Disease Prediction from Symptom Descriptions Using Deep Learning and NLP Technique

Salmah Saad Al-qarni[1], Abdulmohsen Algarni[2]
Department of Informatics and Computer Systems-College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia[1]
Dept. Computer Science-College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia[2]

***Abstract*—Accurate disease prediction from symptom descriptions is vital for improving early detection and enabling remote healthcare services, especially in the evolving landscape of digital health. Traditional diagnosis methods face significant limitations due to their reliance on structured datasets and subjective assessments, leading to delays and inefficiencies in the diagnosis process. Our strategy is to employ advanced NLP techniques such as tokenization and TF-IDF, along with DL techniques like LSTM, CNN-LSTM, and GRU, to analyze unstructured symptom data and more accurately predict diseases The study also compares two text transformation techniques (TF-IDF vectorization and tokenization) with traditional Machine Learning (ML) methods like Decision Trees to specify the best technique. Through intensive experiments on two datasets (one with 24 diseases and one with 41 diseases), the efficiency of the proposed methods is verified and the importance of using NLP and deep learning in revolutionizing healthcare is illustrated, particularly in upgrading remote diagnosis and enabling early medical intervention. The best-performing model, CNN-LSTM using tokenized text, achieved 99.90% accuracy on a 41-disease dataset, and LSTM with TF-IDF achieved 98.8% accuracy on a 24-disease dataset, outperforming or matching results from more complex models in prior studies. The findings show that combining NLP and deep learning enables accurate, efficient disease prediction, advancing remote care and early intervention in digital healthcare.**

Keywords*—Natural language processing; disease prediction; machine learning; deep learning; classification

I. INTRODUCTION

Healthcare systems worldwide always face huge challenges in precisely and early diagnosing diseases, which are mainly caused by the limitations of traditional diagnostic approaches [1]. Not only do diseases impose a heavy influence on the quality of individuals' lives but also place very high economic burdens on healthcare systems. Traditional diagnostic approaches mainly rely on structured clinical data and subjective judgments, which are likely to result in subjectivity, inconsistency, errors, and delays. With a growing rate of chronic and acute illnesses, there is a critical need for novel solutions that can effectively and efficiently address these diagnosis issues. With the complexity of the diseases, the clinicians and doctors require more sophisticated and automated tools to enable timely decision-making, resource planning, and improved patient care [2].

Deep learning (DL) and Natural Language Processing (NLP) have been transformative technologies in healthcare, particularly in automating and enhancing diagnosis processes [3]. NLP allows computational systems to read, comprehend, and manipulate human language, which allows healthcare professionals to extract useful knowledge from unstructured patient data, such as symptom descriptions provided in natural language. At the same time, DL, a subset of AI that employs neural networks with many layers, has already begun to achieve great success in recognizing complex patterns and making predictions from intensive data analysis. NLP merged with DL has the potential to transform medical diagnosis by dramatically enhancing prediction accuracy, cutting down diagnostic time, and enabling remote medical decision-making [4].

Contemporary trends in digital health further reinforce the importance of NLP and DL. For example, more than 100 digital diagnostic products are now commercially available to assess disease risk and expedite diagnosis, many of which are powered by AI and machine learning. The rapid advancement of such AI-driven symptom checkers and decision support tools underscores how computational models of language can augment clinical practice by triaging patient complaints and flagging potential conditions earlier than traditional workflows. The global COVID-19 pandemic has also accelerated the adoption of telemedicine and remote monitoring solutions, where NLP plays a critical role in interpreting patient-reported symptoms in real time. These developments illustrate a broader shift towards integrating language-based AI into healthcare delivery, improving accessibility and personalization of care even outside conventional clinical settings [5].

In parallel, the accessibility of large-scale medical text data and increase in computational power have paved the way for data-driven diagnostic systems. Patients now routinely describe symptoms on online forums, chatbots, and mobile health applications, creating an abundance of unstructured symptom narratives available for analysis. Leveraging this wealth of textual data requires advanced NLP for language understanding and robust DL for pattern recognition – domains where recent techniques have demonstrated exceptional promise. By integrating NLP and DL into clinical workflows, it becomes possible to bridge the gap between traditional symptom checkers and expert diagnostic systems, facilitating faster, more personalized, and scalable diagnostic support for patients and providers alike [6].

This study builds on these trends and challenges by introducing several key contributions. It utilizes varied datasets (including one with 24 diseases and another with 41 diseases)

to improve the validity and generalizability of the predictive models. Additionally, it conducts a comparative study of multiple machine learning algorithms, incorporating two text representation techniques – TF-IDF vectorization and tokenization – to identify the most effective approach for symptom-based disease classification. Finally, the study emphasizes an iterative model refinement process to boost accuracy and reliability, ensuring that the proposed NLP+DL models are well-optimized for deployment in real healthcare settings. Through these contributions, the work illustrates how combining NLP and deep learning can enable more accurate and efficient disease prediction, thereby advancing remote care and early intervention in the evolving landscape of digital healthcare.

The proposed methodology was chosen due to its adaptability to real-world symptom expression formats and its ability to extract both semantic and sequential patterns from text. Traditional classifiers often require structured or pre-encoded symptom data and struggle with ambiguity and variability in natural language inputs. Moreover, prior methods frequently overlook interpretability, computational trade-offs, or generalization across datasets. In contrast, our framework explicitly addresses these limitations through multi-dataset validation, model transparency (via feature analysis), and an architecture that balances performance with deployment feasibility.

To validate the effectiveness of the proposed approach, the rest of this study is organized as follows: Section II explores related work in natural language processing and deep learning for disease prediction. Section III outlines the methodology, detailing the text representation methods and model architectures used. Section IV presents the results, including the datasets and a thorough analysis of the experimental findings. Section V concludes the study and suggests directions for future research.

II. Related Work

In the past several years, several studies have tried to utilize ML and DL methods for disease prediction based on symptom descriptions. Such methods usually combine natural language processing (NLP) techniques and different ML models in an effort to improve prediction accuracy and computational efficiency. Some of the research studies aim to exploit hybrid systems, preprocessing techniques, and optimization algorithms to develop more robust and effective models for disease diagnosis with potential for future digital health and remote diagnosis.

A range of studies have explored automated disease prediction using symptom descriptions, applying both traditional and deep learning techniques. One study introduced a hybrid ensemble model that combined Naïve Bayes, SVM, and XGBoost to improve the classification of categorical symptom data [7]. This model achieved 97.2% accuracy, approximately 2% lower than that of another work which focused on optimizing classical classifiers using severity-weighted symptom scores, reporting 99.19% accuracy with a tuned KNN model. It offered a lightweight and computationally efficient alternative for large-scale screening applications [8].

In contrast, a deep learning approach that applied MCN-BERT and BiLSTM to symptom text achieved the highest performance, reaching 99.58% accuracy. This showcased the advantage of contextual language models and advanced optimizers in capturing subtle patterns in unstructured clinical descriptions [9]. Another comparative study evaluated multiple classical classifiers on a structured symptom dataset and found that Random Forest outperformed other models with an accuracy of 99.5%, about 2.3% higher than earlier ensemble-based methods, confirming its strength in handling tabular symptom data [10]. Additionally, a separate model used translated symptom data and XGBoost, achieving 99.45% accuracy while maintaining relatively simple preprocessing and training requirements. This approach proved to be a practical option for early illness detection, especially in resource-limited settings [11].

