phenomenon of volatility interaction with the Hong Kong Hang Seng Index and the Hang Seng Index futures market [14]. In terms of linked analysis between the markets, Sibel (2012) believes that financial infection is important for monetary policy [15], risk measurement, asset pricing and investment portfolio allocation. Therefore, the impact of the US subprime mortgage crisis on emerging markets is necessary. The Garch model is used to test the existence of financial infection between foreign exchange markets during the US subprime mortgage crisis. This model has certain advantages than other methods. Sabiruzzaman et al. believes that in addition to considering modeling of stock price fluctuations [16], we must also consider modeling transaction volume fluctuations, and transaction volume plays a key role in the financial market. Among them, they studied the fluctuation of the Hong Kong Stock Exchange’s daily transaction volume index, using the GARCH model to simulate the volatility of the transaction volume, and found that the GARCH model was well fitted with the data. BURNS [17] proposes that these commonly used multi-GARCH models are too high at estimates, so they improve the model and propose the PC-Garch model. This model reduces the calculation amount, can better fit the data, enables relevant estimates to be more stable, and better applies to the stock market. Prateek et al. compared the Realized Garch model and GARCH model [18], Egarch model in the prediction of the stock market volatility in 2020, Eric [19] et al. using the Garch-Midas model to use the monthly data of the macroeconomic variables. It proves that China’s stock market has speculation characteristics. At the same time, it shows that the volatility of the macroeconomic aspect in the stock market has played a larger and larger role, especially CPI inflation. When modeling the asset income, this article can use the Garch model (GARCH) model with a broad sense. This problem, so more and more scholars consider improving the distribution of model conditions. In order to further describe the nature of financial asset income, different types of partial -state thick tail distribution is in the GARCH model in the application, Mighri, SU, and PHERSbuilt the Garch-SKT model based on SKT distribution [20, 21, 22]. Robert In the discussion on the Garch model [23], the data detection, the number of samples and volume of the data, they analyzed the data within one year of the daily stock trading of an Australian company, and found that Garch corresponds to the corresponding correspondence The analysis results are closely related to the number of samples and the number of transactions. Cathy and Mike proposed a harem difference model [24], It combines the GARCH model and non -linear clusors to study the average and variance volatility of the model of the financial market. The research results show that when the average volatility is high, or when the fluctuations are more lasting, the negative effects will come. Visser [25] proposes the concept of scale model and volatility replacement, combining high -frequency data with the GARCH model within one day to analyze the factors that affect the stock market.

II METHODOLOGY AND SCHEMES

A Methodology

Literature research methodology: in this study, this article summarized the methods and characteristics of stock market price manipulation and gained a comprehensive and in-depth understanding of its behavior.

Case study: in this study, this article analyzed cases of stock price manipulation in recent years.

Empirical analysis: in this study, this article conducted empirical analysis using data from stock price manipulation cases published by the CSRC (China Securities Regulatory Commission) in recent years.

Quantitative data analysis:

Theoretically, the annual rate of return of the CSI 300 Index acts as the benchmark rate of return. And the annual return of individual stocks is the strategic rate of return.

$$B_r = (\frac{B_{end}}{B_{start}})^{(250/n)} - 1 \tag{1}$$

Bend = ending benchmark, Bustard initial benchmark, n= trading days for back testing.

Measuring the systematic risks in the investment process through the β value, reflects the investors’ strategic sensitivity to benchmark changes.

$$\beta = \frac{\operatorname{cov}(P_{n,} B_n)}{\sigma^2} \tag{2}$$

P_n = strategic benchmark return, B_n = daily benchmark return, σ^2 = variance of daily benchmark return, $\operatorname{cov}(P_n, B_n)$ = covariance of strategic benchmark return and daily benchmark return

α refers to excess returns irrelevant to the market, that is, non-systematic risk which can make indirect judgment upon market manipulation. Formula as follows:

$$\alpha = R_p -（R_f + \beta_p (R_m - R_f)）\quad (3)$$

R_p= strategic annualized return

R_m= benchmark annualized return

R_f= risk-free rate

β _p= strategic beta value

α >0,when R_p exceeds risks, excess returns gained, manipulation existed.

α =0,when R_p almost equals risks, appropriate returns gained, no manipulation.

α <0,when R_p is less than risks, a small quantity of returns gained, manipulation existed.

Quantitative data analysis: in this study, this article mainly calculated the data of the opening price, closing price, trading volume, and transaction amount, so as to obtain turnover rate, trading volume fluctuation and logarithmic return rate. Eventually this article obtained GARCH-α parameter and GARCH-β parameter in order to carry out data analysis and comparison with actual data. Model schemes are as follows (Fig. 1).

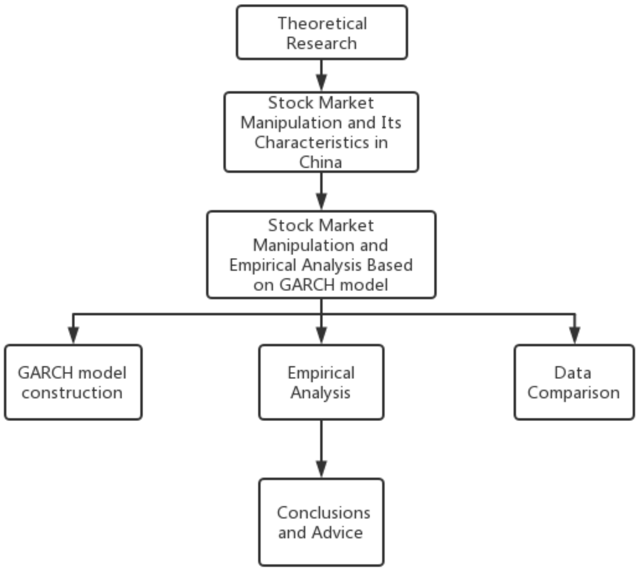

Theoretical Research
Stock Market Manipulation and Its Characteristics in China
Stock Market Manipulation and Empirical Analysis Based on GARCH model
GARCH model construction
Empirical Analysis
Data Comparison
Conclusions and Advice

Fig. 1. Model schemes.

Model analysis: In this study, this article utilized GARCH model as identification model of stock price manipulation.

Building up ARCH model is defined of the form:

$$y_t = \gamma_0 + \sum_{i=1}^{k} \gamma_i X_{it} + \mu_t \quad (4)$$

$$\sigma_t{}^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \mu^2{}_{t-1} \quad (5)$$

The Premise is that $\mu_t \sim (o, \sigma_t{}^2)$. ARCH model laid out solid foundations for subsequent analysis, which can better deal with heteroskedasticity and explain changes in market price fluctuations.

Building up GARCH model is defined as:

$$y_t = X_t \pi + \in_t \quad (6)$$

$$\sigma_t{}^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \alpha_{t-1}^2 \quad (7)$$

Variance σ_t^2 is subjective to the former residual ε_{t-1}^2 and the former variance α_{t-1}^2 while the sum of α and β is constantly the root of autoregressive model which determines the continuity of fluctuation shock. In many occasions, the root is very close to 1.

GARCH (p,q) is defined in the form as follows:

$$\sigma_t^2 = \omega + \alpha \sum_{i=1}^{q} \in_{t-1}^2 + \beta \sum_{j=1}^{p} \sigma_{t-j}^2 \quad (8)$$

P refers to the maximum lag of GARCH model while q refers to the maximum lag of ARCH model. In this study, independent testing will be carried out to predict price manipulation using GARCH model. Judgment will be made through the followings:

no manipulation: α + β ≦1

manipulation existed: α + β >1

Research schemes

input stock price data

calculate logarithmic return rate, turnover rate and trading volume fluctuation

draw sequence diagrams

descriptive statistic

stationary test (unit root test)

autocorrelation test

ARCH effect test

GARCH parameter estimation: this study conducted indicators like turnover fluctuation, trading volume fluctuation, prices fluctuation, GARCHA α parameter and GARCHA β parameter. Through comparative analysis, it is verified that GARCHA α parameter and GARCHA β parameter are effective in judging stock market price manipulation.

Calculate the GARCH fluctuation rate, conduct a comparative analysis with former data, and judge time points of manipulation with practical data of the stock.

III Analysis of Empirical Results

A Indicator Analysis

In terms of forecasting model construction, this study collected fifteen stock price data in dataset 1, seven of which had market manipulation, and eight had none. On that data basis this article constructed GARCH model to identify market manipulation. Finally, whether the sum of GARCH-α and

GARCH-β is greater than 1 using logarithmic return rate, trading volume fluctuation, turnover rate along with their fluctuation forecast trend charts will be set as judging standards of manipulation’s existence.

Logarithmic return rate formula as follows:

R=ln(closing price on a day)-ln(closing price of the former day)

Trading volume fluctuation formula as follows:

TVF = turnover on a day- turnover of the former day

Turnover rate formula as follows:

TR = turnover/sum of issuing stocks *100%

Abbreviations

R: Logarithmic return rate

TVF: Trading volume fluctuation

TR: Turnover rate

B Trend Chart Analysis

Fig. 2 to 8 shows the four indicator trend comparison.

In the study, closing price, R, TVF, and TR in seven stocks with market manipulation in dataset 1 are compared using EViews software. It is found that during the manipulation, closing price, R, TVF, and TR all showed a sharp rise while they experienced an obvious slump after the manipulation. Besides that, there was data loss before and after manipulation. Take Tianjin Futong Information Science& Technology Co., Ltd. (stock code: 000836), its data went missing from May 24, 2017 to November 24, 2017.

By comparing the trend chart and the data analysis of the China Securities Regulatory Commission announcement, it can be seen that the market manipulation has the following characteristics:

The participants in stock market manipulation are complex and diverse.

From the announcement data, this article can know that the main body of the stock manipulation are those who hold significant authority and social status including the the chairman of the board, insiders, legal representatives, general managers, shareholders, the actual controller, managing partners and other administrative seniors.

Manipulation holds characteristics of zero transparency and severe consequences.

To evade the supervision of the China Securities Regulatory Commission, insiders usually adopt relatively covert methods. Specifically, insiders use other people’s accounts instead of their own when manipulating. Huang Xiaodiao as one of insiders borrowed other people’s accounts and trade stocks of Neoglory Prosperity Inc. (stock code: 002147). The China Securities Regulatory Commission imposes heavy fines on stock market manipulation, and adopts measures to ban the parties from entering the stock market within a specified period, with serious penalties.

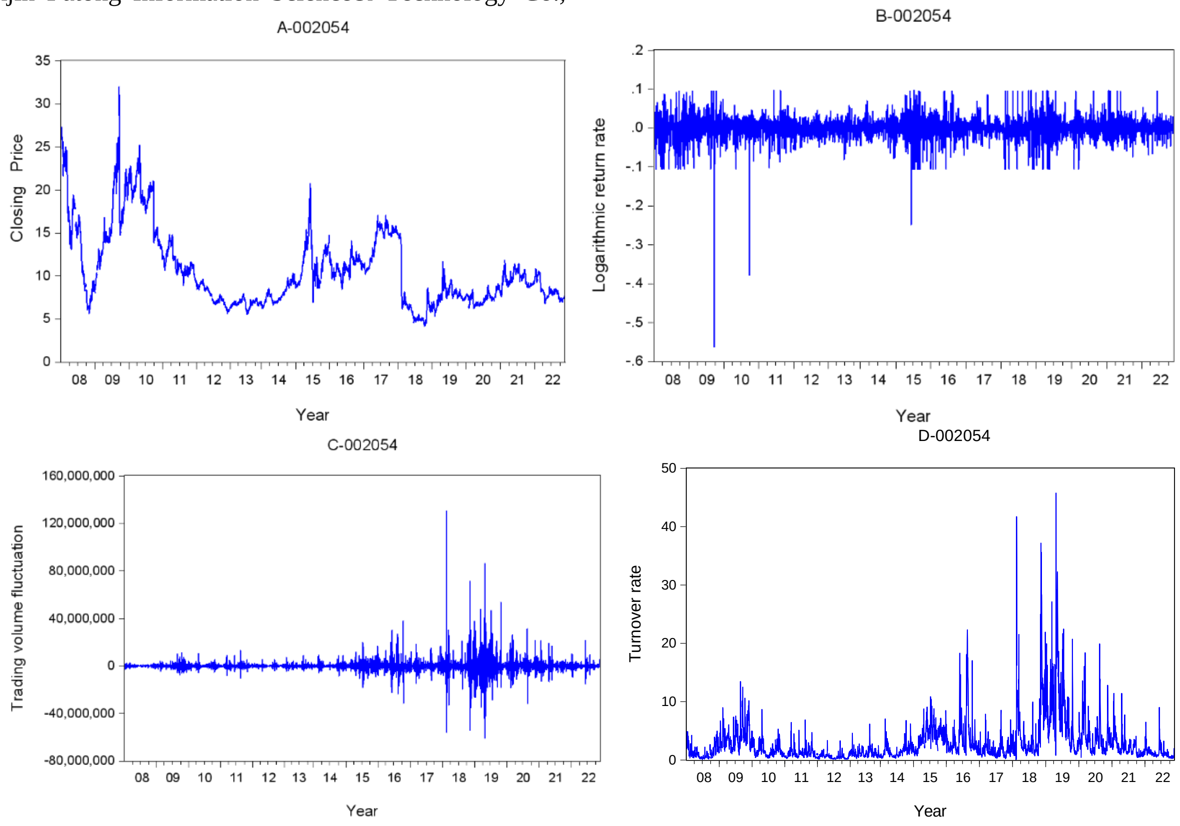


Fig. 2. Stock 002054 four indicator trend comparison.

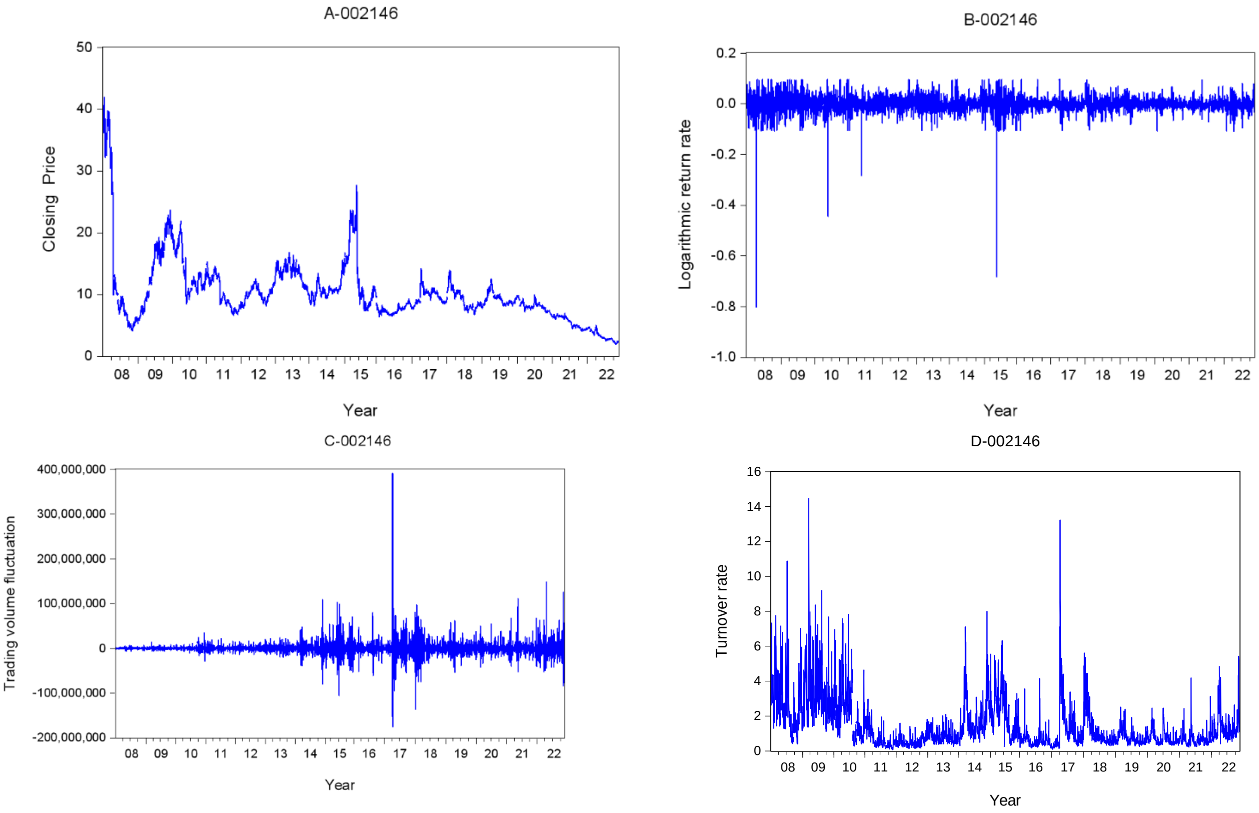


Fig. 3. Stock 002146 four indicator trend comparison.

