

Text Mining Techniques for Intelligent Grievances Handling System: WECARE Project Improvements in EgyptAir

Shahinaz M.Al-Tabbakh¹, Hanaa M.Mohammed², Hayam. El-zahed³

PhD in Computer Networks, Computer Sciences and Applications Group, Faculty of Women for Arts, sciences and Education, Ain Shames University, Cairo-Egypt¹

Internet Developer Dept. Manager of IT Sector-EGYPTAIR Holding Cooperation, Cairo- Egypt²

Prof of Solid-State Physics, Faculty of Women for Arts, Sciences and Education, Ain Shames University, Cairo-Egypt³

Abstract—The current work provides quick responding and minimize the required time of processing of the incoming grievances by using automated categorization that analyses the English text contents and predict the category. This work built a model by text mining and NLP processing to extract the useful information from customer grievances data to be used as a guideline to air transport industry. A customer grievances' system in EGYPTAIR called WECARE has had large feeds of data which can be collected in data sets through various channels such as e-mail, website or mobile Apps. Then the incoming data sets are analyzed and assessed by organization's staff then it is assigned to related department through manual classification. Finally, it provides proposed solution for the issue. Thence grievances categorization that handled manually is time consuming process. So, this work decided a model to improve WECARE system in Egypt Airlines. Classification based data mining Techniques are used to identify data into groups of categories across the variable touch points. The system has 166 categories of problems, but for experimental purposes we decided to study six categories only. We have applied four commonly used classifiers, namely, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naïve Bayesian and Decision Tree on our data set to classify the grievances data set then selecting the best of them to be the candidate grievances classifier in enhanced WECARE system. Among four classifiers applied on the dataset, KNN achieved the highest average accuracy (97.5%) with acceptable running time. Also, the work is extended to make hint to the system user, about how to solve this grievance issue based on previous issues saved in Knowledge Base (KB). Several experiments were conducted to test solution hint module by changing similarity score. The benefits of performing a thorough analysis of problems include better understanding of service performance.

Keywords—Knowledge base; grievances; NLP; SVM; KNN; Naïve Bayesian; decision tree

I. INTRODUCTION

The massive customer data in databases and World Wide Web is available in textual form so that manual analyses and deriving of useful information are not possible. Text mining is a computational automated technique used to find out considerable patterns of information from the unstructured texts [1]. This technique has created a strong industrial impact in decision making and non-trivial especially in companies

that works in airlines and communication industries [2]. Businesses use text mining applications to resolve customer demographics, to foretell future trends, to gain knowledge of contestants' developments and to make proactive and information-driven decisions [3].

A grievance handling system is a system that manages the process of how organizations handle, manage, respond and report to client's grievances. The manual categorization of the large number of grievances is extremely difficult, time consuming, expensive, is often not feasible and lead to unsatisfaction of the customer [4]. So to improve the quality of service the system need to minimize the processing time by replacing the manual categorization with automatic categorization, there must be an intelligent method to do so. Scaling passenger grievances in air industry requires in-depth Natural Language Processing (NLP) of the grievances. This problem is challenging due to two main reasons [5]:

- The data come from various persons from different affiliated organizations. The authors of the grievance have different writing styles and input formats for recording the grievances as well as the actions taken (if any) in response to grievance.
- Huge size of the data. A typical airline manages thousands of passengers; each of them can potentially contribute unsolicited feedback. As opposed to survey studies where the airlines would ask the information and control the format, Undesirable feedback is initiated by the passenger, the passenger's family, and in some cases, by the care provider. The benefits of performing a thorough analysis of problems include better realizing of service performance, better understanding of how to focus efforts to reduce troubles, and a better understanding of how people are affected by these Problems.

We organize the rest of this paper as follows. Section 2 presents the related studies. Section 3 presents a survey of some theoretical Aspects. Section 4 describes the proposed methodology and system improvements and experiments that have been conducted to assess the proposed categorization approaches. Section 5 presents the experimental results and discussion. Section 6 describes the application of the proposed

classification approach and the improvement of WECARE system for Egypt airlines to produce hint to the system user, about how to solve this grievance issue based on previous issues saved in Knowledge base (KB). Lastly, Section 7 provides conclusions and recommendations for future work.