Table I provides a comparison of five representative studies, highlighting their approaches, data, performance, and noted limitations. These prior works have made important strides by achieving high predictive accuracy using various NLP and AI methods, as summarized below.

As seen in Table I, prior studies in this domain report remarkably high accuracies (often above 95% and even nearing 99%), suggesting that automated symptom-based disease classifiers can perform extremely well under certain conditions. However, a notable observation is that most of these works focus on showcasing their performance results while providing relatively limited discussion of technical constraints or real-world challenges. For instance, several studies used complex ensembles or deep models but did not comment on the computational memory requirements or infrastructure needed to deploy those models at scale. The interpretability of the models is another aspect often glossed over – methods involving transformers or ensemble boosting are essentially black boxes from a clinician's perspective – yet only one study explicitly acknowledged the issue of explaining predictions. Moreover, generalizability is frequently under-addressed: many models were trained and tested on a single dataset (sometimes a synthetic or idealized one), raising questions about how they would perform on different patient populations or symptom inputs. In cases, where a non-clinical or limited dataset was used, authors occasionally noted the need for more diverse data, but they generally stopped short of testing their models beyond the original setting. This pattern of presenting near-perfect accuracy without proportional attention to practicality can make the proposed solutions seem overly ideal. It creates an impression that the problem of automated diagnosis is "solved" in a controlled environment, even though issues of model robustness, scalability, and integration into medical practice remain open challenges.

TABLE I. COMPARISON OF RELATED WORK

Authors	Year	Techniques Used	Dataset	Accuracy
Indupalli & Pradeepini	2023	Hybrid blended stacking ensemble (CatBoost + XGBoost + Naïve Bayes)	Symptom-based disease dataset with categorical symptom inputs (post-COVID context; multiple diseases, categorical symptoms)	97.2%
Fuster-Palà et al.	2024	Optimized classical ML classifiers (SVM, Random Forest, KNN, ANN) with symptom severity encoding.	Public symptom dataset containing 41 diseases with associated symptoms encoded by severity scores (Open-source structured dataset for disease screening).	99.19%
Hassan et al.	2024	Transformer-based language model (MCN-BERT) and deep neural network (BiLSTM) for symptom text classification.	Two datasets: (1) Dataset-1: 1,200 patient symptom description entries (each a unique combination of a disease label and its described symptoms); (2) Dataset-2: 23,516 Twitter posts for adverse drug reaction detection (tweets labeled ADR or non-ADR).	99.58% on Dataset-1 96.15% on Dataset-2
Das et al.	2022	Comparative analysis of four supervised ML classifiers: k-Nearest Neighbors, Naïve Bayes, Decision Tree, and Random Forest.	Structured disease-symptom dataset with 41 prevalent diseases and 132 possible symptoms. Each case in the data is characterized by a selection of 5 key symptoms out of the 132, representing a patient's input features. (This dataset is a common benchmark for symptom-based disease prediction.)	99.5%
Abidin et al.	2022	Machine learning classification using Naïve Bayes, Linear Discriminant Analysis (LDA), and an ensemble XGBoost model.	Symptom dataset (translated into English) covering 41 diseases and 17 selected clinical symptoms. The translation step ensured uniform input language for the model. This smaller feature set (17 symptoms) was used to train and evaluate the classifiers on disease identification tasks.	99.45%

Our work distinguishes itself by explicitly addressing many of these limitations. First, we prioritize a model design that is computationally efficient and feasible for deployment. Instead of relying on extremely large language models or complex stacked ensembles, we implement a CNN-LSTM architecture—a powerful deep learning model that is faster to train than transformer-based approaches. This choice means our best model achieves comparable accuracy to prior works (e.g., ~99% on a 41-disease set) without the need for extraordinary computational resources or cloud infrastructure. Such efficiency is crucial for real-world use, particularly in resource-constrained healthcare settings. Second, we emphasize generalizability and robustness by evaluating our approach on two different datasets (one with 24 diseases and another with 41 diseases). By demonstrating consistent performance across multiple datasets, we provide evidence that the model can handle variations in disease profiles and symptom expressions. This is what make this method better than earlier studies that were often validated on a single dataset, and it gives greater confidence that our method could be adapted to new data or scaled to broader disease coverage. Third, while our models are predominantly deep neural networks, we have incorporated interpretable elements into the experimentation.

For example, we included a Decision Tree classifier in our comparisons and performed feature representation analyses (TF-IDF versus tokenization) to understand the impact of different input processing techniques. This approach offers insights into the model's behavior – such as which symptoms or words are most influential – thereby injecting a degree of interpretability and transparency into the results that pure black-box approaches lack. Finally, our study consciously discusses practical deployment considerations by iteratively refining the models and comparing complexity versus performance trade-offs, we avoid simply reporting an accuracy number in isolation. In summary, the proposed model aims not only to achieve high accuracy, but also to deliver a solution that is balanced and realistic – one that minimizes memory overhead, maintains interpretability where possible, and generalizes better to varied inputs. By addressing these often-overlooked aspects, our work contributes a more practically viable tool for disease prediction from textual symptom descriptions, advancing the field towards implementations that can be trusted and adopted in modern healthcare workflows.

III. METHODOLOGY

Accurate disease prediction from natural language symptom descriptions poses a challenge due to the unstructured nature of the input text and the variability in how symptoms are expressed. Traditional approaches often struggle to extract meaningful features from such data, affecting prediction performance and reliability.

To address this, our methodology focuses on transforming symptom descriptions into structured representations using two popular text vectorization techniques: TF-IDF and Tokenization [12] [13]. In this study, a consistent workflow was applied that includes preprocessing, handling class imbalance, and training multiple models using each text representation method. Both traditional ML (Decision Tree) and DL models (LSTM, GRU, CNN-LSTM) were trained and evaluated. The process was repeated for both text representation techniques to systematically compare their impact on classification performance. This approach enables a fair assessment of which technique yields more accurate and robust predictions in the context of disease classification. The method is visualized in Fig. 1.

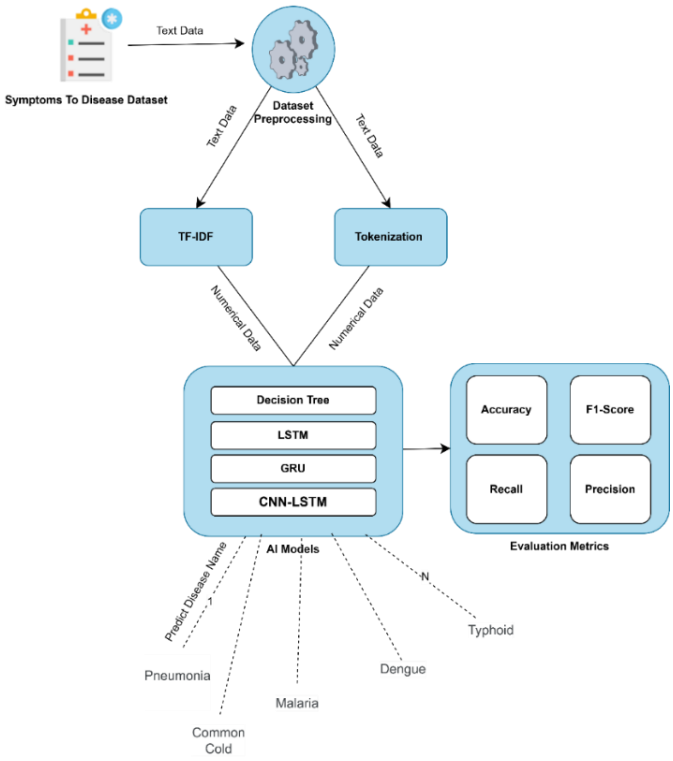

Fig. 1. The Proposed method.