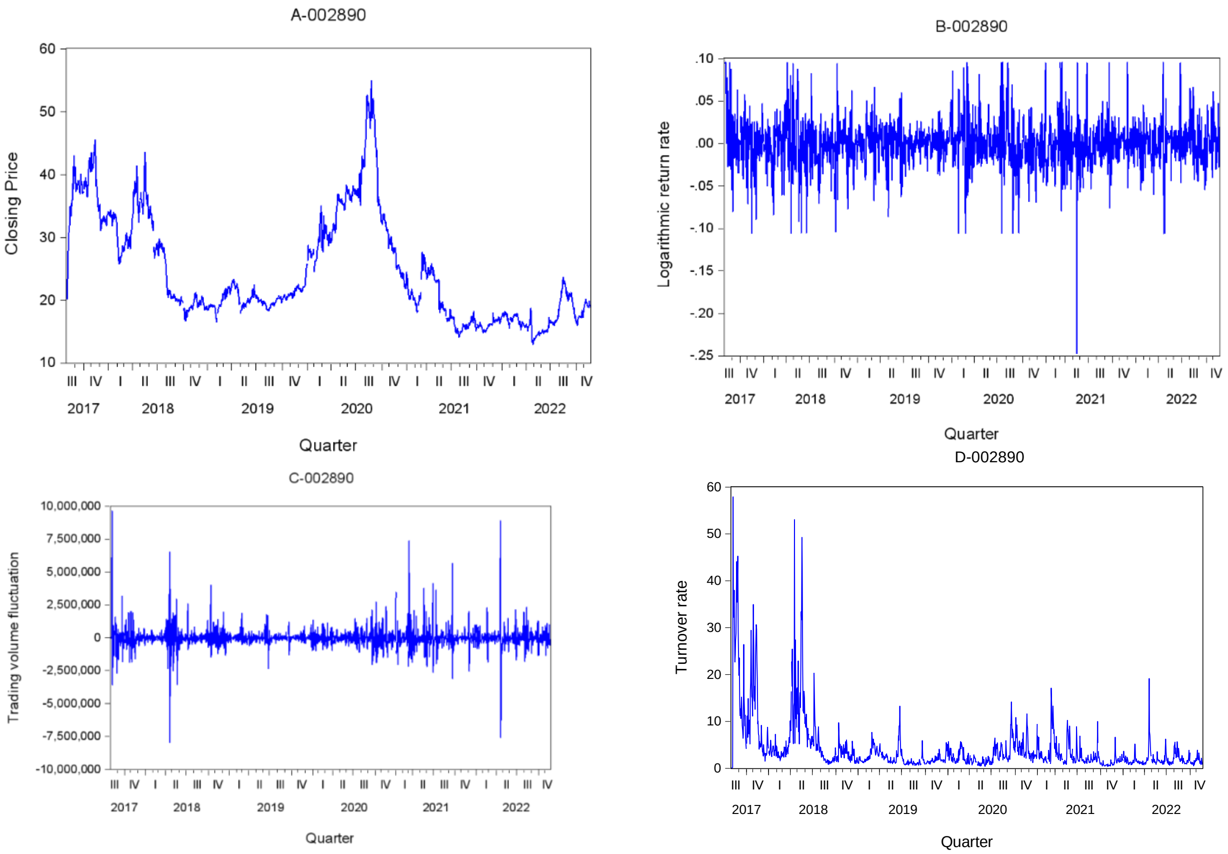


Fig. 4. Stock 002890 four indicator trend comparison.

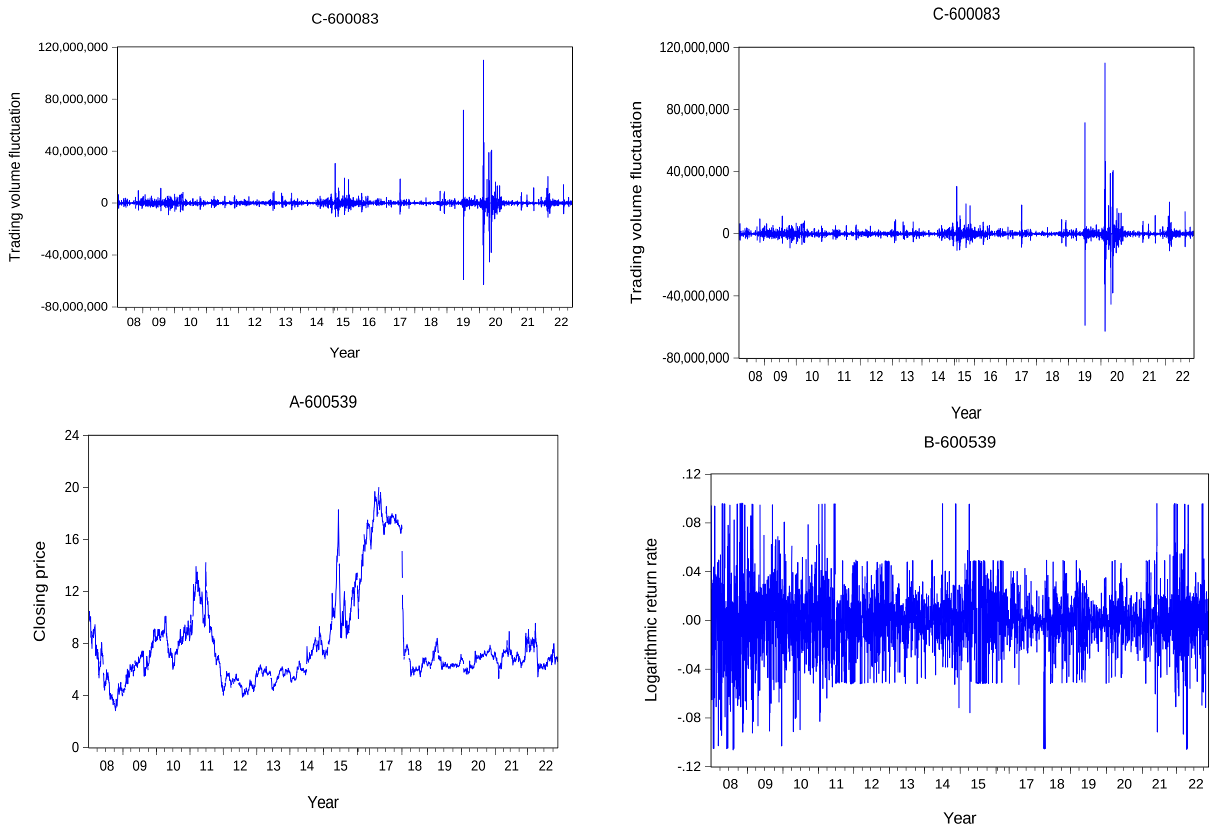


Fig. 5. Stock 600083 four indicator trend comparison.

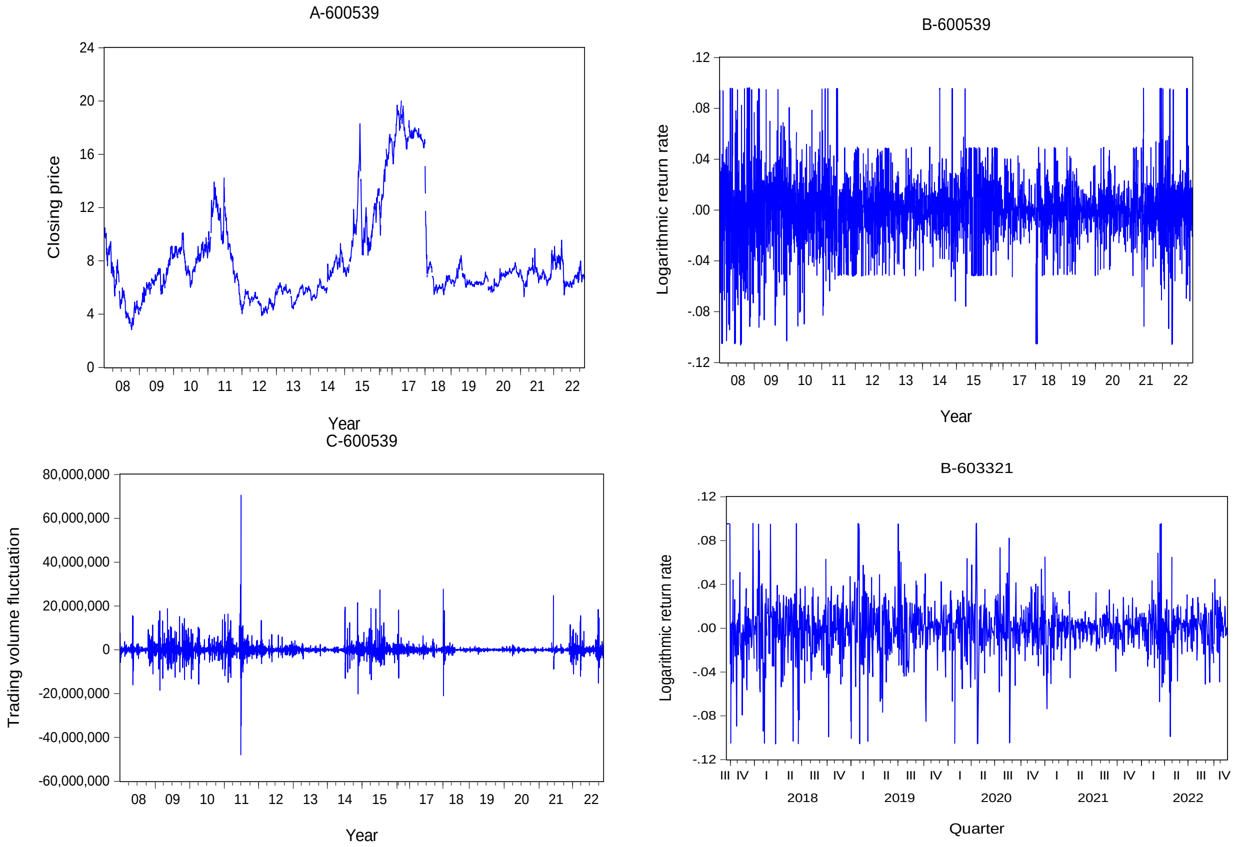


Fig. 6. Stock 600539 four indicator trend comparison.

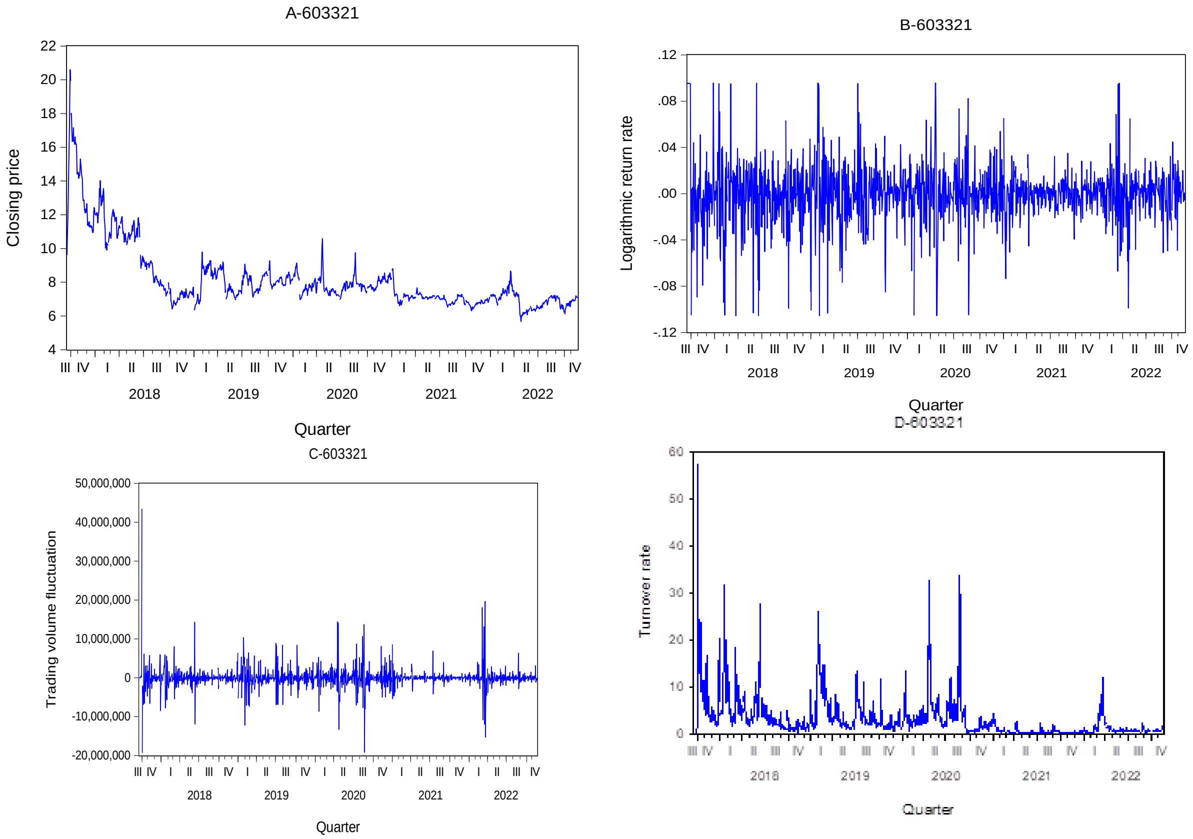


Fig. 7. Stock 603321 four indicator trend comparison.

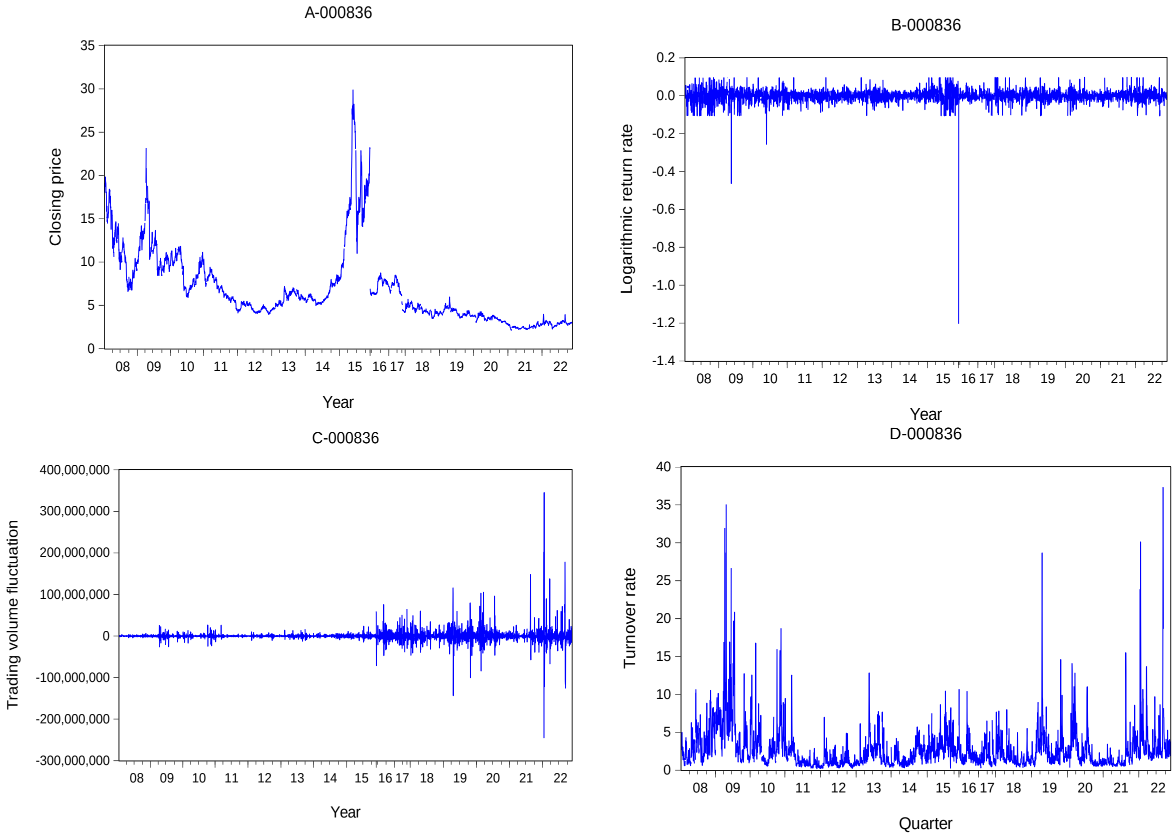


Fig. 8. Stock 000836 four indicator trend comparison.

Stock market manipulation is widespread in many industries.

Based on the stock data of dataset 1 and the China Securities Regulatory Commission’s penalty announcement, stock market manipulation can be seen in the tourism real estate industry and machinery manufacturing industry (Neoglory Prosperity Inc. stock code: 002147), in optical fibre and cable industry and quartz-fibre manometer ware industry(Tianjin Futong Information Science& Technology Co., Ltd, stock code: 000836), in elevator industry (Zhejiang Meilun Elevator Co.,Ltd, stock code: 603321), in real estate industry (RiseSun Real Estate Development Co., Ltd, stock code: 002146), in main business of shipping and terminal operation(COSCO SHIPPING Holdings Co., Ltd, stock code: 601919), in construction and building materials industry (Taiyuan Lionhead Cement Co., Ltd, stock code: 600539). They are the lively examples indicating wide ranges of manipulation.