II. RELATED WORK

Customer satisfaction is noticed as one of the most important key performance pointers of success of any agency. There are few studies performed in airlines industry for grievances handling and service recovery based on data mining technique and natural language processing. But there are a considerable number of studies made in other applications especially in healthcare systems and quality management. The current section presents a review of the relevant literatures. Maia et al. apply the text mining methods for classification of documents for automation of grievances screening in a Brazilian Federal Agency. This work applied four machine learning algorithms: SVM, Naïve Bayesian, Random forest, and Decision Trees. They were estimated with the following measures: kappa, specificity, F measure and sensitivity for each algorithm. The best of them was random forest with 0.84 F measure and 0.77 Kappa. Also this work limited the scope of the work to just 4 units out of 82, the results obtained show that it is possible to implement an automatic classifier using text mining for grievances screening [6]. Sheheta and Karray [7] proposed a new concept based model to improve the text categorization quality by employing the semantic structure of the sentences in documents. The introduced model involves three levels of connotation-based analyses. Firstly, the sentence-based connotation analyses which analyze the semantic structure of each sentence to engage the sentence connotations using the proposed Conceptual Term Frequency (CTF) measure. Secondly, the document-based connotation analyses which analyze every connotation at the document level using the concept-based Term Frequency TF. Last, the corpus-based concept analysis that analyses concepts on the corpus level using the document frequency DF as a global measure. The connotation-based analyses assigns weight to each connotation in a document. The top connotations that have maximum weights are used to build standard normalized feature vectors using the standard VSM for the purpose of text categorization. Al-Nagar [8] developed a three phases automatic complaint system for UNRWAA organization. First phase analyze the complaint message contents, categorize it by using text categorization algorithms and try to decide where to direct the question request automatically to the right person in order to get it answered. Second Phase system, used text similarity methods to suggest the answers. The third phase system applied the summarization technique to update the FAQ library with the most asked questions. The analysis approved that SVM classifier achieved the highest average accuracy with 75%. Also, for suggestion part, the best F-Measure resulted 73% at similarity score 0.5. Al-messieri et al. [5] proposed a new tactic of mapping complaints into sentiment vectors utilizing domain specific developed linguistic Inquiry and Word Count (LIWC) dimensions. He demonstrated and implemented a machine learning model for patient grievances classification based on the proposed method. He accommodated the

disparity in the used language and style and explored using domain specific grammatical dependency for feature extraction. He designed a method to extract domain-specific terms which used to construct a set of grammatical dependencies. He applied eight machine learning models for patient complaint classification using the explored rules to achieve significantly higher results as compared with basic unigram features using the same models. Yakut et al. [9] explored customer review data for in-flight services of airline companies and draw customer models with respect to such data. He applied two modelling techniques as feature-based modelling and clustering-based modelling. In feature-based modelling, customers are grouped into categories based on features such as cabin flown types, experienced airline companies. In clustering-based modelling, customers are first clustered by means of k-means clustering and then modeled. Then the multivariate regression analysis was used to model customer classes in both cases. Tang H., et al. [10] discussed some tasks used to do an automatic assign to one document as positive or negative such as similarity approach, where IR method is used to get the documents that are relevant to the sentence in query. Then, calculate its scores of similarities with each sentence in others documents and calculate an average value. If that average value of opinionated documents is greater than that of initial document, then the sentence is classified as a positive sentence else it is negative.

III. TEXT MINING TECHNIQUES

Text mining is like data mining, but it is an extended version of data mining. It leads to discovery of new knowledge from large volume of the existing unstructured data [11]. It is also called, as text data mining and information discovery from word-based databases. Generally, text mining processes has text categorization or classification, document summarization, entity extraction, topic tracking, text clustering, information visualization, question answering, etc. [12]. Text Mining tries to extract fruitful information from multiple data sources. One difference with numeric analysis of data is that the documents always are unstructured. That is why in mining the text the pre-processing tasks are important. These operations are responsible for transforming data from unstructured to structured format for better document manipulation. Text mining is commonly used for: Classify documents according on their content, organize thesis's contents for search and retrieval, automated comparison of information in different industry and extract specific information from any document [13].