A. Deep Learning Models

1) Deep learning background. A neural network is a computer model that replicates the human brain's structure in the form of layers of interconnected processing nodes (neurons). The neurons transform information through weights and activation functions, which allow the network to learn patterns and make predictions. A 1D Convolutional Neural Network (CNN) is a neural network that takes sequential data as input with convolutional filters in one dimension and is highly appropriate for tasks such as time series analysis, text analysis, or speech analysis [14]. An RNN (Recurrent Neural Network) is another neural network that takes data sequentially and deals with sequences and time-dependent patterns very well [15]. Sequential data or time series can also be in text, video, speech recognition, voice recognition, time series forecasting, NLP tasks, and other mediums. Therefore, models that are specifically meant for sequential data are appropriate for applications such as disease prediction from symptom text. RNNs produce the current output using the previous information in the sequence. On the other hand, if a sequence is long, it is hard for the network to convey information from the earlier steps to the later steps. For example, in working through a paragraph of text for prediction, an RNN might forget crucial information from the start. Though RNNs possess very restricted internal calculations, they are okay for short sequences. In order to tackle the short-term memory or disappearing gradient problem in RNNs, LSTM and GRU models were proposed as alternatives. The Gated Recurrent Unit (GRU) is an RNN variant that is essentially the same as LSTM [16]. The GRU uses the hidden state to transmit information, unlike the cell state of the other models. The GRU, which trains LSTM faster since it requires fewer tensor operations, has two gates: the reset gate, similar to LSTM, that determines how much previous information to forget, and the update gate, similar to the combination of LSTM's forget and input gates, that determines what information to retain or discard [16]. The control flow of LSTM is the same as RNN, in that it takes the data and maintains the information along with it as it proceeds. The processes in the cells of LSTM differ, however, in that the cells here are meant to forget or recall [17]. LSTM contains two more gates compared to GRU: the output gate and the forget gate. The forget gate decides what to keep or leave out of the prior cell state, the input gate regulates what parts of the cell state are transmitted to the hidden state, and the output gate governs how much of the information from the prior state is transferred or left out in order to impact the next hidden state [17]. The forget gate in the LSTM cell is utilized to element-wise multiply the preceding memory cell.

2) LSTM Layer. The LSTM layer is meant to address the vanishing gradient issue of the regular RNNs. It has "gates" like the input gate, the output gate, and the forget gate to control the flow of data in the LSTM network [17]. The gates regulate how information is remembered, updated, or discarded at every time step. The LSTM updates Eq. (1-6) are as follows:

$$g_t = \phi\left(U_g \cdot [u_{t-1}, y_t] + c_g\right) \quad (1)$$

$$j_t = \phi\left(U_j \cdot [u_{t-1}, y_t] + c_j\right) \quad (2)$$

$$k_t = \phi(U_k \cdot [u_{t-1}, y_t] + c_k) \quad (3)$$

$$\tilde{S}_t = \tanh(U_S \cdot [ut - 1, y_t] + c_S) \quad (4)$$

$$S_t = g_t \odot S_{t-1} + j_t \odot \tilde{S}_t \quad (5)$$

$$u_t = k_t \odot \tanh(S_t) \quad (6)$$

In this context, ϕ refers to the sigmoid activation function. The variables g_t, j_t, and k_t correspond to the activations of the forget, input, and output gates at time step t, respectively. S_t represents the memory cell's internal state at that same time, while u_t denotes the hidden state [17]. The symbol $\odot$ indicates an element-wise multiplication operation. The notation $[u_{t-1}, y_t]$ signifies the concatenation of the hidden state from the previous time step and the current input. The parameters U_g, U_j, U_k, U_S along with the biases c_g, c_f, c_k, and c_S are trainable components of the model—comprising weight matrices and bias vectors used to compute the gate activations and update the cell state.

3) GRU Layer. The GRU layer reduces the architectural complexity of the LSTM by combining the input gate and the forget gate into one update gate, and the cell state and the hidden state. The GRU update Eq. (7-10) are:

$$m_t = \sigma(Q_m \cdot [u_{t-1}, y_t] + c_m) \quad (7)$$

$$n_t = \sigma(Q_n \cdot [u_{t-1}, y_t] + c_n) \quad (8)$$

$$\tilde{u}t = \tanh(Q \cdot [n_t \odot ut - 1, y_t] + c) \quad (9)$$

$$u_t = (1 - m_t) \odot u_{t-1} + m_t \odot \tilde{u}_t \quad (10)$$

Here, m_t is the update gate that decides how much of the state needs updating. The reset gate, $n_{t,}$ decides how much of the previous state needs to be erased. The candidate activation, $\tilde{u}t$, proposes a new potential value for the state, as decided by mt. The new state at the present timestep, ut, is computed as a weighted average of the former state and the candidate state, with the update gate mt controlling the balance.

4) CNN-LSTM. A CNN-LSTM network takes advantage of the strengths of CNN and LSTM. The CNN captures important spatial or local features of the input data, while the LSTM models are temporal dependencies in sequences. The combined model is well adapted to the task of sequential data with complex patterns, such as text classification and time series forecasting.

B. Machine Learning Model

1) Decision Tree (DT). DT are among the most widely used machine learning algorithms [18]. While they are applicable to both classification and regression tasks, they are more commonly employed for classification. A Decision Tree structures its predictive model in a hierarchical, tree-like format: internal nodes represent feature-based decision points, the root node defines the initial splitting condition, and the leaf nodes correspond to the final predicted classes or outcomes. The DT nodes are arranged in levels, and the highest or first node is the root node. Every internal node comprises tests on input variables or attributes (i.e., nodes with one or more children). The classification model descends downwards through the respective child nodes based on the test results, splitting recurring until reaching a leaf node. The leaf nodes denote the final decision results [19].

C. Evaluation Metrics

In order to assess the performance of a classifier, there are some typical measures: accuracy (most widely used measure), sensitivity (also referred to as recall in most papers), specificity, precision, and F1 score [20]. Classification outcomes for each class are either "True Positive" (TP), "True Negative" (TN), "False Positive" (FP), or "False Negative" (FN). Depending on these values, the higher-level measures are computed according to the following Eq. (11-14):

$$\text{Accuracy } = \sum_c \frac{TP_c + TN_c}{TP_c + FP_c + TN_c + FN_c}, c \in \text{ classes} \quad (11)$$

$$\text{Precision } = \sum_c \frac{TP_c}{TP_c + FP_c}, c \in \text{ classes} \quad (12)$$

$$\text{Recall } = \sum_c \frac{TP_c}{TP_c + FN_c}, c \in \text{ classes} \quad (13)$$

$$F1_{\text{score}} = 2 * \frac{\text{precision} * \text{sensitivity}}{\text{precision} + \text{sensitivity}} \quad (14)$$

Following is a description of the above metrics:

- Accuracy: The proportion of correctly classified samples out of all samples.
- Accuracy: The ratio of samples which are "True Positive" to all the samples which were predicted as "Positive".
- Recall (or Sensitivity): The ratio of accurately forecasted "True Positive" examples to all genuine positive samples.
- F1 Score: The harmonic mean of precision and recall, a single score that combines both measures and takes into account precision and sensitivity.