Stock market manipulation takes various forms:

By controlling the account information and accounts that in full charge to engage in securities transactions. Liu Xiaodong from Dymatic Chemicals,Inc (stock code:002054) colluded with Hu Kan from Zhejiang Meilun Elevator Co.,Ltd. (stock code:603321) using multiple securities accounts to buy in a concentrated manner, and apply for large orders to drive up the stock prices.

C Empirical Analysis

From Table I, this article can conclude that in two stocks, the sum of GARCH-α and GARCH-β in R, TVF and TR was all greater than 1. While in four stocks, the sum of GARCH-α and GARCH-β in TVF and TR was greater than 1 altogether with the sum of GARCH-α and GARCH-β in R greater than 0.97. The results proved that the model is marvelous in manipulation identification.

From Table II, this article can conclude that in two stocks, the sum of GARCH-α and GARCH-β in R, TVF and TR was less than 1. While in two stocks the sum of GARCH-α and GARCH-β in R and TR was less than 1. And in other two stocks the sum of GARCH-α and GARCH-β in R and TVF was less than 1.

D Model Checking

This study collects five stocks in dataset 2, and conducted GARCH model detection and identification on R and TVF, so as to predict and judge the existence of manipulation along with its time points and characteristics.

With identification and detection of GARCH model, evaluation results of two indicators for five stocks in dataset 2 were showed in Table III. From the table above, this article can conclude that in the first and third stocks, the sum of GARCH-α and GARCH-β in R and TVF was less than 1. In other 3 stocks, the sum of GARCH-α and GARCH-β in R was less than 1 while the sum of GARCH-α and GARCH-β in TVF was greater than 1. To improve the accuracy of the model, this study splits the six-year stock data of five stocks into annual data for re-testing. The evaluation results are as follows (see Tables IV-VIII).

TABLE I. EVALUATION RESULTS OF THREE INDICATORS FOR SEVEN STOCKS WITH MANIPULATION

	000836	**600083**	**2054**	**2146**	**2890**	**600539**	**603321**
R-α	0.2138	0.0779	0.1516	0.1865	0.1767	0.1103	0.1318
R-β	0.8329	0.9008	0.8337	0.8397	0.6244	0.8744	0.8613
sum	1.0467	0.9786	0.9854	1.0262	0.8011	0.9847	0.9931
TVF-α	0.5563	0.8384	0.4355	0.2018	0.3045	1.2759	1.046
TVF-β	0.7095	0.5869	0.7808	0.8927	0.331	0.5369	0.0027
sum	1.2658	1.4253	1.2163	1.0945	0.6356	1.8128	1.0487
TR-α	0.4148	0.8694	0.3827	0.2331	0.4097	1.1418	1.046
TR-β	0.6322	0.533	0.7929	0.7856	0.7133	0.6172	0.0027
sum	1.047	1.4025	1.1756	1.0187	1.123	1.7591	1.0487

TABLE II. EVALUATION RESULTS OF THREE INDICATORS FOR EIGHT STOCKS WITHOUT MANIPULATION

	300075	**300274**	**858**	**300059**	**600760**	**601919**	**625**	**713**
R-α	0.2704	0.1215	0.0624	0.0569	0.0653	0.0659	0.1816	0.144
R-β	0.6688	-0.0303	0.9173	0.9164	0.918	0.9225	0.8373	0.8097
sum	0.9392	0.0912	0.9796	0.9734	0.9833	0.9884	1.0189	0.9537
TVF-α	0.2977	0.532	0.0537	0.7075	0.2761	0.6196	0.2725	0.2019
TVF-β	0.7805	-0.0084	0.9397	0.6059	0.6479	-0.0028	0.7725	0.8261
sum	1.0782	0.5236	0.9934	1.3134	0.9239	0.6168	1.0449	1.0279
TR-α	0.1995	0.2055	0.0556	0.0937	0.477	0.2904	0.1716	0.246
TR-β	0.7613	0.7763	0.9404	0.8861	0.6032	0.807	0.871	0.7997
sum	0.9608	0.9818	0.9961	0.9798	1.0802	1.0974	1.0426	1.0457

TABLE III. EVALUATION RESULTS OF TWO INDICATORS FOR 5 STOCKS IN DATASET 2

	1	**2**	**3**	**4**	**5**
R-α	0.1195	0.1499	0.0974	0.0944	0.17447
R-β	0.8222	0.8045	0.8855	0.8453	0.772909
sum	0.9417	0.9544	0.9828	0.9397	0.9474
TVF-α	0.3121	0.9221	0.5972	0.2529	0.221103
TVF-β	0.4799	0.4924	-0.0188	0.7514	0.845225
sum	0.792	1.4145	0.5784	1.0042	1.0663

TABLE IV. ANNUAL EVALUATION RESULTS OF TWO INDICATORS IN STOCK 1

	2016	**2017**	**2018**	**2019**	**2020**	**2021**
R-α	0.037108	0.050826	0.305762	0.110901	0.234068	0.028942
R-β	1.025163	1.031713	0.507474	0.872531	0.716099	0.916706
sum	0.988055	0.980887	0.813236	0.983432	0.950167	0.945648
TVF-α	0.73323	0.144921	1.016567	0.12721	0.482486	0.559319
TVF-β	0.413856	0.048914	0.001476	0.89974	0.518359	0.050465
sum	1.147086	0.193835	1.018043	1.02695	1.000845	0.508854

TABLE V. ANNUAL EVALUATION RESULTS OF TWO INDICATORS IN STOCK 2

	2016	**2017**	**2018**	**2019**	**2020**	**2021**
R-α	0.060005	0.486671	0.235432	0.118615	0.33448	0.141124
R-β	1.049991	0.329566	0.346146	0.829489	0.726385	0.740907
sum	0.989986	0.816237	0.581578	0.948104	1.060865	0.882031
TVF-α	0.47295	0.128954	1.528713	0.467837	1.396246	0.642267
TVF-β	0.133565	0.020445	0.066155	0.071897	0.425462	0.090602
sum	0.339385	0.108509	1.462558	0.539734	1.821708	0.551665

TABLE VI. ANNUAL EVALUATION RESULTS OF TWO INDICATORS IN STOCK 3

	2016	**2017**	**2018**	**2019**	**2020**	**2021**
R-α	0.057664	0.202555	0.027912	0.091298	0.236559	0.100316
R-β	1.023889	0.803294	0.3513	0.872017	0.666382	0.842742
sum	0.966225	1.005849	0.323388	0.963315	0.902941	0.943058
TVF-α	0.625886	0.816549	0.175514	0.294882	0.95409	0.326408
TVF-β	0.068155	0.097357	0.024964	0.568844	0.076571	0.082987
sum	0.694041	0.913906	0.200478	0.863726	0.877519	0.243421

TABLE VII. ANNUAL EVALUATION RESULTS OF TWO INDICATORS IN STOCK 4

	2016	**2017**	**2018**	**2019**	**2020**	**2021**
R-α	0.038497	0.219204	0.13059	0.094054	0.101291	0.074151
R-β	0.931862	0.611703	0.458642	0.846511	0.693069	0.890664
sum	0.970359	0.830907	0.589232	0.940565	0.79436	0.964815
TVF-α	0.861852	0.436585	0.544132	0.264732	0.272204	0.120104
TVF-β	0.099919	0.030305	0.220179	0.606584	0.518322	0.720804
sum	0.961771	0.40628	0.764311	0.871316	0.790526	0.840908

TABLE VIII. ANNUAL EVALUATION RESULTS OF TWO INDICATORS IN STOCK 5

	2016	**2017**	**2018**	**2019**	**2020**	**2021**
R-α	0.049556	0.005235	0.06662	0.295535	0.177969	0.113084
R-β	1.023353	0.526743	0.885713	0.5255	0.757596	0.848661
sum	0.973797	0.521508	0.952333	0.821035	0.935565	0.961745
TVF-α	0.57346	0.038856	2.602458	0.203553	0.306933	0.149094
TVF-β	0.041831	0.32493	0.111961	0.790593	0.150759	0.08027
sum	0.531629	0.286074	2.714419	0.994146	0.457692	0.229364

After the test, this article found that the sum of GARCH-α and GARCH-β of some indicators was greater than 1 in stock 1 and stock 3. But this article still believed both has no manipulation, considering the total time was less than 1, and fluctuations of indicators could arise from company's thriving development. In all the five years (Table III), the sum of GARCH-α and GARCH-β of R in stock 4 was less than 1 while the sum of GARCH-α and GARCH-β of TVF was greater than 1. However, the sum of GARCH-α and GARCH-β of R and TVF was less than one annually. Therefore, this article believed that there was no manipulation in stock 4.

Besides, the sum of GARCH-α and GARCH-β of R and TVF in 2020 of stock 2 was greater than 1. While the sum of GARCH-α and GARCH-β of TVF in the 5 years (Table III) was greater than 1. Data was abnormal. The sum of GARCH-α and GARCH-β of TVF in 2018 of stock 5 was way greater than 1 while the sum of GARCH-α and GARCH-β of R and TVF was less than 1. Therefore, this article analyzed on stock 2 and stock 5 with the following Fig. 9 and 10.

From Fig. 9, R and TVF of stock 2 was asymmetric in mid-October. However, its TVF had a significant surge in mid-October. Therefore, this article believes manipulation took place in mid-October 2020. R and TVF of stock 5 showing a great opposite trend from mid-March to April could also be the prove that manipulation happened from mid-March to around April in 2018.

Characteristics of the trend charts are as follows:

Opposite trends took place in two indicators' prediction value.

The R prediction value is rather low while the TVF prediction value promptly rises.

The closing price, R and TVF snowballed during the manipulation.

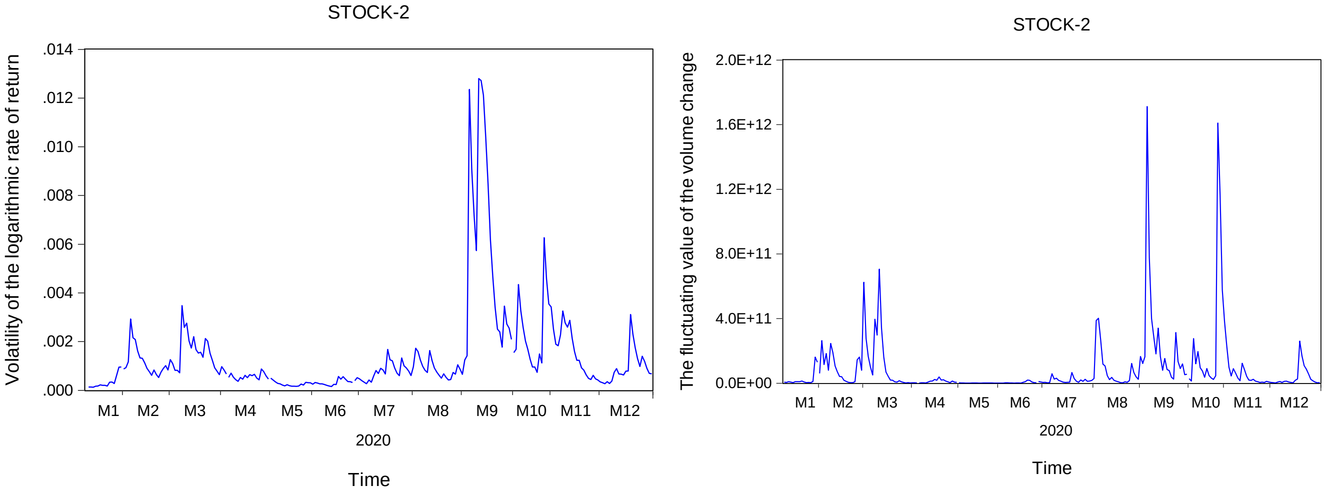


Fig. 9. R and TVF in 2020 of stock 2.

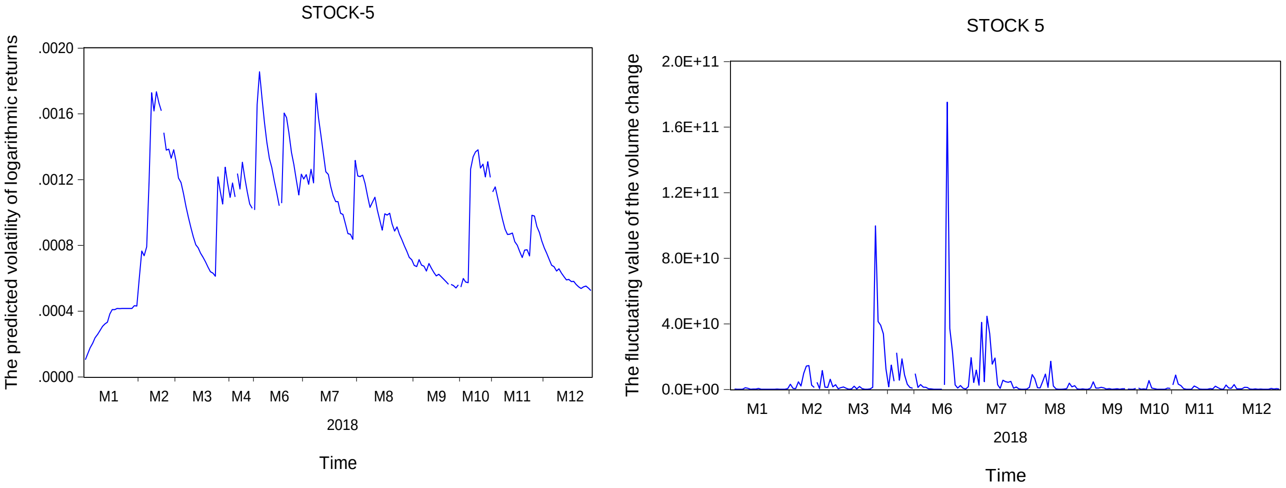


Fig. 10. R and TVF in 2018 of stock 5.

IV CONCLUSIONS AND ADVICE

A Conclusions

This paper analyzes the manipulation behavior of the stock market based on relevant theories and GARCH model. On the basis of GARCH model, data analysis is carried out by using EViews software. Meanwhile, indexes such as logarithmic rate of return, turnover rate change, turnover change, GARCH-α and GARCH-β are constructed for manipulation behavior identification.

Analysis of the data obtained in this study shows that when the closing price, logarithmic return rate, trading volume change and turnover rate increase sharply, the market manipulation probability of individual stocks increases significantly, and after the manipulation, the indexes all show a precipitous decline, in which the Fortis information (stock code: 000836) represented by May 24, 2017 to November 24, 2017. This is basically consistent with the empirical analysis of this study. At the same time, Ding (2019) constructed GARCH model, Zhang (2015) based on the GARCH model of time series, Jiang (2022) eliminated the influencing factors of adding heteroskedasticity to time series, and Liu Ning (2004), Wang Wu (2011), Geng, Liu (2019), Nie and Hu (2020) et al. learned from relevant GARCH model construction that the characteristics of high risk and high return in Shanghai stock market do not exist in the basic view of finance. However, according to the asymmetric GARCH model, the above researches show that when the risk is higher than a certain level for a period of time, there is a direct proportional relationship between return rate and risk. This is consistent with the analysis results in this study. Based on the scholars' literature on the application of relevant GARCH model in stock price manipulation analysis and the summary of this paper, it is known that: There are risks in the market, but when the relevant indicators of the stock price fundamentals fluctuate too much, the relevant GARCH model can be used for analysis and research, and the probability risk of stock price manipulation can be obtained, so as to carry out reasonable position planning and market management, and guide managers and investors to make more optimal decisions.

It is known from the empirical results that the prediction accuracy rate is 85.71%, but due to the small number of samples, the reliability of such data analysis results is low. In the subsequent improvement, the sample size can be increased to reduce the error value. 4. The index studied in this paper is mainly based on the price index. In addition to controlling trading volume and changing stock prices, manipulation in the actual market is often filled with information differentiation, which increases the difficulty in detecting market manipulation. Therefore, based on this model, other dimensional models should be added to monitor market stocks combined with multidimensional models, which can effectively enhance the early warning system of advance indicators and achieve better results for regulatory departments to maintain market orderly and regularized.

B Advice

The regulatory authorities should further classify the market manipulation behaviors and establish the benchmark values of various indicators, which is conducive to the faster and more accurate detection of illegal market behaviors in the follow-up monitoring.

The increase of sample size can more effectively reduce errors caused by model data, make model monitoring more accurate, reduce fault tolerance rate and improve accuracy.

The competent authorities should put forward the requirements of strengthening education for all investors in the market, at the same time improve relevant systems and regulations, improve anti-manipulation supervision methods, make the market more legal and systematic, and provide a healthy and green investment environment for market investors.

V CONFLICTS OF INTEREST

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ACKNOWLEDGMENT

This research was funded by the Hunan Philosophy and Social Science Foundation Project no. 20YBA121.

REFERENCES

[1] R. Y. Chou, “Volatility Persistence and Stock ValuationSome Empir-ical Evidence Using Garch.” Journal of Applied Econometrics, 1986, vol. 31, pp. 307-327.

[2] S. H. Poon, and S. J. Taylor, “Stock returns and volatility: an empirical study of the UK stock market.” Jounal of Banking and Finance, 1992, vol. 16, pp. 37-59.