All of the above metrics can be used for any machine learning system. Thus, all the metrics mentioned in this section will be used to compare the classifier systems devised in this study. The performance of the classification system will also be compared with that of the earlier studies.

IV. Results

This study evaluated the effectiveness of TF-IDF versus Tokenization approaches for symptom-based disease classification across two distinct datasets. The experimental framework incorporated both traditional machine learning (Decision Tree) and deep learning architectures (LSTM, GRU, CNN-LSTM) to provide a comprehensive comparison of text representation methods. Dataset 1 contained 1,200 clinical cases spanning twenty-four diseases, while Dataset 2 comprised 4,920 instances across forty-one disease categories, offering diverse testing conditions for model generalization. Performance metrics like accuracy, precision, recall, and F1-score were systematically analyzed to determine optimal methodologies for clinical text classification tasks.

To ensure robust model evaluation, we used two distinct datasets (twenty-four and forty-one diseases respectively) rather than relying solely on internal validation splits. This cross-dataset validation allows us to test the generalizability of the models to different distributions and disease spaces. Performance was assessed using multiple metrics (accuracy, precision, recall, F1-score), and the models consistently achieved high performance across both datasets.

A. Dataset

1) Dataset 1. The first dataset comprises 1,200 data instances and two features: i) label, comprising disease labels, and ii) text, comprising symptom descriptions in natural language, illustrated in Table II. The dataset consists of twenty-four diseases, with fifty symptom descriptions for each, giving 1,200 data instances. It is concerned with the association between diseases and symptoms and comprises pairs, where each pair is a disease and a symptom. The dataset is developed by Niyarr Barman and can be found on Kaggle[1]. Data curation was done by collecting data from diverse credible sources of medical data, including research papers, medical journals, and clinical databases. The information was

[1] https://www.kaggle.com/datasets/niyarrbarman/symptom2disease

painstakingly extracted and verified to ascertain the validity of symptom-disease relationships.

TABLE II. SAMPLE DATA FROM DATASET 1

Disease	Symptom Description
Psoriasis	I've been experiencing a red, itchy rash with dry, scaly patches on my arms, legs, and trunk for several weeks.
Psoriasis	My skin is scaling, especially on my scalp, elbows, and knees, and the scaling is usually accompanied by burning or stinging sensations.
Psoriasis	I'm also having joint pain in my fingers, wrists, and knees. The pain is throbbing and achy and gets worse with motion.
Drug Reaction	My tongue also changes in taste and scent, leaving a metallic aftertaste. Can have excruciating joint and muscle pain.
Drug Reaction	I have headaches and migraines, and I have been having difficulties sleeping. I've been shaking and shivering all over. Sometimes I become lightheaded.

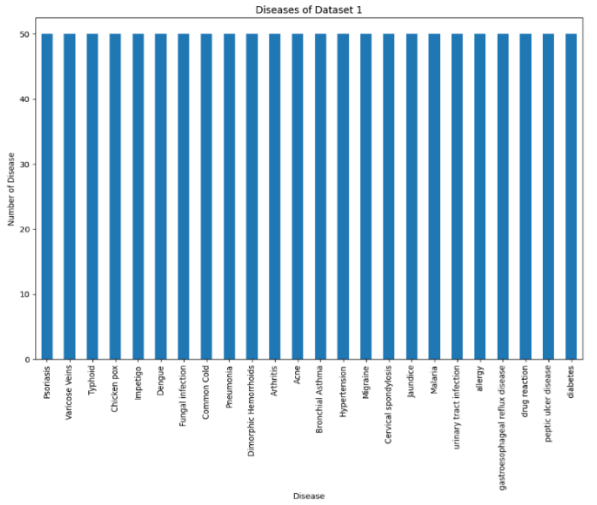

Fig. 2. Distribution of different diseases in Dataset 1.

Fig. 2 shows the distribution of different diseases in Dataset 1. The diseases are identified on the x-axis, and the y-axis indicates how many times each disease occurs in the dataset. The graph provides an idea of how the data are distributed across the various diseases, indicating how many times each disease occurs.

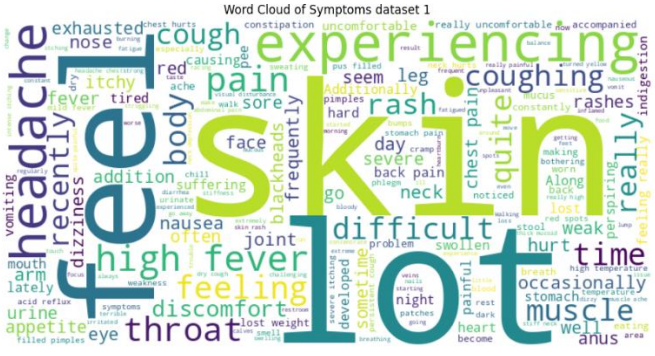

Fig. 3. Word cloud of symptoms in Dataset 1.

Fig. 3 is a word cloud representing the most prevalent symptoms in Dataset 1. The size of each symptom word is representative of its frequency of appearance in the symptom descriptions, with larger words representing more prevalent symptoms and smaller words representing less prevalent symptoms. The word cloud is a good visual display of the prevailing symptoms in the dataset, which identifies frequent patterns in the symptom descriptions.

2) Dataset 2. The dataset used for this project is the Disease Symptom Prediction[2]. The dataset has close to 5,000 entries, with each entry having a list of symptoms. Each entry is tagged with one of forty-one possible diseases. The symptoms are given in the same form as in the original file, and a separate file gives data about the severity level of each symptom from 1 to 7. Dataset 2 contains eighteen columns and 4,920 rows. Table III shows a sample data from this dataset.

TABLE III. SAMPLE DATA FROM DATASET 2

Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4
Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic_patches
Migraine	acidity	indigestion	headache	blurred_and_distorted_vision
Cervical spondylosis	back_pain	weakness_in_limbs	neck_pain	
Fungal infection	skin_rash	nodal_skin_eruptions	dischromic_patches	
Fungal infection	itching	nodal_skin_eruptions	dischromic_patches	

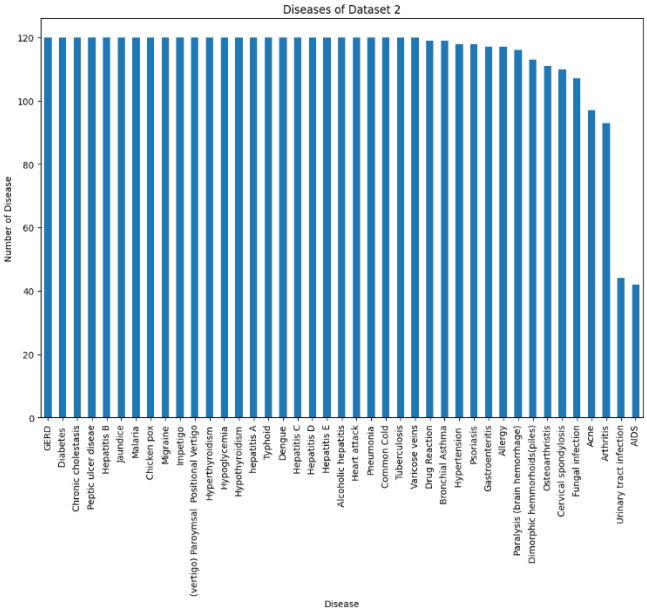

Fig. 4. Distribution of different diseases in Dataset 2.

Fig. 4 is a bar chart showing the distribution of diseases in Dataset 2. Diseases are on the x-axis and the frequency of each

[2] https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset

disease in the dataset on the y-axis. The plot provides an overview of the distribution of data across diseases, showing how frequently each disease occurs.

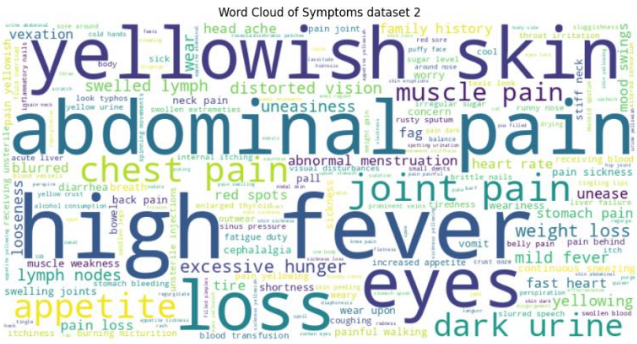

Fig. 5. Word cloud of symptoms in Dataset 2.

Fig. 5 is a word cloud that illustrates the most frequent symptoms in Dataset 2. The words are scaled based on how often they appear in the symptom descriptions, with more common symptoms shown in larger words and less common ones in smaller words.

B. Dataset Preprocessing

1) Background. In this study, two primary text representation techniques (TF-IDF and Tokenizer) were applied to compare their performance when used with both deep learning and machine learning models.

a) TF-IDF: Term Frequency-Inverse Document Frequency is a technique of text vectorization that provides a weight for each term for the improved representation of rare words in a corpus and reducing the influence of common, non-informative terms. This is helpful for the reason that irrelevant features can confuse the learning algorithm and worsen the performance. TF-IDF is obtained by multiplying TF with IDF. TF is the occurrence ratio of term k to the number of unique words n in the dataset, as indicated in Eq. (15) and Eq. (16). The IDF is the term frequency in all the documents, where Eq. (15) and Eq. (16) are the number of documents in total and the number of documents containing term k. So, common words will have a low TF-IDF score and rare words will have a high score.

$$TF = \frac{n_k}{n} \tag{15}$$

$$IDF = \log_2\left(\frac{d_n}{df_k}\right) \tag{16}$$

b) Tokenizer: Tokenization converts text into sequences of integer tokens, where each unique word is assigned a unique index. This prepares the text for deep learning models that require numerical input. For the tokenization approach, we employed a word-level tokenizer followed by an embedding layer as the first layer in each neural network (LSTM, GRU, CNN-LSTM).

2) Dataset 1 preprocessing. For Dataset 1, NLP preprocessing steps was applied including punctuation removal, stop word removal, and lemmatization. After preprocessing, both TF-IDF and Tokenizer were applied, allowing us to compare the models' performance using these two text representation approaches.

3) Dataset 2 preprocessing. Dataset 2 presented a challenge due to a large number of duplicated rows. To address this, column shuffling and automatic word replacement using WordNet synonyms was applied to enhance variation. All symptom columns were merged into a single text column. After resolving these issues, NLP preprocessing (punctuation removal, stop word removal, and lemmatization) was applied, followed by TF-IDF and Tokenizer to compare their effectiveness in representing symptoms for prediction tasks.

C. Hyperparameter Configuration

Model configurations were carefully optimized for each text representation approach. For TF-IDF implementations, 10 to 20 training epochs with batch sizes of 16 to 32 were employed, using the Adam optimizer throughout. Tokenizer-based models required more extensive training, with 30 to 70 epochs and consistent batch sizes of 16. The Tokenizer vocabulary was limited to 1,200 tokens for Dataset 1 and 5,000 for Dataset 2, with corresponding maximum sequence lengths of 24 and 30 tokens respectively. All sequential models utilized post-padding and included an out-of-vocabulary token (<OOV>) to handle unseen symptom descriptions. The Hyperparameters for Dataset 1 and Dataset 2 are shown in Table IV and Table V respectively.

TABLE IV. HYPERPARAMETERS FOR DATASET 1 (1200 INSTANCES, 24 DISEASES)

Parameter	TF-IDF	Tokenizer
Epochs	20	70
Batch Size	16	16
Optimizer	Adam	Adam
Max Words	—	1,200
Max Sequence Length	—	24
OOV Token	—	<OOV>
Padding	—	post

TABLE V. HYPERPARAMETERS FOR DATASET 2 (4920 INSTANCES, 41 DISEASES)

Parameter	TF-IDF	Tokenizer
Epochs	10	30
Batch Size	32	16
Optimizer	Adam	Adam
Max Words	—	5,000
Max Sequence Length	—	30
OOV Token	—	<OOV>
Padding	—	post

D. Performance Analysis

The results demonstrate distinct performance patterns between text representation methods. TF-IDF consistently achieved superior results with traditional models, as evidenced by the Decision Tree's 99.8% accuracy on Dataset 2, compared to just 69.0% with Tokenization. For deep learning

architectures, the advantage of TF-IDF was less pronounced, with CNN-LSTM achieving 99.9% accuracy using Tokenization on Dataset 2, suggesting that hybrid models can effectively leverage sequential patterns in symptom descriptions. Dataset 1 revealed the scalability challenges of Tokenization, where performance gaps between representation methods widened for the Dataset 1 classification task. Notably, the GRU model's accuracy dropped from 97.9% (TF-IDF) to 85.4% (Tokenizer) on this harder dataset. These findings indicate that while Tokenization can achieve excellent results with appropriate model architectures, TF-IDF remains more robust for general applications, particularly when combining multiple algorithm types or working with expanded disease taxonomies. The superior performance of CNN-LSTM across both datasets suggests that combining spatial feature extraction with sequential processing may offer the most reliable approach for clinical text classification tasks. The Performance analysis on dataset 1and Dataset 2 is represented in Table VI and Table VII respectively.

Fig. 6 and Fig. 7 visualize the comparison of performance for Dataset 1 and Dataset 2, respectively.