[3] J. A. Cai, “Markov Model of Switching R egime AR CH.” Journal of Business & Economic Statistics, 1994, vol. 12, pp. 309-316.

[4] J. Ding, “Research on the Identification of Stock Price Manipulation in China’s Stock Markets”, 2019, 9.

[5] G. Zhang, “Research on China’s Stock Market Manipulation and Supervision”, 2015.

[6] Y. D. Jiang, “Research on Stock Arbitrage Strategy Based on GARCH Model and Cop. OU Process, 2022, vol. 5, pp. 27.

[7] Y. Guo, “Application of GARCH Model in the Research of Stock Market Index Yield Fluctuation.” 2016.

[8] J. Geng, Y. C. Liu, “Analysis and Forecast of Stock Returns Based on GARCH Model”, 2019, pp. 010.

[9] J. F. Wang, Q. Wu, “Analysis and Forecast of China Stock Markets Fluctuation Based on GARCH Family Model,” 2011, pp. 34-0074-05.

[10] Q. P. Nie, B. Q. Hu, “A Summary of GARCH Model Family and Its Application in Financial Markets, 2020.

[11] Y. Li, S. B. Xin, “Research on the application of GARCH family models in China,” 2014 (5).

[12] G. H. Pan, N. L. Hu, H. Z. Liu, G. Q. Li, “Empirical Analysis of Gold Prices Based on ARMA-GARCH Model.” Gold, 2009, vol. 31, pp. 5-8.

[13] N. Liu, “ARCH Research on Shanghai Stock Market Fluctuation”. 2004, vol. 12, pp. 18-22.

[14] R. So, & Y. Tse ,2004, “Price dicover in the Hangseong Index Maks: lndad, Future, and bhe Tacker Fund”, louadlof Futures Markets, 2004, vol. 24(9), pp. 887-907.

[15] C. SIBEL, “The more contagion effect on emerging markets:the evidence of DCC-GARCH model.” Economic Modelling, 2012, vol. 29, pp. 1946-1959.

[16] M. Sabiruzzaman, H. Monimul, et al. “Modeling and fore－ casting trading volume index: GARCH versus TGARCH approach.” The Quarterly Review of Economics and Finance, 2010, vol. 50, pp. 141-145.

[17] P. Burns, “Multivariate GARCH with only univariate estimation 2009-06-20）. http://www.burns-stat.com.

[18] S. Prateek, Vipul. “Forecasting stock market volatility using realized GARCH model: International evidence”. The Quar- terly Review of Economics and Finance, 2016, vol. 59, pp. 222-230.

[19] G. Eric, J. Roselyne, “Macro fundamentals as a source of stock market volatility in China:a GARCH-MIDAS approach”. Economic Modelling, 2013, vol. 34, pp. 59-68.

[20] Z. Mighri, K. Mokni, “Mansouri F Empirical analysis of asymmetric long memory volatility modelsin value-at-risk estimation”. Journal of Risk, 2010, vol. 13(1), pp. 55-128.

[21] J. B. Su, “How to mitigate the impact of inappropriate distributional settings when the parametric value at-risk approach is used.” Quantitative Finance, 2013, vol. 14(2), pp. 1-21.

[22] R. S. Phers, “Computational tolos for comparing aymeti GARCH models via Bayes fcotou.” Mathematics and Computers in Simulation, 2012, vol. 82, pp. 858-867.

[23] D. B. Robert, W. F. Robert, “GARCH modeling of individual stock data the impact of censoring”, Journal of International Financial Markets, 2001, vol. 11, pp. 215-222.

[24] W. S. Cathy, K. P. Mike, “On a threshold heteroscedastic model.” International Journal of Forecasting, 2006, vol. 22, pp. 73- 89.

[25] M. P. Visser, “GARCH parameter estimate using high-frequency data.” Journal of Financial Econometrics, 2011, vol. 9(1), pp. 162-197.

Cloud Task Scheduling using the Squirrel Search Algorithm and Improved Genetic Algorithm

Qiuju DENG[1], Ning WANG[2], Yang LU[3]
Chongqing College of Mobile Communication, Chongqing, 401520, China[1, 2, 3]
Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing, 401420, China[1]

***Abstract*—With cloud computing, resources can be networked globally and shared easily between users. A range of heterogeneous needs are met on demand by software, hardware, storage, and networking. Dynamic resource allocation and load distribution pose challenges for cloud servers. In this regard, task scheduling plays a significant role in enhancing the performance of cloud computing. With the increase in the number of users and the capability of cloud computing, cloud data centers are experiencing concerns regarding energy consumption. To leverage cloud resources energy efficiently and provide real-time services to users, a viable cloud task scheduling solution is required. To address these problems, this paper proposes a new hybrid task scheduling algorithm based on squirrel search and improved genetic algorithms for cloud environments. The proposed scheduling algorithm surpasses existing scheduling algorithms across multiple parameters, including makespan, energy consumption, and execution time.**

Keywords—Cloud computing; energy efficiency; task scheduling; genetic algorithm

I. Introduction

Recent technological and scientific advances in Complementary Metal-Oxide Semiconductor (CMOS) [1, 2], machine learning [3], cloud computing [4], 5G connectivity [5, 6], Blockchain [7], artificial intelligence [8, 9], smart grids [10], Internet of Things (IoT) [11, 12], and optical networks [13, 14] are bringing numerous benefits to society. Schedulers (brokers) in cloud computing determine potential solutions for assigning constrained resources to requests in order to achieve multiple goals (e.g., energy consumption, response time, resource utilization, reliability) [15-17]. It is believed that the study conducted in [18] laid the foundation for modern scheduling techniques. Schedules are used in many applications today, including power system control, multi-media data object scheduling on the Internet, and manufacturing printed circuit boards [19]. Over the past three decades, distributed computing systems have become one of the most important aspects of modern scheduling [20]. In recent years, various standalone computers have been combined with working together as a cluster system. By integrating heterogeneous resources from geographically dispersed areas, grid systems overcome the shortcomings of cluster systems by using more resources [21]. Cloud computing has recently become popular, combining the strengths of clusters and grids [22].

Due to the wide solution space, most scheduling problems are NP-hard and require a long period of time to be resolved within a minimal period [23]. The scheduling of limited resources in modern computing systems cannot be optimized using a polynomial time-scheduling algorithm [24]. The researchers of [25] illustrated the dilemma posed in this case by giving a simple example: approximately 0.02 percent of the possible solutions consume up to 1.01 the necessary amount of time to reach the optimal result. It is proven that a complex problem can be extremely challenging to solve. Therefore, researchers have been motivated to develop effective algorithms to solve such scheduling problems. Scheduling techniques can be static or dynamic [26]. Due to the dynamic nature of cloud environments, more dynamic algorithms must be incorporated to achieve breathtaking results. In contrast, static algorithms are only used when workloads vary only slightly. Thus, deterministic methods cannot solve the task scheduling problem. This problem has been solved significantly in polynomial time by meta-heuristic algorithms, which are non-deterministic methods [27].

Virtualization technology and dynamic task scheduling techniques can benefit cloud service providers and users. By scheduling tasks effectively, resources are conserved (the resource utilization ratio is increased), and incoming tasks are also completed in the shortest possible time (the makespan is minimized) [28]. With the growing workloads in cloud data centers, task scheduling has become increasingly critical due to the scarcity of resources. In order to improve QoS criteria and the mapping of incoming tasks to available resources, cloud task scheduling needs further study. In scheduling, the goal is to determine optimal resources for executing incoming tasks, thereby enabling a scheduling algorithm to enhance various QoS factors such as response time, energy consumption, resource utilization, and makespan [29]. The rest of the paper is organized as follows. The next section reviews the previous works. Section III describes the proposed method. Experimental results are reported in Section IV. The conclusion is provided in Section V.

II. Related Work

A QoS-aware cloud task scheduling algorithm was proposed by Wu, et al. [30]. In the proposed algorithm, tasks are first prioritized using their special attributes, then sorted according to their priority. Second, the algorithm schedules tasks based on the sorted task queue according to the completion time for each task on different services. Based on CloudSim experiments, the algorithm can achieve good load balancing and performance by using priority and completion time to determine QoS. An improved sunflower optimization algorithm was introduced by Emami [31] for optimizing existing task scheduling algorithms. The algorithm schedules tasks in polynomial time. Experimental results have shown that the algorithm outperforms its competitors. Makespan and energy consumption have improved by 0.74% and 3%, respectively, compared to the best counterpart.

Yang, et al. [32] developed a simplified cloud computing task scheduling model. This paper uses game theory to simplify cloud computing task scheduling algorithms as opposed to previous studies. This algorithm considers the reliability of a balanced task when scheduling tasks with game theory. A task scheduling model for computing nodes is developed based on the balanced scheduling algorithm. The rate allocation strategy is calculated using game strategy in the cooperative game model. Experimental results indicate that the proposed algorithm performs better than others.

Srichandan, Kumar [6] developed an approach to task scheduling that combined the advantages of two widely used biologically-inspired algorithms: genetic and bacterial foraging. This article makes two main contributions. In the first place, the scheduling algorithm minimizes the time between tasks, and in the second place, it reduces energy consumption economically and environmentally. According to experimental results, the proposed algorithm provides superior performance for convergence, stability, and solution diversity.

Abd Elaziz, et al. [33] presented a method for scheduling cloud tasks to minimize the time consumed scheduling different tasks across different virtual machines. This method uses Differential Evolution (DE) to improve the Moth Search Algorithm (MSA). The MSA mimics moth movements in nature using Levy flights and phototaxis as indicators of the ability to explore and exploit resources. The exploitation ability still needs to be improved so that DE can be used for local searches. Three experimental series are conducted to evaluate the proposed algorithm. An analysis of twenty global optimization problems is carried out using the traditional MSA and the proposed method in the first experiment. The proposed algorithm was compared to other meta-heuristic and heuristic algorithms on synthetic and real trace data in the second and third experimental series. Performance measurements in both experiments demonstrate that the proposed algorithm outperforms competitors.

Using cat swarm optimization algorithm, Mangalampalli, et al. [34] addressed data center-specific parameters, such as power consumption, migration time, and makespan. VM mapping was performed by calculating the priorities of tasks at the task level. Based on the cloudsim simulator, this algorithm generates random inputs for total power costs. HPC2N and NASA workload archives are used as inputs to the algorithm. The proposed algorithm is compared to existing algorithms such as PSO and CS. Using HPC2N and NASA workloads, significant improvements are observed in different parameters.

Various meta-heuristic algorithms have been used in the works discussed above. These approaches share the common characteristic of using random population to initialize the metaheuristics and hybrid metaheuristics. The initial population has a significant impact on metaheuristic algorithms. Randomness is a fundamental requirement for avoiding local minimum traps. However, the algorithm convergence can be improved if some particles are assisted heuristically in selecting effective or near optimum starting points. The proposed algorithm utilizes heuristic algorithms for initializing the papulation in order to significantly improve the algorithm's performance.

III. Proposed Method

There are many scheduling algorithms to minimize the tasks' completion time in distributed systems. These types of scheduling systems find the most proper resources to assign to the tasks. Minimizing the tasks' completion time does not lead to minimizing each task's execution time. Task scheduling goals in cloud computing are to propose optimal scheduling of the tasks with load balancing guarantee and guarantee Quality of Service (QoS) criteria like response time, execution time, system throughput, cost, reliability, and availability. A new method is proposed for scheduling cloud tasks.

A. *Formulating the Problem*

The utilized method has four main parts, including the network information server, the network resource broker, the tasks, and the resources that act in the following manner. Users send requests to process tasks. The information about the task is embedded in the request, including the required CPU time for each task, the size of each task, and the total number of tasks. The network resource broker starts calculating the program parameters after the received message from the user. Moreover, the information server provides the resources information for the network resource broker. The proposed method will be used to select the input for processing the resource. The local update of the nodes is performed after assigning a task to a resource. The global update of the nodes is performed after executing a task by a reference. Fig. 1 shows the flowchart of the proposed algorithm.

The execution results are transmitted to the user. The fitness function is the function that receives a candidate solution for a problem as input and provides an output that determines a good amount of the solution. The key characteristic of the optimization algorithms is determining the fitness value of each solution. The algorithm tries to schedule K tasks to M virtual machines in each repetition. Virtual machines are optimally scheduled in accordance with their processing capacity, given by Eq. (1).

$$Capacityvm_j = Mipsvm_j \times PesNumvm_j \quad (1)$$

where $pesNumbervmj$ is the number of processors in the vmj virtual machine and MIPSVMJ is the number of million instructions per second of all processors on VMJ virtual machine. Task scheduling reduces the execution time of virtual machine tasks. The execution time is estimated by Eq. (2).

$$ExecutionTime_j = \frac{TaskLength_j}{Capacityvm_j} \quad (2)$$

where $TaskLengthj$ denotes the length of the jth request on the queue, and Capacityvmj refers to the processing capacity of the virtual machine on the jth location of the solution (J=1, 2, ..., K).

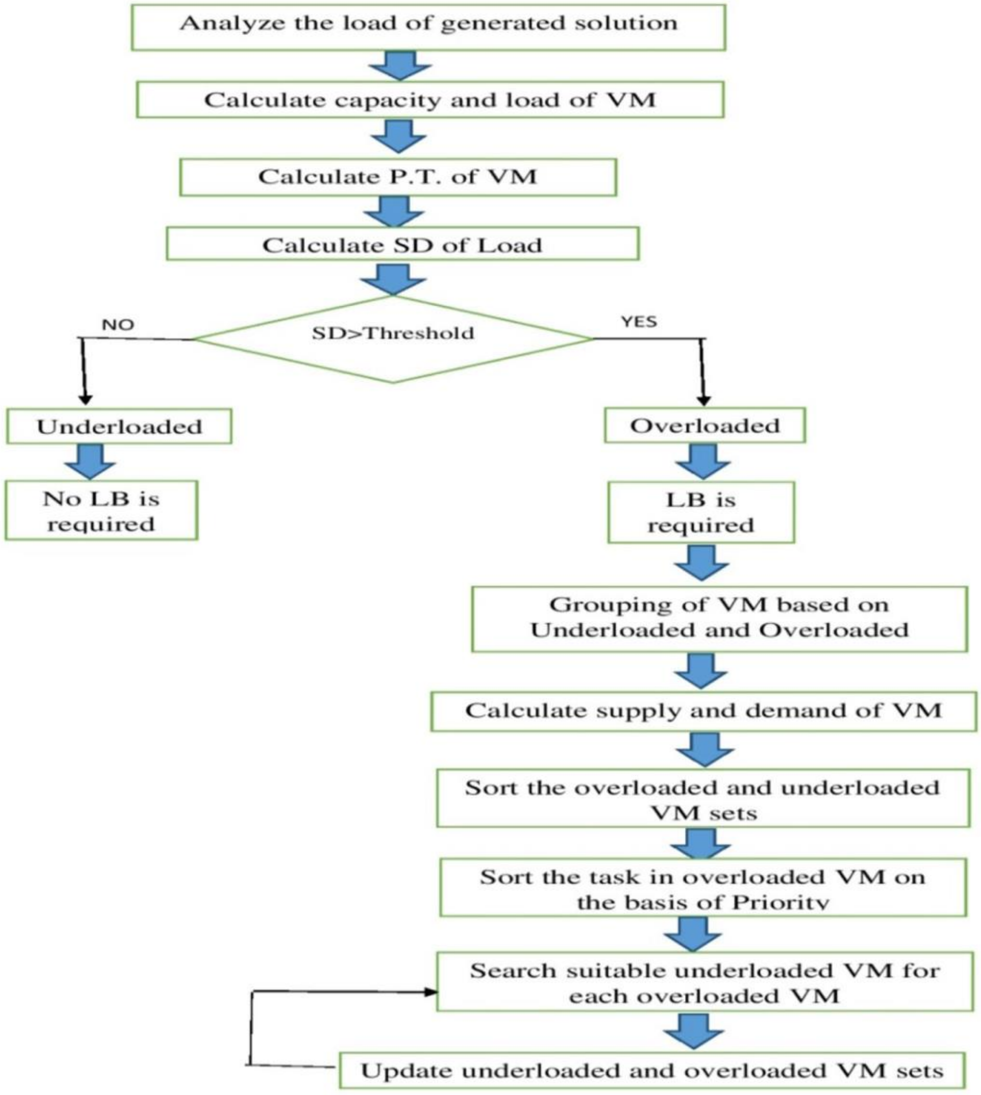

Analyze the load of generated solution
Calculate capacity and load of VM
Calculate P.T. of VM
Calculate SD of Load
SD>Threshold
NO
YES
Underloaded
No LB is required
Overloaded
LB is required
Grouping of VM based on Underloaded and Overloaded
Calculate supply and demand of VM
Sort the overloaded and underloaded VM sets
Sort the task in overloaded VM on the basis of Priority
Search suitable underloaded VM for each overloaded VM
Update underloaded and overloaded VM sets

Fig. 1. Flowchart of the proposed algorithm.