TABLE VI. PERFORMANCE ANALYSIS ON DATASET 1 (1200 INSTANCES, 24 DISEASES)

Model	Text Rep.	Accuracy	Precision	Recall	F1-Score
LSTM	TF-IDF	98.80%	98.90%	98.80%	98.70%
LSTM	Tokenizer	95.00%	95.60%	95.00%	94.70%
GRU	TF-IDF	97.90%	98.10%	97.90%	97.90%
GRU	Tokenizer	85.40%	86.50%	85.40%	85.20%
CNN-LSTM	Tokenizer	97.00%	97.00%	97.00%	97.00%
Decision Tree	TF-IDF	81.70%	83.10%	81.70%	81.70%
Decision Tree	Tokenizer	33.00%	32.00%	33.00%	32.00%

TABLE VII. PERFORMANCE ANALYSIS ON DATASET 2 (4920 INSTANCES, 41 DISEASES)

Model	Text Rep.	Accuracy	Precision	Recall	F1-Score
LSTM	TF-IDF	99.78%	99.81%	99.80%	99.80%
LSTM	Tokenizer	94.40%	94.00%	92.20%	93.00%
GRU	TF-IDF	97.85%	93.51%	94.78%	94.03%
GRU	Tokenizer	99.60%	99.60%	99.60%	99.60%
CNN-LSTM	Tokenizer	99.90%	99.90%	99.90%	99.90%
Decision Tree	TF-IDF	99.80%	99.80%	99.80%	99.80%
Decision Tree	Tokenizer	69.00%	70.00%	69.00%	69.00%

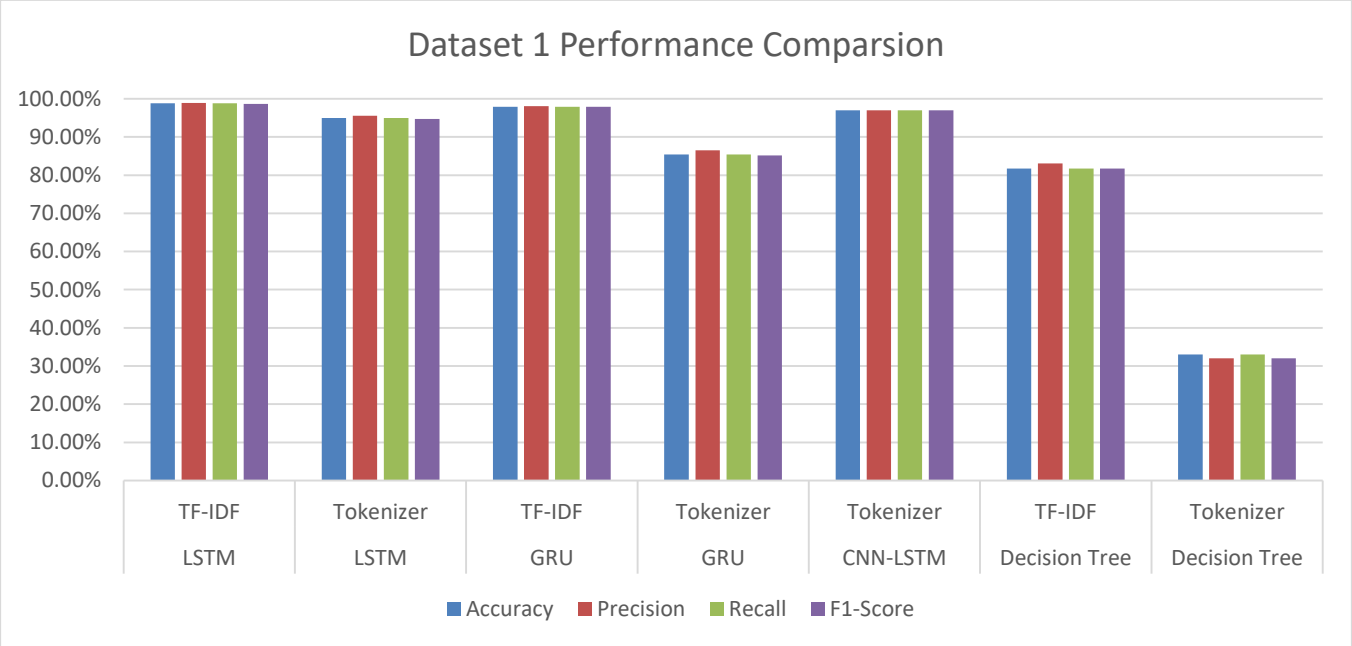

Fig. 6. Dataset 1 performance comparison.

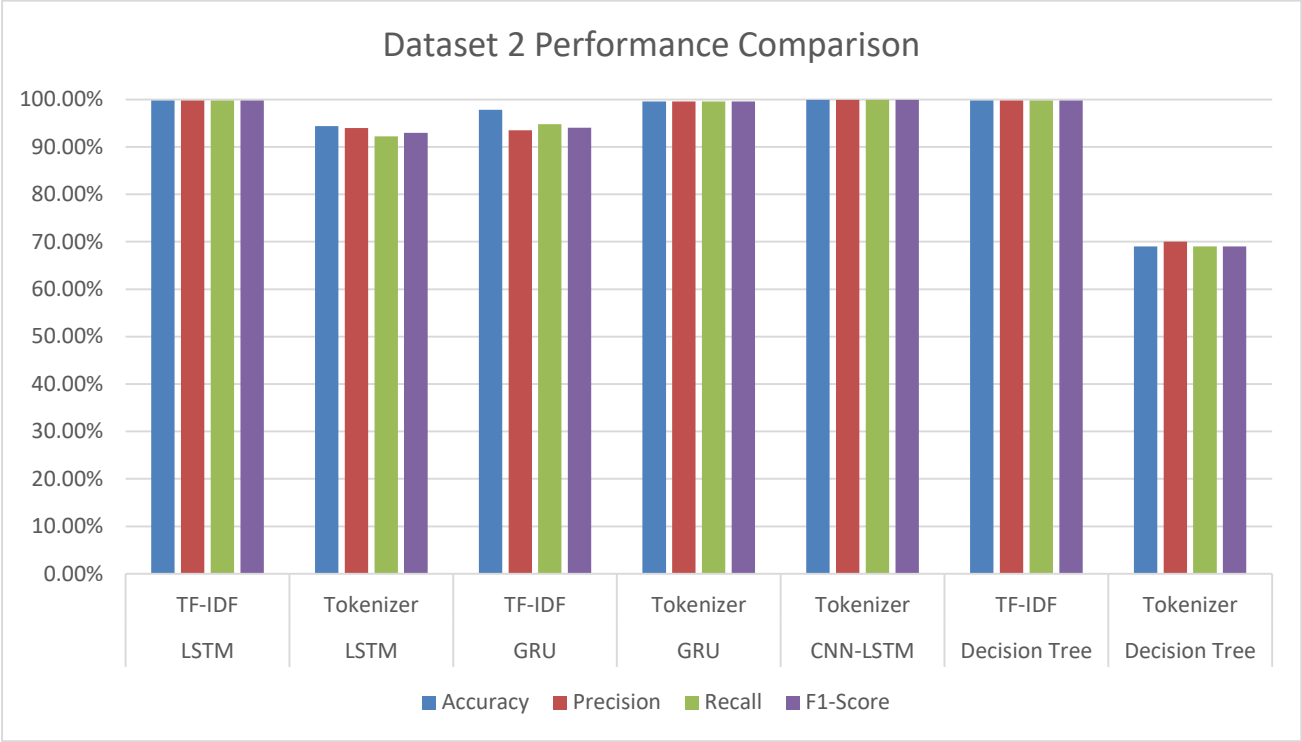

Fig. 7. Dataset 2 performance comparison.