The amount of load on the virtual machine and the amount of load resulting from accepting a new request are considered for load balancing in virtual machines. Hence the virtual machine load is defined as Eq. (3).

$$Loadvm_j = CurrentLoadvm_j + TaskLength_j \qquad (3)$$

where $CurrentLoadVMj$ signifies the virtual machine load on the jth location of the solution (J=1, 2, ..., K). During task assignment, the standard deviation of the solution's virtual machine should be minimized for load balancing.

B. *The Steps of the Proposed Method*

Two main steps of the proposed method for task scheduling in this paper include:

- First step: GA to select the tasks and prioritize them for execution.
- Second step: using the Squirrel Search Algorithm (SSA) to map tasks to the virtual machines and their duration to reduce energy and fair load distribution.

1) Task selection for execution based on GA: First, in this section, general information is expressed about GA, and then the use of this algorithm to select the best task is explained. John Holland invented the main idea of GA from the evolutional theory of Darvin in 1967. Generally, GAs includes the following parts:

- Chromosome: Chromosomes in GAs show points in the search area and possible solutions to the considered problem. The number of genes (variables) on each chromosome (solution) is constant. Binary coding (binary strings) is used to present the chromosomes. A chromosome in this research shows a list of assigned resources for each task.
- Population: A population includes a set of chromosomes. A new population is generated with the same count of chromosomes using the impact of genetic operators on the population.

- Fitness function: First, a fitness function is provided to solve a problem using GAs. The fitness function in this research is based on the last task completion time duration, meaning that it is considered from the start time of the tasks to the last task completion in a parallel manner.
- Selection operator: This operator reproduces some chromosomes among the existing chromosomes in a population. Fitter chromosomes are more likely to reproduce. Elitist Selection is used in this research.
- Crossover operator: The crossover operator generates a new pair of chromosomes from a pair of chromosomes from the productive generation. Uniform crossover is used in this paper, and a random matrix is generated, namely a mask including 0 and 1 and the same length as the existing chromosomes. The mask chromosome determines which genes are transferred to the child from the first parent and which one from the second parent.
- Mutation operator: A mutation operator is applied to the chromosomes after crossover. This operator changes the content of a gene by randomly selecting an operator of a chromosome's gene.

Mapping the tasks of the application workflow to the distributed resources may have many objectives. The focus of this research is on minimizing the sum of the calculation time of the application workflow. The parallel workflow allows each task to have subtasks, and the subtasks are distributed among different resources in order to minimize the total completion time. so that each task can have some subtasks, and the subtasks are distributed among different resources to minimize the total time of the project completion. This system has two main parts:

- Task: it is the work performed in the cloud environment based on the user's request. Each task also is divided into some subtasks.
- Resource: each service in the cloud environment can assign one or some virtual machine and web services to each resource. These resources may have different processor powers and perform the service in different time durations and costs. A cloud computing system faces the challenge of selecting the resources for each task with the least amount of time and cost.

2) Machine selection by the SSA: Squirrel search is a memetic metaheuristic algorithm to find the optimal global solution via heuristic functions. This algorithm is based on memes evolution carried by interactive people and the global exchange of information among the population. In the SSA, the squirrels are transformed due to memetic evolution. In this algorithm, the squirrels are considered the hosts for the memes and are presented as a memetic vector. Each meme includes memo types showing a feature on the chromosome, like genes in GA. The squirrels can exchange their information and correct their memes. The amount of each squirrel search is adjusted by the memes improvement, and each squirrel's position is changed. SSA combines deterministic and random approaches. The deterministic approach makes it possible to use the response-level information efficiently to direct the heuristic search and the random components guarantee the flexibility and strength of the search.

The squirrel search is started with the primary population of P squirrels that are generated randomly from the problem area of Ω. In the Di-dimensional problems, the position of the ith squirrel is presented as (xi1, xi2, …, xiD). Then the merit of each squirrel is calculated based on its position, and the squirrels are sorted decreasingly based on their merits. In the next step, the total population is divided into m groups. This division is performed so that each group includes n squirrels (P=m×n). During the division process, the first squirrel is located in the first group, the second one in the second group, the mth one in the mth group, and the (m+1)th one in the first group again. The squirrels with the best and the worst merit values are presented as Xb and Xw in each group, respectively. Moreover, the squirrels with the best merit among the population are presented as Xg. Then using an evolutional process, the worst existing squirrels' merit on each cycle of the algorithm is corrected.

3) Selecting the best machine by SSA: In this section, SSA is used to execute the tasks globally. In this method, each squirrel is considered a response to the problem, and the squirrels are distributed randomly. There are some sets with an equal number of squirrels. In order to assign tasks to virtual machines, three main measures should be considered, namely the task size, the machine processing power, and makespan.

The input tasks and the virtual machines are presented as $TaskLsit=\{t1,t2,\ldots,tn\}$ and $VM=\{vm1,vm2,\ldots,vmm\}$ respectively. The squirrel hybrid mutation evolutional approach maps the tasks to the local virtual machine. The algorithm steps are presented in the following.

4) Generating the First Generation: Like other evolutional algorithms, the primary population is generated randomly. In the proposed method, each virtual machine is considered a squirrel to perform the tasks. In each repetition of the algorithm, it tries to schedule K tasks by M virtual machines. The processing capacity of the virtual machines can affect the optimal scheduling of tasks to the virtual machines. Before assigning the squirrels to the sets, the fitting function value of each squirrel should be calculated using Eq. (4).

$$F = \frac{\text{Processing\_power}}{makespan} \qquad (4)$$

This fitting function is calculated based on the machine processor and makespan. The lower the makespan, the better situation the machine has. Hence, the above equation results in the highest fitting function value for the most powerful machines.

After calculating the fitting function for all the squirrels' populations, they are sorted decreasingly, and there is a list of empty sets. The total population of the squirrels is divided into M sets. The division is performed so that each set has N squirrels. For the division, the first squirrel belongs to the first set, the second one belongs to the second set, the Mth one belongs to the Mth set, and the (M+1)th belongs to the first set again. It is repeated until the last squirrel. Each M sets include N squirrels. Since the squirrels are sorted decreasingly based on the fitness function, the first and the last assigned squirrels to the set are the best and the worst solutions, respectively. Hence, the order of entering the squirrels into the sets is important. Locality and makespan criteria are considered to find the best answer by the squirrel algorithm, which are explained in the following. The processing capacity of each virtual machine is calculated using Eq. (5):

$$\text{Processing\_power} = \text{powerj} \times \text{PCount} \quad (5)$$

where power is the processing power of the virtual machine and Pcount is the number of empty processors. The execution time of each task on the virtual machine is estimated by Eq. (6):

$$\text{ExeTime} = \text{TaskTime} \times \text{Processing\_power} \quad (6)$$

where TaskTime is the size of the task which wants to be executed. The execution time of each virtual machine is different. Less execution time of a task on the virtual machine makes less makespan on the machine. In order to accomplish this, the following algorithm is applied.

The worst squirrel's location in each set of the local search based on the fitting function is improved according to the best answer location on that set or even the best answer of all sets. Hence the average of the squirrel fitting increases. The following algorithm is used for this aim:

- Step 1: the best and the worst squirrels of each set based on the fitting value are called $Xbest$ and $Xworst$, respectively.
- Step 2: the worst squirrel of each set ($Xworst$) tries to improve itself by exchanging its information rather than the best squirrel ($Xbest$). In order to reduce the makespan value achieved when all virtual machines are processing the same amount of data, the following improvement is performed.
- Step 3: two $Xbest$ and $Xw\quad rst$ squirrels are selected so that their fitting function has the most difference, and this value should apply to all. Thus, the number of tasks of Xworst is transferred to Xbest. This transfer is performed until both fitting functions are equal.
- Step 4: after duplication of these two values, the list of squirrels in the set is sorted again. This process is performed for the next Xbest and Xworst.
- Step 5: this process is continued until the fitting function value of all squirrels is equal to the set fitting function average value.
- Step 6: all the sets are combined and sorted based on the fitting value, decreasingly, after internal evolution in each set. Then they are divided into some sets, and the evolution continues until the stop condition.

Usually, the stop condition of the algorithm is selected based on the constant variations of the best answer fitting or the algorithm repetition up to a determined number. In this problem, the considered stop condition is the determined value, $globalit$.

IV. Simulation

The proposed algorithm for task scheduling is implemented using Cloudsim. Moreover, the proposed method is simulated on the San Diego dataset. The San Diego dataset is a widely used benchmark dataset for task scheduling simulations. By using Cloudsim to simulate the proposed algorithm on the San Diego dataset, it allows researchers to compare their results with the existing literature on task scheduling and measure the performance of their proposed algorithm. This section compares the proposed method with [9] and [8] methods based on comparable criteria, including makespan, energy consumption, and execution time. This comparison allows for a clear assessment of the relative merits of the proposed method compared to the existing literature, highlighting the advantages in terms of performance and energy efficiency. Makespan determines the maximum time that each machine is active. If the distributions are not fair, this criterion is for different machines. It is the maximum time of the machine that works more than all other machines. The less general average of this criterion makes the better performance of the scheduling algorithm.

The proposed method is compared in the first experiment with the method presented in [35]. According to the results obtained in this experiment, the proposed method showed better results with regard to makespan, as shown in Fig. 1. This is because the proposed method is able to more efficiently distribute the tasks among the different machines, resulting in a lower makespan. Additionally, the proposed method is able to better account for the different capabilities of the machines, leading to a more even distribution of the tasks and better machine utilization. The proposed method is compared in the second experiment with the method presented in [34]. HPC2N and NASA workloads were used in this experiment to evaluate the proposed method. As shown in Fig. 2 and Fig. 3, the proposed method outperforms previous approaches regarding makespan time. The proposed method is able to better utilize the machines by accounting for the differences in the machines' capabilities. This means that it can better distribute the tasks to the machines, leading to a shorter makespan time, as seen in the results of the second experiment.

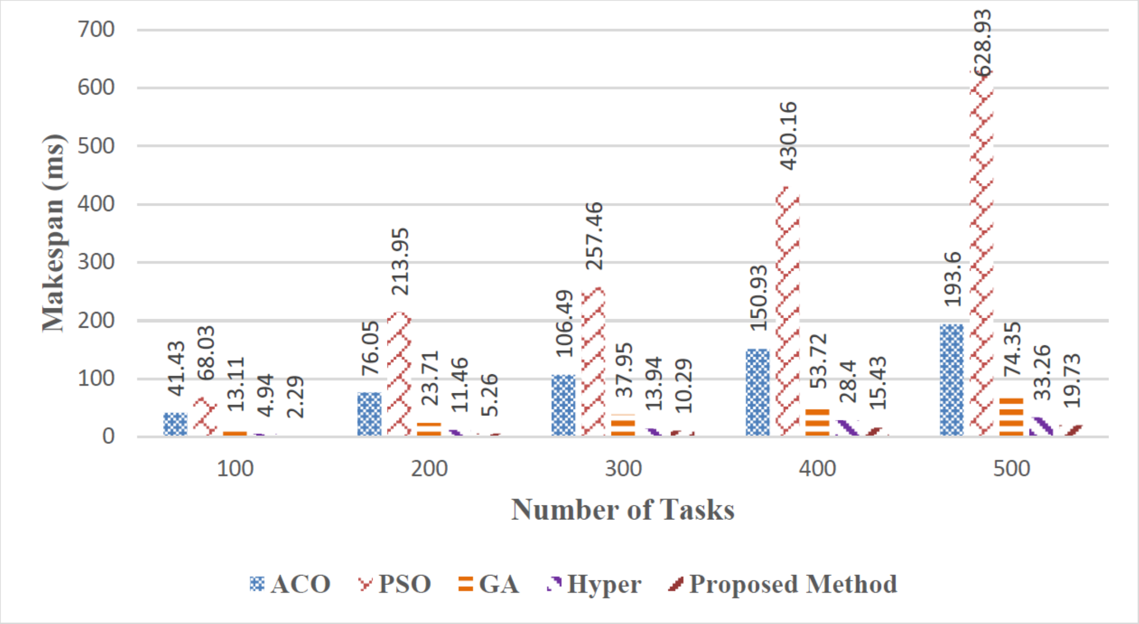


Fig. 2. Makespan comparison

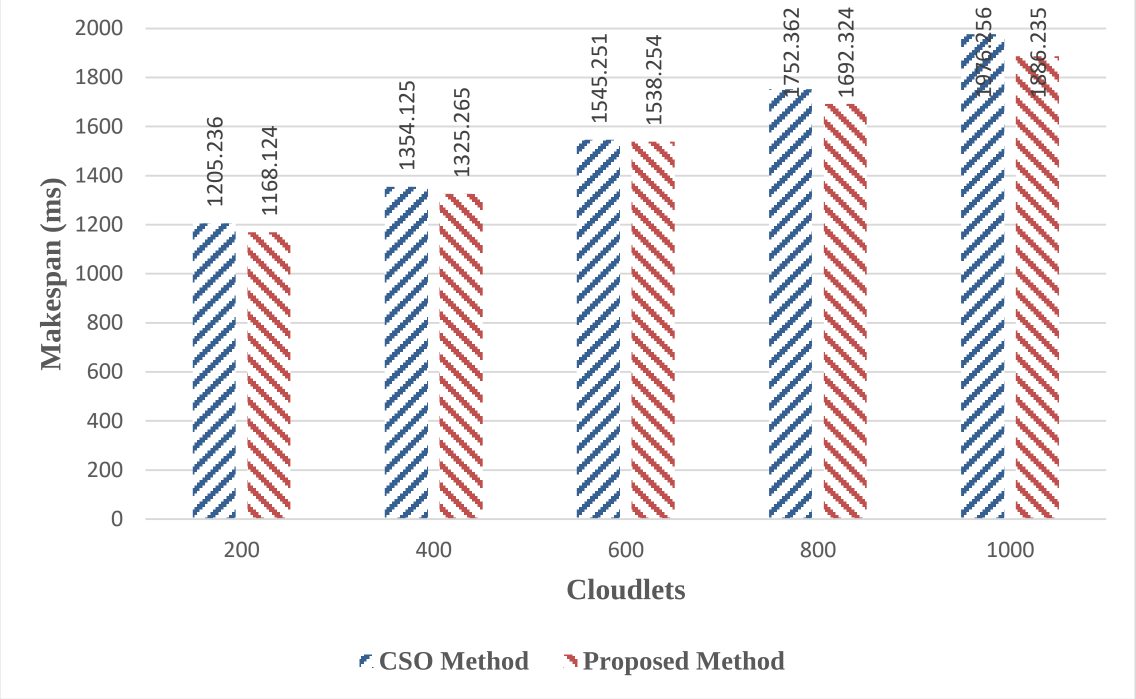


Fig. 3. Makespan comparison based on HP2CN workload.

Energy consumption criterion shows the amount of consumed energy for the execution of the tasks on the machines simulated in two different scenarios. In the first scenario, the number of machines is constant, and the number of tasks increases in each step. It is assumed that each task unit consumes one energy unit. In this scenario, the proposed method's performance is better. The cause of the increasing trend of the consumed energy diagram is the constant number of machines and the increasing number of tasks in each step. Execution time is the average time the tasks reach the resources. The less time, the better the scheduling algorithm. In the third experiment, the energy consumption and execution time of the proposed method is compared according to the data of the article [36]. This experiment uses five physical machines, 20 virtual machines, and 50 to 400 tasks. The obtained results are shown in Fig. 4. In the fourth experiment, the power consumption and execution time are evaluated based [37] dataset. Four physical machines and 5 - 50 virtual machines are used in this experiment. The obtained results are shown in Fig. 5 to 7.

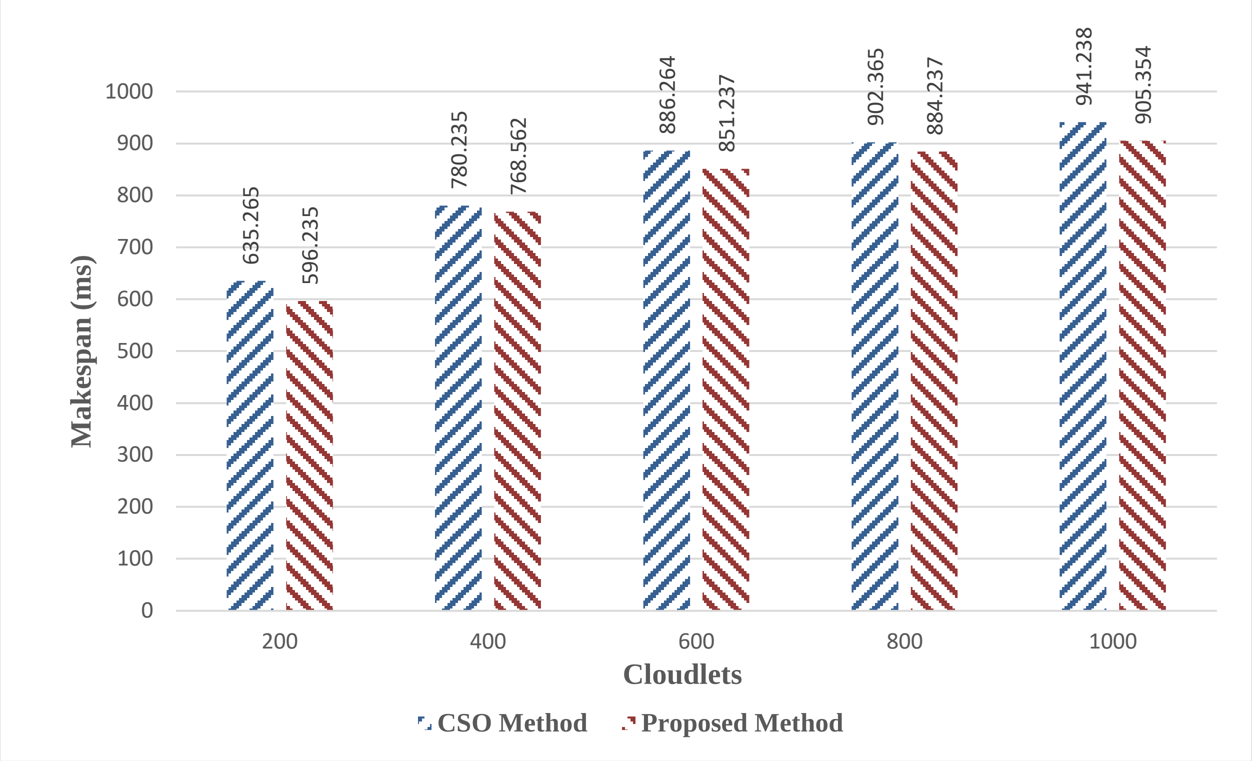


Fig. 4. Makespan comparison based on NASA workload.

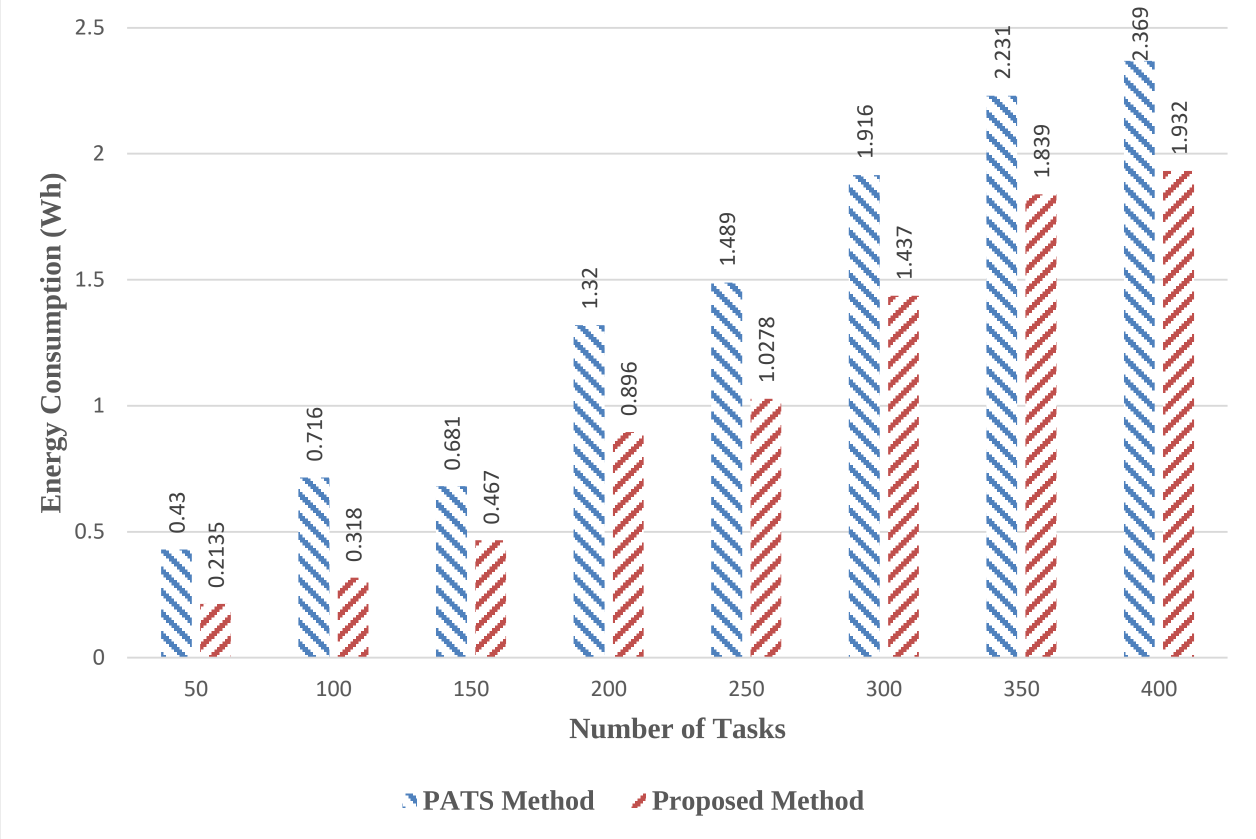


Fig. 5. Energy consumption comparison.

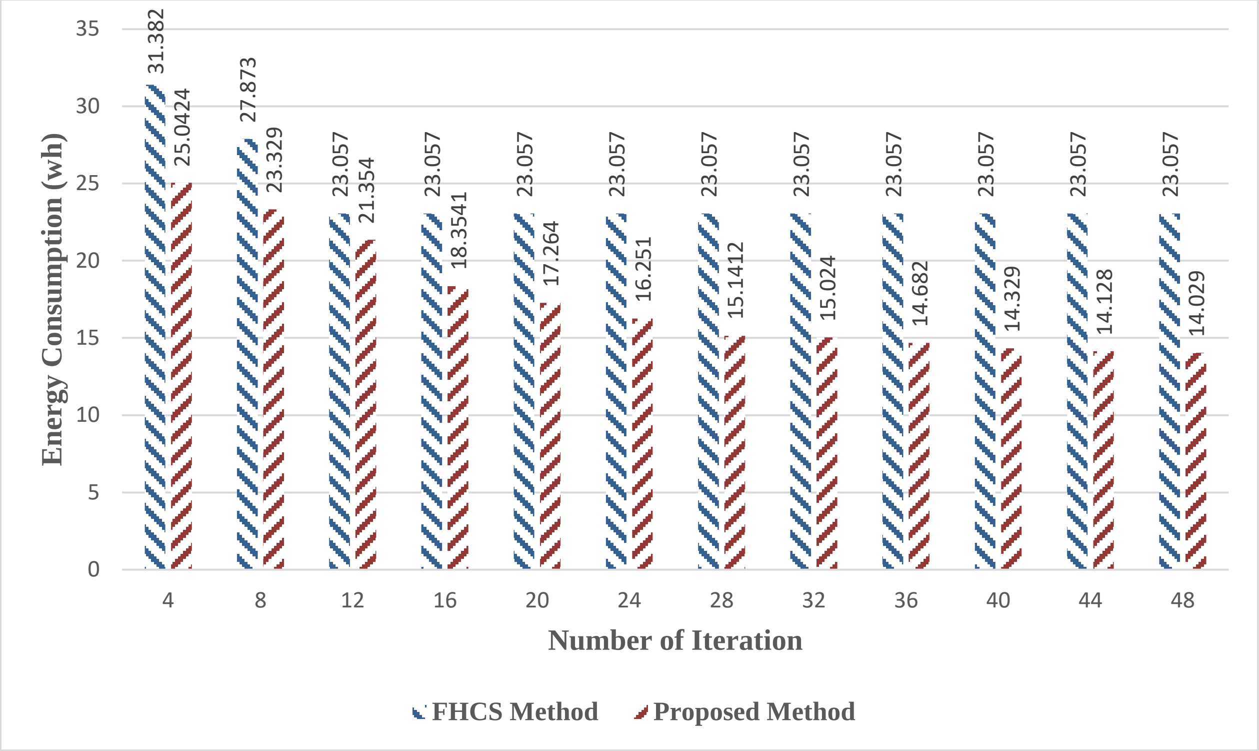


Fig. 6. Energy consumption comparison vs. iteration.

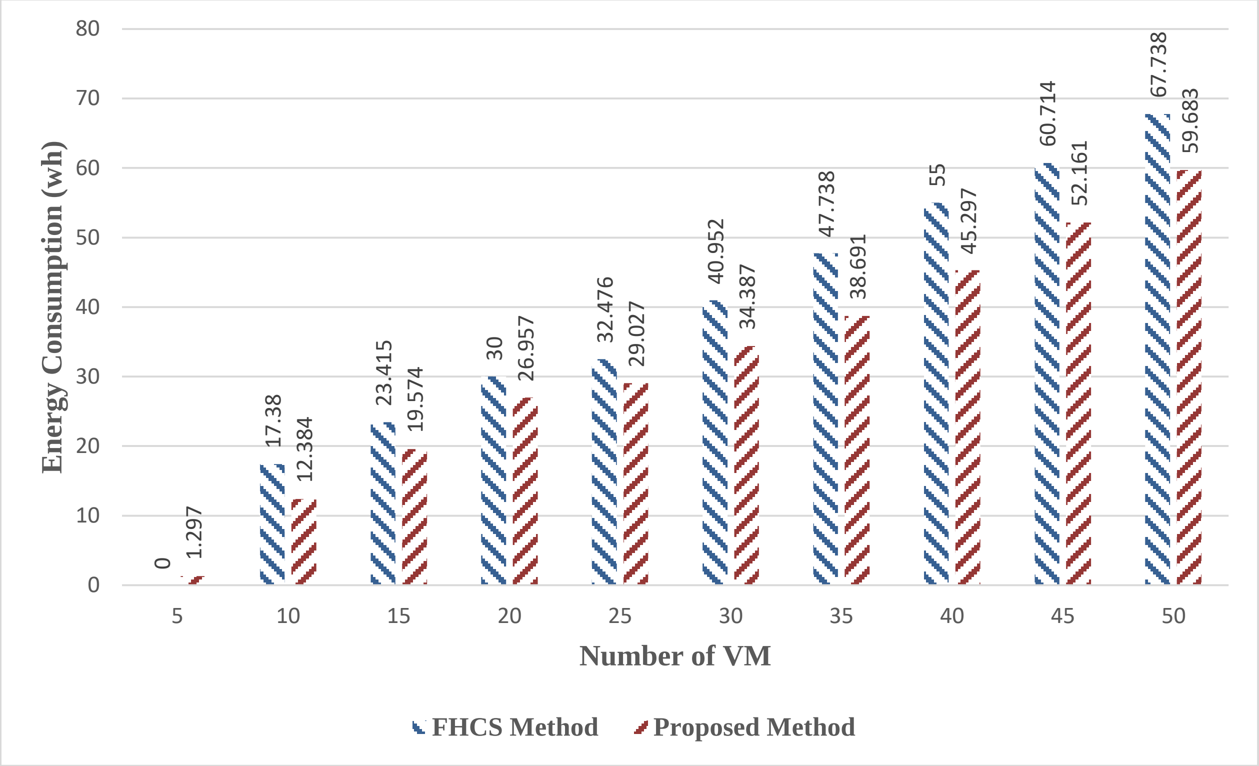


Fig. 7. Energy consumption comparison vs. number of VMs.

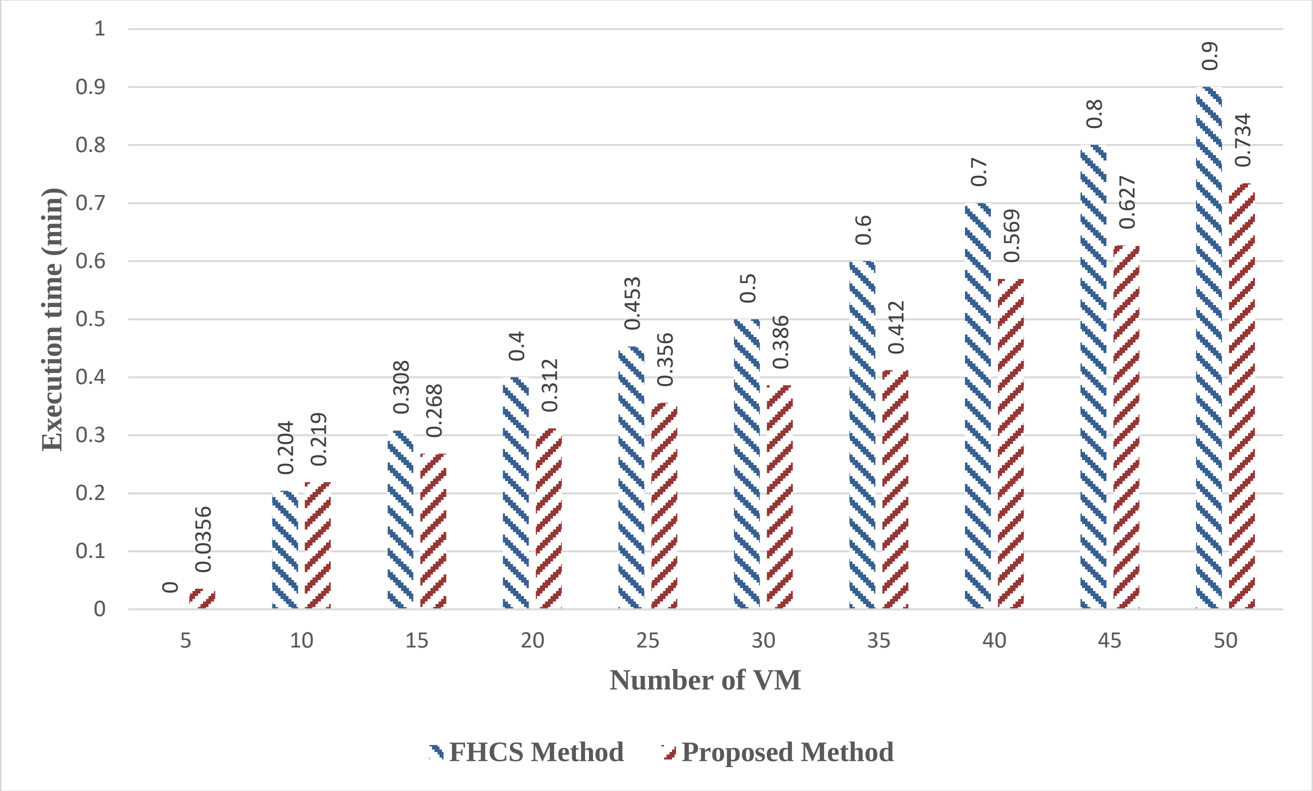


Fig. 8. Execution time comparison.

V. Conclusion

In this paper, by applying efficient scheduling to virtual machines, the efficiency of the system is enhanced, resulting in a shorter response time. It makes quick calculations and reduces energy consumption. This problem aims to apply an efficient scheduling method on virtual machines on a cloud system to meet all operational requests, and each performance criterion is optimized. Hence, metaheuristic algorithms are used. This algorithm is used by mathematical modeling of the political-social evolutional process to solve many optimization problems. This optimization evolutional strategy performance in convergence rate and reaching the global optimal is very high. While integrating multiple meta-heuristic methods may provide a hybrid heuristic with good performance, some meta-heuristics are not complementary, so combining them may not improve or even degrade performance. Performance is also affected by the integration strategy. In order to improve the performance of distributed systems on a wide range of aspects, we will study the complementarity of multiple meta-heuristics and develop an efficient integration strategy for the hybrid of multiple meta-heuristics.

References

[1] S. Seyedi and B. Pourghebleh, "A new design for 4-bit RCA using Quantum Cellular Automata Technology," Optical and Quantum Electronics, vol. 55, no. 1, p. 11, 2023.

[2] S. Seyedi, B. Pourghebleh, and N. Jafari Navimipour, "A new coplanar design of a 4-bit ripple carry adder based on quantum-dot cellular automata technology," IET Circuits, Devices & Systems, vol. 16, no. 1, pp. 64-70, 2022.

[3] J. Akhavan and S. Manoochehri, "Sensory data fusion using machine learning methods for in-situ defect registration in additive manufacturing: a review," in 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2022: IEEE, pp. 1-10.

[4] T. Taami, S. Krug, and M. O'Nils, "Experimental characterization of latency in distributed iot systems with cloud fog offloading," in 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), 2019: IEEE, pp. 1-4.

[5] P. He, N. Almasifar, A. Mehbodniya, D. Javaheri, and J. L. Webber, "Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review," Sustainable Computing: Informatics and Systems, vol. 36, p. 100822, 2022.

[6] I. Ataie, T. Taami, S. Azizi, M. Mainuddin, and D. Schwartz, "D 2 FO: Distributed Dynamic Offloading Mechanism for Time-Sensitive Tasks in Fog-Cloud IoT-based Systems," in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC), 2022: IEEE, pp. 360-366.

[7] S. Meisami, M. Beheshti-Atashgah, and M. R. Aref, "Using Blockchain to Achieve Decentralized Privacy In IoT Healthcare," arXiv preprint arXiv:2109.14812, 2021.

[8] F. Vahedifard, S. Hassani, A. Afrasiabi, and A. M. Esfe, "Artificial intelligence for radiomics; diagnostic biomarkers for neuro-oncology," World Journal of Advanced Research and Reviews, vol. 14, no. 3, pp. 304-310, 2022.