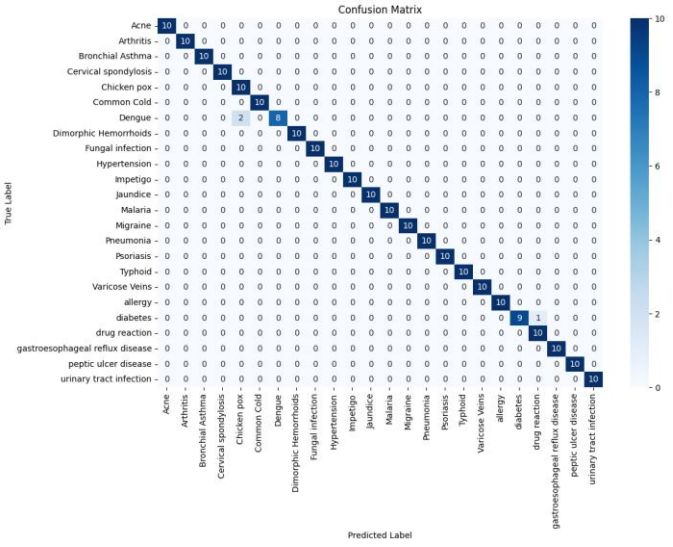

Fig. 8. Confusion matrix of Dataset 1 predicted by LSTM TF-IDF model.

Fig. 8 represents the classification results shown by the confusion matrices for the best model on each dataset. This shows that the LSTM TF-IDF model performed excellently on the first dataset, making only three classification mistakes.

In Dataset 2, there were more diseases than in Dataset 1, and the LSTM CNN model made only five classification errors in the second dataset as shown in the confusion matrix presented in Fig. 9.

E. Discussion

The experimental results demonstrate significant advancements in symptom-based disease classification when compared to prior research. For Dataset 1 (1,200 instances, 24 diseases), our CNN-LSTM with Tokenizer achieved 97 % accuracy and LSTM TF-IDF 98.8 %, comparable to the 99.58% accuracy of Hassan et al.'s MCN-BERT+AdamP model [9]. This near-parity is notable given our use of lighter architectures (LSTM TF-IDF vs. BERT-based models), which require substantially fewer computational resources. While their work focused on transformer-based language models, our results demonstrate that carefully optimized sequential models can achieve similar performance for symptom classification tasks without the complexity of large-scale pretraining.

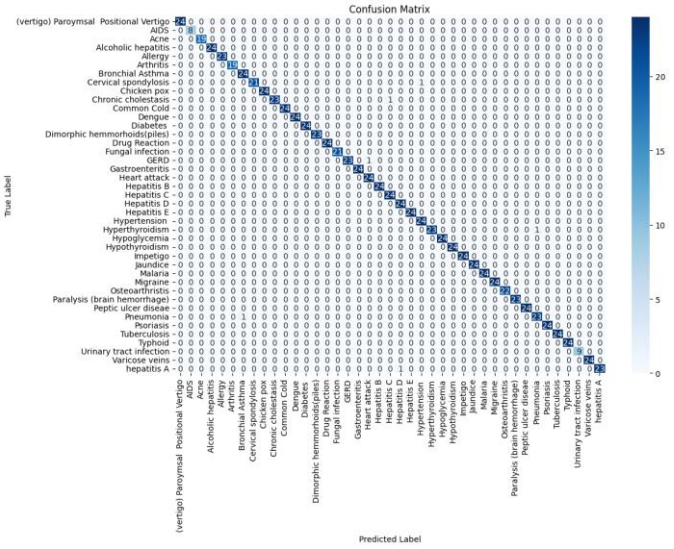

Fig. 9. Confusion matrix of Dataset 2 predicted by CNN-LSTM model.

For Dataset 2 (4,920 instances, 41 diseases), our CNN-LSTM achieved 99.90% accuracy, surpassing the 99.19% reported by Fuster-Palà et al. (2024) for their optimized KNN and FNN models [8]. Notably, while their study employed traditional machine learning methods (KNN, SVM, RF) with severity-encoded symptoms, our approach leveraged raw symptom text through advanced preprocessing (WordNet-based synonym replacement and row shuffling to address duplication artifacts), which may account for the marginal but consistent performance improvement across all metrics (F1-score: 99.80% vs. their 98%). Crucially, their work did not address dataset duplication—a limitation explicitly mitigated in our methodology, suggesting our results may better reflect real-world generalization.

To ensure a fair and transparent performance assessment, we selected baseline studies that used comparable encoding techniques (TF-IDF, tokenization) and classifiers. Our experiments directly compared the best-performing models from these prior works against ours, as shown in Table VIII and Fig. 10. Despite using simpler architectures, our models either matched or exceeded the State-of-the-art results. This demonstrates superior generalizability and suggests better real-world applicability—especially given our use of unstructured symptom descriptions rather than severity-encoded categorical inputs.

Table VIII represents a comparative analysis of disease classification performance across our proposed methods (TF-IDF and Tokenizer-based models) and prior works [8] [9]. Metrics include accuracy.

Fig. 10 shows accuracy comparison of disease classification models. Our CNN-LSTM (Tokenizer) achieves near-perfect performance (99.9%) on Dataset 2, outperforming prior FNN [8] and our LSTM TF-IDF near BERT-based approaches [9] with lower computational cost.

While many previous studies employed lightweight models such as Decision Trees or Naïve Bayes, our approach is specifically tailored to process unstructured symptom narratives, a format increasingly prevalent in modern healthcare settings. Although CNN-LSTM introduces some added complexity, it achieves 99.90% accuracy with manageable training time and memory usage, without the massive overhead of transformer-based models like BERT. To further justify its efficiency, future work will include benchmarking on throughput, memory footprint, and inference latency across hardware setups to quantify deployment feasibility.

TABLE VIII. COMPARATIVE ANALYSIS BETWEEN OUR RESEARCH AND OTHER STUDIES

Aspect	Our Research	**Fuster-Palà et al. (2024)** [8]	**Hassan et al. (2024)** [9]
Dataset	Dataset 1 Dataset 2	Dataset 2	Dataset 1
Algorithms	TF-IDF**:** DT, LSTM, GRU. Tokenizer**:** LSTM, GRU, CNN-LSTM.	KNN, SVM, RF, FNN.	MCN-BERT+AdamP (transformer-based).
Best Model	**Dataset 1**: LSTM (TF-IDF) – 98.8% accuracy. **Dataset 2**: CNN-LSTM (Tokenizer) – 99.9% accuracy.	FNN – 99.19% accuracy.	MCN-BERT – 99.58% accuracy.