[9] S. A. Saeidi, F. Fallah, S. Barmaki, and H. Farbeh, "A novel neuromorphic processors realization of spiking deep reinforcement learning for portfolio management," in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2022: IEEE, pp. 68-71.

[10] S. H. Haghshenas, M. A. Hasnat, and M. Naeini, "A Temporal Graph Neural Network for Cyber Attack Detection and Localization in Smart Grids," arXiv preprint arXiv:2212.03390, 2022.

[11] A. Mehbodniya, J. L. Webber, R. Neware, F. Arslan, R. V. Pamba, and M. Shabaz, "Modified Lamport Merkle Digital Signature blockchain framework for authentication of internet of things healthcare data," Expert Systems, vol. 39, no. 10, p. e12978, 2022.

[12] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023.

[13] F. Khosravi, M. Tarhani, S. Kurle, and M. Shadaram, "Implementation of an Elastic Reconfigurable Optical Add/Drop Multiplexer based on Subcarriers for Application in Optical Multichannel Networks," in 2022 International Conference on Electronics, Information, and Communication (ICEIC), 2022: IEEE, pp. 1-4.

[14] F. Khosravi, G. Mahdiraji, M. Mokhtar, A. Abas, and M. Mahdi, "Improving the performance of three level code division multiplexing using the optimization of signal level spacing," Optik, vol. 125, no. 18, pp. 5037-5040, 2014.

[15] I. Attiya, M. Abd Elaziz, L. Abualigah, T. N. Nguyen, and A. A. Abd El-Latif, "An improved hybrid swarm intelligence for scheduling iot application tasks in the cloud," IEEE Transactions on Industrial Informatics, 2022.

[16] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective Task Scheduling in cloud-fog computing using goal programming approach," Cluster Computing, vol. 25, no. 1, pp. 141-165, 2022.

[17] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy-efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, p. e6959, 2022.

[18] S. M. Johnson, "Optimal two-and three-stage production schedules with setup times included," Naval research logistics quarterly, vol. 1, no. 1, pp. 61-68, 1954.

[19] B. Sellami, A. Hakiri, S. B. Yahia, and P. Berthou, "Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network," Computer Networks, vol. 210, p. 108957, 2022.

[20] A. Kumar et al., "Optimal cluster head selection for energy efficient wireless sensor network using hybrid competitive swarm optimization and harmony search algorithm," Sustainable Energy Technologies and Assessments, vol. 52, p. 102243, 2022.

[21] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic," Peer-to-Peer Networking and Applications, pp. 1-21, 2022.

[22] W. Shu, K. Cai, and N. N. Xiong, "Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing," Future Generation Computer Systems, vol. 124, pp. 12-20, 2021.

[23] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[24] L. Abualigah and A. Diabat, "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments," Cluster Computing, pp. 1-19, 2020.

[25] E. Taillard, "Some efficient heuristic methods for the flow shop sequencing problem," European journal of Operational research, vol. 47, no. 1, pp. 65-74, 1990.

[26] X. Chen et al., "A WOA-based optimization approach for task scheduling in cloud computing systems," IEEE Systems Journal, vol. 14, no. 3, pp. 3117-3128, 2020.

[27] A. Amini Motlagh, A. Movaghar, and A. M. Rahmani, "Task scheduling mechanisms in cloud computing: A systematic review," International Journal of Communication Systems, vol. 33, no. 6, p. e4302, 2020.

[28] M. Hosseinzadeh, M. Y. Ghafour, H. K. Hama, B. Vo, and A. Khoshnevis, "Multi-objective task and workflow scheduling approaches in cloud computing: a comprehensive review," Journal of Grid Computing, pp. 1-30, 2020.

[29] M. Soualhia, F. Khomh, and S. Tahar, "Task scheduling in big data platforms: a systematic literature review," Journal of Systems and Software, vol. 134, pp. 170-189, 2017.

[30] X. Wu, M. Deng, R. Zhang, B. Zeng, and S. Zhou, "A task scheduling algorithm based on QoS-driven in cloud computing," Procedia Computer Science, vol. 17, pp. 1162-1169, 2013.

[31] H. Emami, "Cloud task scheduling using enhanced sunflower optimization algorithm," ICT Express, vol. 8, no. 1, pp. 97-100, 2022.

[32] J. Yang, B. Jiang, Z. Lv, and K.-K. R. Choo, "A task scheduling algorithm considering game theory designed for energy management in cloud computing," Future Generation computer systems, vol. 105, pp. 985-992, 2020.

[33] M. Abd Elaziz, S. Xiong, K. Jayasena, and L. Li, "Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution," Knowledge-Based Systems, vol. 169, pp. 39-52, 2019.

[34] S. Mangalampalli, S. K. Swain, and V. K. Mangalampalli, "Multi Objective Task Scheduling in Cloud Computing Using Cat Swarm Optimization Algorithm," Arabian Journal for Science and Engineering, vol. 47, no. 2, pp. 1821-1830, 2022.

[35] A. Gupta, H. S. Bhadauria, and A. Singh, "Load balancing based hyper heuristic algorithm for cloud task scheduling," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 6, pp. 5845-5852, 2021.

[36] H. Zhao, G. Qi, Q. Wang, J. Wang, P. Yang, and L. Qiao, "Energy-efficient task scheduling for heterogeneous cloud computing systems," in 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019: IEEE, pp. 952-959.

[37] B. B. Naik, D. Singh, and A. B. Samaddar, "FHCS: Hybridised optimisation for virtual machine migration and task scheduling in cloud data center," IET Communications, vol. 14, no. 12, pp. 1942-1948, 2020.

Bisayan Dialect Short-time Fourier Transform Audio Recognition System using Convolutional and Recurrent Neural Network

Patrick D. Cerna, Rhodessa J. Cascaro, Khian Orland S. Juan, Bon Jovi C. Montes, Aldrei O. Caballero
College of Computer and Information Science, MAPUA Malayan College Mindanao, Davao City, Philippines

***Abstract*—Speech is a form of oral communication that reinforces thoughts and ideas that have general purpose and meaning. In the Philippines, Filipinos can speak at least three languages. English, Filipino, native language. The Philippine government says the Philippines has more than 150 regional native languages, one of which he says is Cebuano. This research aims to implement automatic speech recognition (ASR) specifically for the Bisayan dialect, and researchers use machine learning techniques to create and operate the system. ASR has served its purpose in recent years not only in the official language of the Philippines, but also in various foreign languages. The required datasets were collected throughout the study to train and build the models selected for the speech recognition engine. Audio files are recorded in waveform file format and contain Visayan phrases and sentences. Audio was captured through hours of recorded audio and process using Tensorflow short time Fourier transform (STFT) algorithm to ensure the accurate representation. In order to analyze the audio data, the recordings were specially converted to digital format, specifically .wav and making it sure all records are uncorrupted with only one channel, and finally have a sample rate of 22050kHz. A data mining process was carried out by integrating CNN layers, dense layers, and RNNs to predict the transcription of speech input using multiple layers that determine the output of the speech data. The researchers used the JiWER Python library, which was used in parallel when evaluating WER. This is because the trained scripted data set contains at least 500 time recordings totaling 61.78 minutes. Overall, the WER output is at best 99.53% and the percentage of records used is acceptable.**

***Keywords*—Bisayan dialect; speech recognition; dense layer; CNN; RNN**

I. INTRODUCTION

Automatic speech recognition (ASR) systems use algorithms to convert spoken words into text. Companies like Google Cloud, Microsoft Azure, IBM Watson, and YouTube use ASR systems for education, transcription, and disability support. All of these can provide high quality transcripts. A study showed that YouTube was able to transcribe with a word error rate of 28% compared to manual transcription with a word error rate of 17.4% [1]. However, for this study, we only used high quality FLAC files with little background noise. Poor recording quality and high background noise will result in a high word error rate. Since video data is resistant to audio noise, it can be used for speech recognition. Automatic Speech Recognition (ASR), also known as Automatic Speech Recognition, is the process of converting speech signals into text. The text can be in the form of words, phrases, syllables, or any subword unit [2]. ASR is a subset of Natural Language Processing (NLP) in the knowledge domain of artificial intelligence (AI) and aims to make communication between humans and computers as natural as possible. Its ultimate goal is to make human-computer communication indistinguishable from human-human conversation. A deep neural network (DNN) is a neural network with two or more levels of complexity that uses mathematical modeling to process data, and was used in speech recognition in 2010. DNNs have had great success with automatic speech recognition. However, due to differences in speaking styles and attitudes, models that can account for small changes or perturbations in the feature space lead to overfitting and poor generalization, which is desirable [3].

Accent can be defined as a style of pronunciation in a language. In a Filipino setup, a speaker's accent can be heavily influenced by other speakers in their approximate geographic location. This allows people to speak the same language with different accents, resulting in languages (such as Filipino) that are used with multiple accents. Extensive analysis of this provides information on speaker status, age, gender, dialect, and ethnicity [4]. Recently, the value of recognizing a speaker's accent has begun to be noticed in the field of computers. Its influence has been recognized as a basis for the development of various large-scale speech applications such as automatic speech recognition [5]. However, automatic detection of accents is a challenging research topic because languages can have multiple pronunciation styles. Automatic accent detection (also known as accent identification) is based on the consistency of acoustic patterns that can be identified in speaking styles to identify pronunciations in the same accent cluster.

Previous studies on ASR used hidden Markovs, but some focused only on vowel recognition in Tagalog [6]. Another study by Fajardo et al. [7] Automatic Filipino speech recognition is done using a convolutional neural network (CNN) model with SqueezeNet architecture for Filipino. Meanwhile, E developed continuous speech recognition for Bicol and Kapampangan using the CMU Sphinx Toolkit. The speech corpus was collected by the researcher and consists of seven hours of recordings from 150 native speakers of Kapampangan and Bicolano [8]. A hardware-based speech recognition system [9] is built with circuits incorporating neural networks, including passive and active filters to drive microphones, input/output ports, EEPROM memory, and other components. Advantages of this type of system include speed, accuracy, and lower cost than software-based speech recognition products.

Cebuano is a native language of Cebu City, of which the Visayan dialect is part. This has been extended not only to the Visayas but also to many places in Mindanao due to its growth and usefulness. There was no formal training in the Visayan dialect during the author's primary and secondary education. However, it is learned through the life stages and experiences of the community and relatives. When the K-12 Enhanced Basic Education was signed into law in the Philippines in 2013, native language subjects were included as part of the language classes for kindergarten and primary school. This gave children insight, depth of knowledge, and understanding of their mother tongue, as well as exposure to English and Filipino from an early age. Just as the Visayan dialect has evolved over the years, like other native languages that have been taught in primary school for almost a decade, these languages may one day gain the upper hand for some reason.

However, not all languages are being supported by this technology due to a very large amount of languages existing in our world, and only a few people are working on it. Thus, the researchers were able to implement Automatic Speech Recognition (ASR) in the Bisaya dialect. By doing so, the researchers would be able to provide a new path for the Bisaya dialect in the implementation of technologies alongside the expanding community in our country. In addition, the researchers would also be able to promote the use of their language in technologies, thus making it popular among the people to reduce the risk of being lost in the future. However, not all languages are being supported by this technology due to a very large amount of languages existing in our world, and only a few people are working on it. Thus, the researchers are planning to incorporate Automatic Speech Recognition (ASR) in the Bisaya dialect. By doing so, the researchers would be able to provide a new path for the Bisaya dialect in the implementation of technologies alongside the expanding community in our country. In addition, the researchers would also be able to promote the use of their language in technologies, thus making it popular among the people to reduce the risk of being lost in the future.

II. Related Works

Raval and Gajjar [10] conducted a study in an effort towards filling the gap between differently-abled people like deaf and dumb and the other people. The obtained results after extracting background were used for forming data that contained 24 alphabets of the English language. The Convolutional Neural Network proposed here is tested on both a custom-made dataset and also with real-time hand gestures performed by people of different skin tones. The accuracy obtained by the proposed algorithm is 83%. To develop a system that can read and interpret a sign like Amrutha and Prabu [11], one must train it using a large dataset and the best algorithm. As a basic SLR system, an isolated recognition model is developed. The model is based on vision-based isolated hand gesture detection and recognition. Assessment of ML-based SLR model was conducted with the help of four candidates under a controlled environment. The model made use of a convex hull for feature extraction and KNN for classification that yielded 65% accuracy.

On the other hand, Adithya and Rajesh [12] presents an efficient convolutional neural network (CNN) based model for automatically recognizing fingerspellings in sign languages. The model has been tested on a novel Indian sign language (ISL) fingerspelling dataset as well as a publicly available hand posture dataset, and has obtained promising results. Similary, Qin et al. [13] construct a lightweight sign language translation network. We construct the dataset called CSL_BS (Chinese Sign Language-Bank and Station) and two-way VTN to train isolated sign language and compares it with I3D (Inflated three Dimension). Then I3D and VTN are respectively used as feature extraction modules to extract the features of continuous sign language sequences, which are used as the input of the continuous sign language translation decoding network (seq2seq). Based on CSL-BS, two-way VTN achieves 87.9% accuracy while two-way I3D is 84.2%. Finally, Suardi et al. [14] created a trial of combining CNN models using the Ensemble method has been successfully carried out with the results being able to increase the accuracy value to 99.4%. and proved that using Ensemble can increase the higher accuracy value.

In the Philipines, Bautista and Yoon-Joong [15] describe the development of speech recognition using the Hidden Markov Model Toolkit (HTK) in Filipino only. Modifications were made to some datasets to remove unwanted background noise that is not needed for speech recognition. Therefore, removing these noises may improve the model's performance. A variety of experiments with the models are specified in the author's study, and the accuracies can be compared to conclude which model is the most effective to use in the study. Finally, Laguna and Guevara [16] used a language identification (LID) system in their approach. LID can recognize languages of unknown languages, Tagalog (TGL), Cebuano (CEB), Hiligaynon (HIL), Kapampangan (KAP), Bicolano (BCL), Warai (WAR), Tausugu (TSG) based on their research. Among these languages were pairwise and hierarchical LIDs, yielding average accuracies of 48.07% for seven languages, 72.64% for pairwise and 53.99% for hierarchical. The researchers say the LID system works best for a small number of target languages that are closely related to Filipino.

III. Materials and Methods

A. Research Design

The study intends to implement and create automatic speech recognition for Bisayan dialect. To be more specific, the study will produce an output of the transcribed text from the input audio. But, to do so, the researchers had to collect the necessary datasets that will be used throughout the study to train and build the selected models for the speech recognition engine. Researchers adapted the Knowledge Discovery in Databases (KDD) framework to manipulate data in their research. This data mining (DM) framework was adopted for the research, which consists of six distinct phases as shown in Fig. 1, based on the research goal. This will help researchers identify and collect valuable datasets for developing their own neural network models to achieve her Bisayan speech recognition. Therefore, researchers also use various techniques to help create and develop useful datasets for training models.

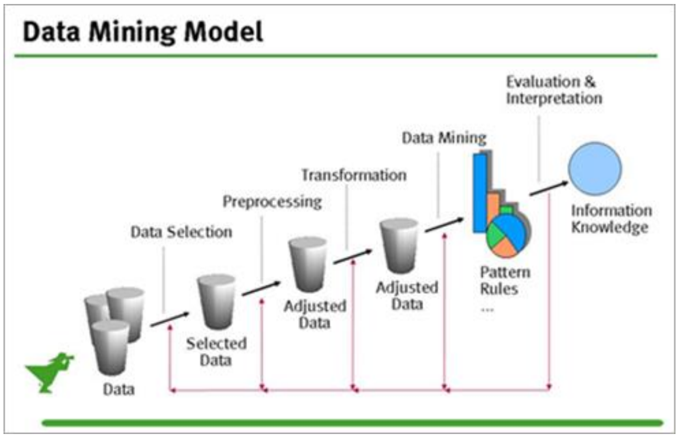


Fig. 1. KDD framework [17].

B. Data Sources and Selection

This study selects research participants who are fluent in the Visayan dialect of the Davao area and collects the required datasets through voice recordings. This helps the authors collect real, authentic data to use in their experiments and analysis throughout the study. These datasets consist of audio files containing unedited recordings of participants' natural voices. This is the raw audio output from the recording device where data collection starts with voice and text data. Audio files are recorded in waveform file format and contain Visayan phrases and sentences. When capturing audio data, hours of recorded audio must be captured to ensure the process is more accurate. During text data collection, everything is controlled and thoroughly researched by researchers to ensure quality and accurate data for research. A phrase or sentence script is provided by the researcher for text data collection (Table I). All these collections are handled according to requirements model and evaluation. So before the actual audio data processing he needs a powerful GPU such as the Nvidia RTX series to process large datasets efficiently and the latest generation of his GTX series performs well I can do it.

Researchers collected speech recognition data from public and private premises with the consent of the participants in Davao City and Mati City, as shown in Fig. 2. Respondents were provided with a script containing random Bisaya phrases of hers. The list of phrases in the Visayan dialect contains hundreds of sentences structured as scripted phrases. Some of these phrases are informal conversations by researchers affiliated with privacy and other illegal matters, but they do not infringe. The recorded phrases are long and short as they provide random topics and fiction for conversation. As shown in Table I, some examples of the following Visayan dialect phrases are read by participating speakers.

C. Methods of Data Processing

The refine input recording audio from raspberry pi powered device which contains waveform file format and this will contain Bisayan phrases or sentences. The captured audio were processed for training and testing through Tensorflow STFT algorithm will use to analyze, synthesize, transform and describe audio signals and JiWER plugins fo Similarity measures. PRAAT application tool for speech/voice feature extraction will also be utilized as the need arises.

Fig. 2. Audio recording of bisayan speakers.

TABLE I. LIST OF BISAYAN SENTENCES

English	Visayan / Cebuano
How are you?	Kamusta Ka?(Ka-mu-sta ka)
Do you speak english?	Kabolo ka mo sultiogenglish? (ka-ba-lo ka mo salty ogenglish?)
A little	Gamay ra (ga-m(eye) ra)
I speak a little Visayan	Makasulti ko gamay (ma-ka-salty ko ga-may)
I don't understand	Wala ko sabot (wa-la ko sa-bot)
Please speak slowly	Hinaya lang ogstorya (Hi-n(eye)-yah lang og story-ya)
What is you name?	Unsay imongngalan? (Un-s(eye)ee-mongnga-lan)
Thank you	Salamat (Sa-la-mat)
Where can I buy… (item)?	Asa ko makapalit ani (item) e.g shoes (ah-sa ko ma-ka-pa-lit ah-ni)
I would like to buy souvenirs	Ganahan ko mopalit pasalubong (Ga-na-han ko mo-pa-lit pa-sa-loo-bong)
Where is…(place)?	Asa ang lugar (place) e.g CR (term used for bathroom) (ah-sa ang loo-gar)
What places can one visit here?	Unsay ma suroyansainyonglugar? (Oon-s(eye) ma soo-roy-yansaee-mong loo-gar)
How much?	Tag pila? (tag-pi-la)
Good Morning	MayongBuntag (m(eye)-yong bun-tug)
Good Afternoon	MayongHapon (m(eye)-yong hah-pon)
Good Evening	MayongGabii (m(eye)-yong gah-bee-e)
See You Later	Kita ta unya (ki-ta ta oon-ya)
Yes	O-o (oh oh)
No	Dili (Di-li)
What?	Unsa? (oon-sa?)
When?	Kan-usa? (kan-oosa?)
Please	Palihug (Pa-lee-hoog)

The CNN Layer, Dense Layer, and RNN will be used is to predict the transcription of the audio input using several layers that would determine the output of the audio. It is expected that the output of this research will then evaluate the accuracy of the speech recognition by using Word Error Rate (WER), accuracy, precision and recall. Fig. 3 shows the separate steps of a process in sequential order, this is known as Flowchart. The process will start by processing the dataset and getting their audio and text data, which will be used for the training of the model. The audio data shall be used as the input data for the model, while the textual data shall be used as the target data of the model's prediction. While in training, the model shall compare its prediction to the target data, and shall evaluate its errors to modify its weight parameters. During testing, it shall compare its prediction to the target data so that it could compute the Word Error Rate of its output. Thus, the value of the Word Error Rate will serve as its evaluation in transcribing Bisaya Dialect.

To analyze the audio data, the recording was converted into a digital format specifically using audacity in particular into a .wav format. According to Frosch [18], digital audio is presented in many formats, and one of them can also compress audio which is termed as the lossy audio compression format. Lossy sound files compression attempts to reduce the amount of data. In simpler meaning, it allows for even more file size reductions by eliminating certain audio information and simplifying the data. At the same time, the frequency has audio features that are also audio signals that can be used to develop statistical or Machine Learning models. Audio files for this study will contain unedited recordings in a lossy format which will significantly benefit for data compression of the study. All audio recorded was set by 22,050 Hz of Project Rate and Mono Recording Channel to lessen file size and let the model train faster. If some of the recordings do not follow the rule or forgot to set, a code in the Programming section will automatically do it for convenience as shown in Fig. 4.

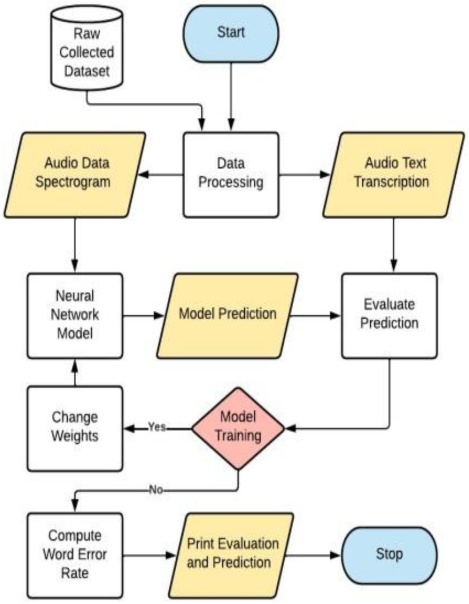


Fig. 3. Flowchart of dialect recognition system.

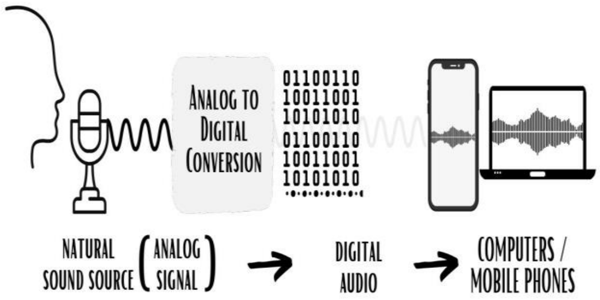


Fig. 4. Analog to digital conversion.

IV. Data Experimentation and Analysis

A. Audio Conversion to Datasets

After gathering the necessary data for the experimentation, the researchers had to optimize and clean the collected data. Thus, the researchers utilized Python packages to easily optimize and clean their audio dataset so that it would be used for the training of the model. The process was done to ensure that all of their datasets are not corrupted, must be in a .wav format, must not be over 10 seconds, should only have 1 channel, and finally, it should have a sample rate of 22050 kHz. If this would not be followed, it would affect the efficiency of the training process, which would further burden the researchers to do their experimentations. Based on Table II, the researchers had gathered a total amount of 68.76 minutes of audio data that consists of 563 recordings. Each recording contains one phrase out of the 500 phrases that were prepared by the researchers to the participants. The list of created phrases or sentences can be found in Bisaya Dialect Phrases.

TABLE II. Datasets Gathered after the Conversion

Group Dataset	No. of Times (mins)	No. of Files
Collected Datasets	68.76	563
Train Datasets	61.78	506
Test Datasets	6.97	57
Augmented Datasets	123.57	1012
Train + Augmented	185.35	1518
Overall	192.33	1575

After optimizing and cleaning the audio datasets, they would be separated into two folders which will be called "train_wav" and "test_wav". These folders shall be the basis for the train datasets and test datasets of the experimentation. After processing the required audio data, the researchers had to create a CSV file for the train and test datasets that would be used to fit the model as shown in Fig. 5. There would be two CSV files that would be generated which are called "train.csv" and "test.csv". These CSVs shall contain the file directory of the audios while also having the text transcription of it. In order for the neural network model (CNN and RNN) to train, the researchers had to extract the audio data and its transcription so that it would be given to the model. The necessary data that needs to be processed is the input and target data for the model. By getting the input data, the researchers use a Python package that is capable of reading the signal from the audio and transforming it into input data for the model. For the target data, the characters in the transcription of the audio shall be converted into numbers which will be the target data that the model should have after computing the input data as shown in Fig. 5.

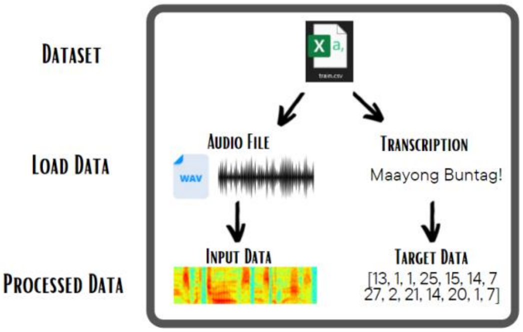


Fig. 5. Audio conversion to target data.

B. Prediction using CNN, RNN and Dense Layer

The CNN Layer, Dense Layer, and RNN were used to predict the transcription of the audio input using several layers that would determine the output of the audio. Based on the figure below, the structure of the neural network model used in this study contains several layers as shown in Fig. 6. The layers within the models consist of several layers. Furthermore, the researchers will use some other underlying layers that are used to optimize the training performance of the model, such as dropouts and normalizations. With this kind of network model structure, the researchers believe that it will be able to perform its purpose and provide results from the given input audio. The researcher will then evaluate the accuracy of the speech recognition by using Word Error Rate (WER), accuracy, precision and recall.

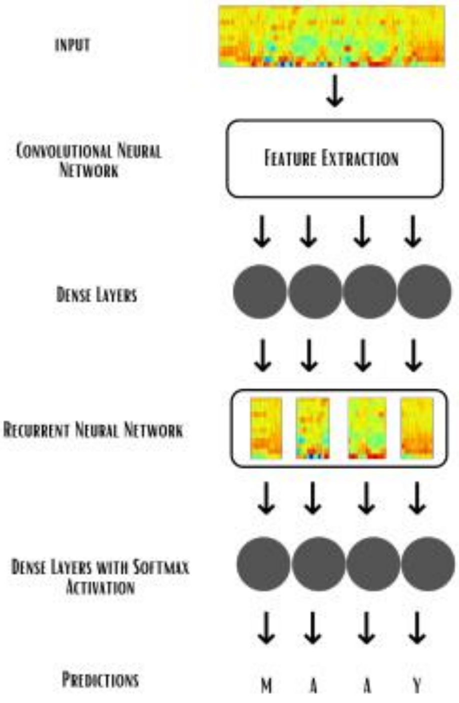


Fig. 6. Prediction of recognition using CNN, RNN and dense layer.

C. Result of Data Analysis

Based on the results of experimentations as reflected in Table III, the researchers believed that adding augmented data would provide better model predictions. Furthermore, having additional RNN layers would also provide positive training results but would make the model predictions perform slower than having a few layers. Thus, depending on the target development and dataset would determine how many layers would be necessary to have better results in the training and deployment. However, since the researchers had only a few datasets to train the model, they had to provide alternative solutions based.

TABLE III. RESULT OF AUDIO RECOGNITION USING CNN AND RNN

Row Number	Audio Dataset	RNN Layers	Optimal Epochs	Word Error Rate
1	11.95 minutes (conversation)	2 layers	100 Epochs	99.53 %
2	11.97 minutes (scripted)	2 layers	92 Epochs	96.37 %
3	11.95 minutes + 11.97 + Noise	2 layers	64 Epochs	98.48 %
4	39.13	5 layers	81 Epochs	76.79 %
5	61.78 Minutes + 123.57 minutes (Augmented)	5 layers	30 Epochs	66.82 %

Furthermore, having additional RNN layers would also provide positive training results but would make the model predictions perform slower than having a few layers. Thus, depending on the target development and dataset would determine how many layers would be necessary to have better results in the training and deployment. However, since the researchers had only a few datasets to train the model, they had to provide alternative based on their experimentations shown in the table above. In the Word Error Rate (WER), this is the standard evaluation of how accurate an Automatic Speech Recognition (ASR) system is. It calculates the number of found "errors" in an ASR transcription text. This has been used by different researchers and big companies worldwide to also measure and identify the accuracy of their machine learning. In this study, the researchers used JiWER Python Library and is used alongside in evaluating WER. Since the trained scripted datasets contains at least 500 and a total of 61.78 minutes' time recording. The WER output shows 99.53% (as reflected in Table III) at best which results to acceptable percentage for the number of datasets used.

V. CONCLUSION AND RECOMMENDATIONS

Overall, the study was able to create and train a neural network model that would be used for Speech Recognition System in Bisaya Dialect. The result of the data experimentation reveals the best results of analysis through 99.53% Word Error Rate for their trained model. The researcher therefore recommends that it would be best to seek ways to address the biggest limitation of the study which is to acquire a large amount of dataset for the training of the model. By doing so, the trained model would be able to have a better prediction and performance, since it has been exposed to a large amount of different data for its training. Aside from that, others should also be thoughtful when it comes to their hardware specifications, as the training of the model would require a heavy amount of processing power which would put a lot of stress onto your hardware. Thus, it would be recommended to have better hardware when training a neural network model with a very large amount of datasets, so that it would lessen the amount of time to train the model.

REFERENCES

[1] J. L. Bautista and Y. Kim, "[PDF] An Automatic Speech Recognition for the Filipino Language using the HTK System | Semantic Scholar," *[PDF] An Automatic Speech Recognition for the Filipino Language using the HTK System | Semantic Scholar*, Jan. 01, 2019. https://www.semanticscholar.org/paper/An-Automatic-Speech-Recognition-for-the-Filipino-Bautista-Kim/7d47daadfa9738461db5073afaa0f4798baed86e.

[2] D. Nagajyothi and P. Siddaiah, "Speech Recognition Using Convolutional Neural Networks," International Journal of Engineering & Technology, vol. 7, no. 4.6, p. 133, Sep. 2018, doi: 10.14419/ijet.v7i4.6.20449.

[3] D. Guiming, W. Xia, W. Guangyan, Z. Yan, and L. Dan, "Speech recognition based on convolutional neural networks," 2016 IEEE International Conference on Signal and Image Processing (ICSIP), Aug. 2016, Published, doi: 10.1109/siprocess.2016.7888355.

[4] M. Tjalve, Accent Features and Idiodictionaries. Department of Phonetics and Linguistics 2007.PhD Dissertaion. University College London.

[5] A. Mukhamadiyev, I. Khujayarov, O. Djuraev, and J. Cho, "Automatic Speech Recognition Method Based on Deep Learning Approaches for Uzbek Language," *Sensors*, vol. 22, no. 10, p. 3683, May 2022, doi: 10.3390/s22103683.

[6] A. C. Farjardo and Y.-J. Kim, "Development of Filipino Phonetically-balanced Words and phoneme-level Hmms," IJARCCE, pp. 1–6, Jan. 2015, doi: 10.17148/ijarcce.2015.4101.

[7] F. R. Jr. Arnel Fajardo, "Convolutional Neural Network for Automatic Speech Recognition of Filipino Language," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.1 S I, pp. 34–40, Feb. 2020, doi: 10.30534/ijatcse/2020/0791.12020.

[8] E. H. Liao, K. Ganareal, C. C. Paguia, C. Agreda, M. Octaviano, and R. Rodriguez, "Towards the Development of Automatic Speech Recognition for Bikol and Kapampangan," 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Nov. 2019, Published, doi: 10.1109/hnicem48295. 2019.9072783.

[9] M. A. Ruiz and C. R. Mendoza, ""Using Hardware-based Voice Recognition toInteract with a Virtual Environment", Virtual Reality Laboratory, Proceedings of the IVEVA 2004 Workshop. Intelligent Virtual Environments and Virtual Agents University of Colima, CEUPROMED, Colima, 28040,Mexico.

[10] J. J. Raval and R. Gajjar, "Real-time Sign Language Recognition using Computer Vision," *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, May 2021, Published, doi: 10.1109/icspc51351.2021.9451709.

[11] K. Amrutha and P. Prabu, "ML Based Sign Language Recognition System," *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, Feb. 2021, Published, doi: 10.1109/icitiit51526.2021.9399594.

[12] V. Adithya and R. Rajesh, "Convolutional Neural Network based Sign Language Recognition to Assist Online Assessment," *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, Sep. 2021, Published, doi: 10.1109/access51619.2021.9563346.

[13] W. Qin, X. Mei, Y. Chen, Q. Zhang, Y. Yao, and S. Hu, "Sign Language Recognition and Translation Method based on VTN," *2021 International Conference on Digital Society and Intelligent Systems (DSInS)*, Dec. 2021, Published, doi: 10.1109/dsins54396.2021.9670588.

[14] C. Suardi, A. N. Handayani, R. A. Asmara, A. P. Wibawa, L. N. Hayati, and H. Azis, "Design of Sign Language Recognition Using E-CNN," *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Apr. 2021, Published, doi: 10.1109/eiconcit50028.2021.9431877.

[15] J. L. Bautista and Y. Kim, "[PDF] An Automatic Speech Recognition for the Filipino Language using the HTK System | Semantic Scholar," *[PDF] An Automatic Speech Recognition for the Filipino Language using the HTK System | Semantic Scholar*, Jan. 01, 2020. https://www.semanticscholar.org/paper/An-Automatic-Speech-Recognition-for-the-Filipino-Bautista-Kim/7d47daadfa9738461db5073afaa0f4798baed86e.

[16] A. F. Laguna and R. C. Guevara. Development, Implementation and Testing of Language Identification System for Seven Philippine Languages. *Philippine Journal of Science* 144 (1): 81-89, June 2015 ISSN 0031 – 7683.

[17] U. Fayyad, "Knowledge discovery in databases: An overview," Inductive Logic Programming, pp. 1–16, 1997, doi: 10.1007/3540635149_30.

[18] Frosch, Hannah N., "Spectral Analysis and Comparison of Analog and Digital Recordings" 2017. Honors Theses. 545. Retrieved from: https://aquila.usm.edu/honors_theses/545/.