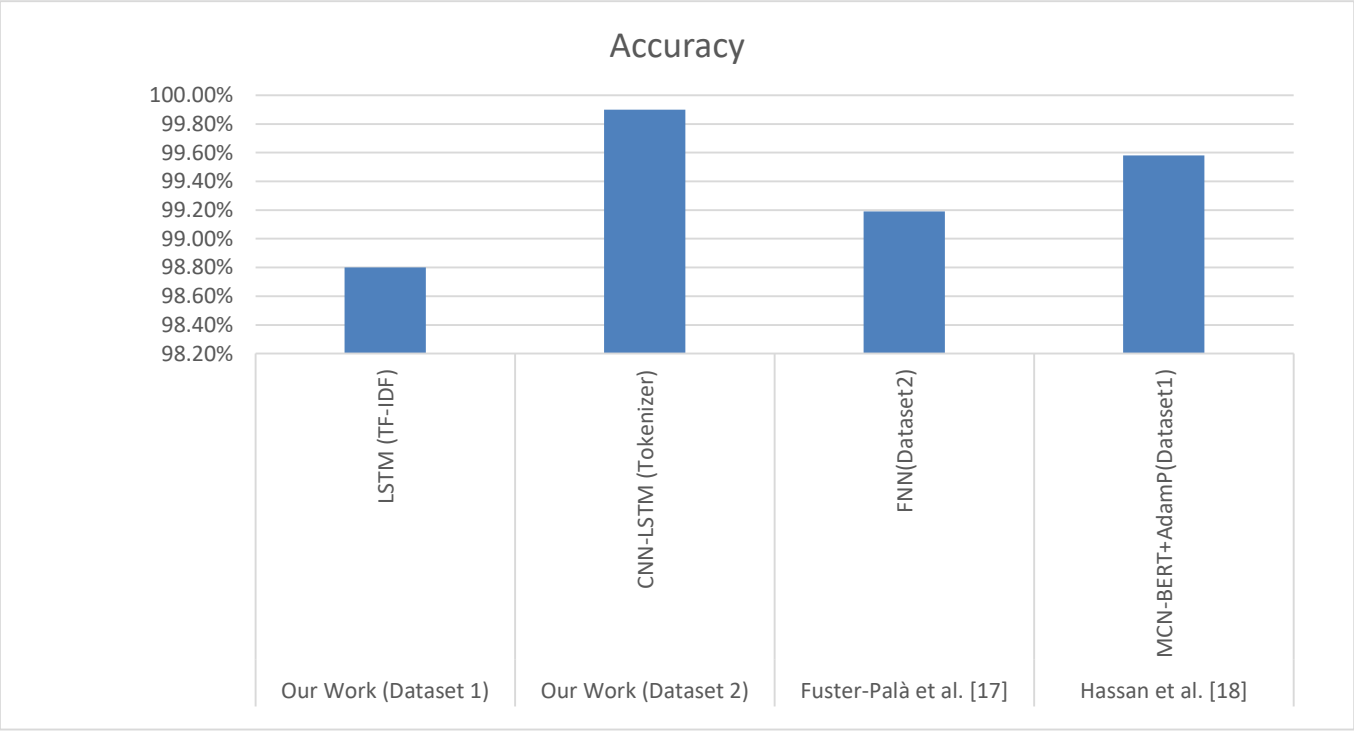

Fig. 10. Accuracy comparison between our research and other studies.

V. CONCLUSION

This project demonstrates the effectiveness of integrating Natural Language Processing (NLP) with deep learning models for disease prediction based on natural language symptom descriptions. By comparing TF-IDF and tokenization approaches across both traditional ML (Decision Tree) and DL models (LSTM, GRU, CNN-LSTM), the strengths of each technique depending on the model type are identified. Our results show that TF-IDF performs better with traditional classifiers, while tokenization combined with CNN-LSTM achieves the highest accuracy (99.90%) on a 41-disease dataset, highlighting the value of sequential pattern recognition. These findings underscore the potential of NLP-based models to improve early diagnosis and enable remote healthcare services, especially in settings lacking structured clinical data. Moreover, our models achieve competitive accuracy while remaining computationally efficient relative to transformer-based approaches, making them promising candidates for practical deployment. Future work will include benchmarking inference speed and resource usage to further validate deployment feasibility.

REFERENCES

[1] G. Almahadin, A. Lotfi, E. Zysk, F. Siena, M. Mc Carthy and P. Breedon, "Parkinson's disease: Current assessment methods and wearable devices for evaluation of movement disorder," *BMC Neurol*, vol. 20, no. 1, p. 1–13, 2020.

[2] Z. Youbin, T. Ning, O. Rawan, H. Zhipeng, D. Tuan, W. Jing, W. Weiwei and H. Hossam, "Smart Materials Enabled with Artificial Intelligence for Healthcare Wearables," *Advanced Functional Materials*, vol. 31, no. 51, 2021.

[3] K. A. Shastry and A. Shastry, "An integrated deep learning and natural language processing approach for continuous remote monitoring in digital health," *Decision Analytics Journal*, vol. 8, 2023.

[4] B. Pandey, D. K. Pandey, B. P. Mishra and W. Rhmann, "A comprehensive survey of deep learning in the field of medical imaging and medical natural language processing: Challenges and research directions," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5083-5099, 2022.

[5] J.-a. Sim, X. Huang, M. R. Horan, C. M. Stewart, L. L. Robison, M. M. Hudson, J. N. Baker and I.-C. Huang, "Natural language processing with machine learning methods to analyze unstructured patient-reported outcomes derived from electronic health records: A systematic review," *Artificial Intelligence in Medicine*, vol. 146, p. 102701, 2023.

[6] W. Rojas-Carabali, R. Agrawal, L. Gutierrez-Sinisterra, S. L. Baxter, C. Cifuentes-González, Y. C. Wei, J. Abisheganaden, P. Kannapiran, S. Wong, B. Lee, A. de-la-Torre and R. Agrawal, "Natural Language Processing in medicine and ophthalmology: A review for the 21st-century clinician," *Asia-Pacific Journal of Ophthalmology*, vol. 13, no. 4, p. 100084, 2024.

[7] M. R. Indupalli and P. G, "A Hybrid Blended Stacking Disease Prediction System Based on Symptoms," *preprint*, 2023.

[8] A. Fuster-Palà, F. Luna-Perejón, L. Miró-Amarante and M. Domínguez-Morales, "Optimized Machine Learning Classifiers for Symptom-Based Disease Screening," *Computers*, vol. 13, no. 9, p. 233, 2024.

[9] E. Hassan, T. Abd El-Hafeez and M. Y. Shams, "Optimizing classification of diseases through language model analysis of symptoms," *Scientific Reports*, vol. 14, no. 1507, 2024.

[10] A. Das, A. Sen and D. Choudhury, "A Collaborative Empirical Analysis on Machine Learning Based Disease Prediction in Health Care System," Techno India University, 2022.

[11] S. Abidin, P. R. Mutkule, P. Rajasekar, A. Kumar, D. Ghosal and M. Ishrat, "Identification of Disease based on Symptoms by Employing ML," in *2022 International Conference on Inventive Computation Technologies (ICICT)*, 2022.

[12] M. Das, K. Selvakumar and P. Alphonse, "A Comparative Study on TF-IDF feature Weighting Method and its Analysis using Unstructured Dataset," *arXiv*, 2023.

[13] C. W. Schmidt, V. Reddy, H. Zhang, A. Alameddine, O. Uzan, Y. Pinter and C. Tanner, "Tokenization Is More Than Compression," *arXiv*, 2024.

[14] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data* , vol. 8, no. 53 , 2021.

[15] I. D. Mienye, T. G. Swart and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, p. 517, 2024.

[16] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji and T. M. Brenza, "Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends," *Alexandria Engineering Journal*, vol. 61, no. 10, pp. 7585-7603, 2022.

[17] M. Waqas and U. W. Humphries, "A critical review of RNN and LSTM variants in hydrological time series predictions," *MethodsX*, vol. 13, 2024.

[18] I. D. Mienye and N. Jere, "A Survey of Decision Trees: Concepts, Algorithms, and Applications," *IEEE Access*, vol. 12, pp. 86716 - 86727, 2024.

[19] H. Blockeel, L. Devos, B. Frénay, G. Nanfack and S. Nijssen, "Decision trees: from efficient prediction to responsible AI," *Frontiers in artificial intelligence*, vol. 6, p. 1124553, 2023.

[20] O. Rainio, J. Teuho and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, 2024.