A. Elements of Text Mining

Text Mining is characterized by some common elements as:

- **Corpus** regarded as a combination of many documents
- **Document** regarded as a combination of many terms
- **Term** is a word of any human language

To put text mining on job, we need to prepare our data to the mining methods. As shown in Table 1, pre-processing includes the following steps:

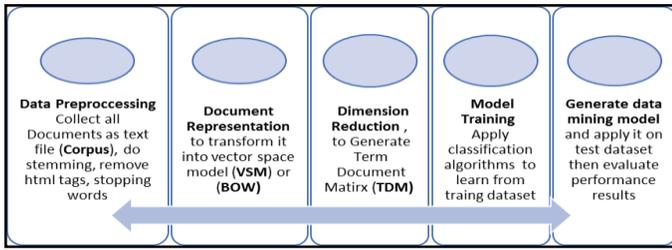


Fig. 1. Text Categorization Steps.

C. Text Categorization

Text categorization (TC), also known as text classification, search in classifying documents for pre-defined query based on their contents. It can be many categories, the definitions is user- dependent for a given task, we might be dealing with as few as two classes (binary classification) or as many as thousands of classes. [16]

In text categorization a method assorts content of documents according to predefined class. Applications of TC include text filtering and ranking of Web pages, as illustrated in Fig. 1.

D. Text Similarity Measurement

In this measurement, the features or tokens of documents are represented as vectors in the space. Typically, the angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle computes the degree of similarity between these two vectors. Similarity, since cosine has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors [17].

Calculate a distance between entries one by one in a list. Assuming X and Y are different entries, using the above formula, the weight of X and Y can be calculated, and is represented in the form of X and Y vectors; xi is a weight of any word in the vector space. X and Y are expressed as:

$$X = \{X_1, X_2, \dots, X_k\} \quad Y = \{Y_1, Y_2, \dots, Y_k\}$$

Cosine relevancy between the two vectors, using the formula of cosine similarity is calculated as follows [17].

$$\cos X, Y = \frac{X \cdot Y}{|X| \cdot |Y|} \quad (3)$$

Cosine similarity formula is a mathematical method to show the relevancy between the different entries. When the value is close to 1, the two entries have greater connection.

As a good summary of previous stages, we calculate terms weighting, which are sorted by size, and it is done as follows:

- 1) Each document is modelled as a bag of words (BOW)
- 2) BOW is a list of terms and count of each term (word).
- 3) The whole collection could be modelled as a "list of Bag of Words"
- 4) Calculate frequency for each word, Applying TFIDF weighting scheme on data texts.
- 5) Get TDM. In TDM, rows resample after documents, columns resample after terms.

- 6) Each table value is count of term frequency.
- 7) To calculate similarity value as cosine of angle lies between documents Vectors
- 8) Apply dot product on vectors of unit-length
- 9) Hande the search query as a document
- 10) Calculate (VSM cosine) similarity between query document and each document in collection.

IV. EXPERIMENTAL FRAME WORK AND RESULTS USING WEKA

For research purposes we chose to use WEKA classifiers in our experiment. We have applied SVM, KNN, Naive Bayes and Decision Tree methods on our data to classify the grievances data set, and then selecting the best of them to be the proposed *grievances Classifier* in our system.

A. Data Preparation for Machine Learning

The basic concept of data classification is to determine the type of class to which a data point belongs based on the features that this point owns. This can be compared to the known features for each of the potential categories, and the data is then categorized as the category with the most characteristics. It's required that information about different classes is collected in advance. It is done by the learning or training a list by using a dataset where data points are previously categorized into many categories.

B. Training the Classifier

To train the classifier, each of the predefined data points is first run from the input data set by a specific method that analyses the data and stores several attributes that can identify that data point. The resultant group is then inserted into an automated learning algorithm that attempts to deduce conclusions based on all the classified features that are collected and constructs a model based on those that can be used to classify the unmarked data [18].

C. Testing Data with the Classifier

The sample classifier created using predefined data can then be used to classify the unrecognized entry on the same rule. Each data point in the data set was run through a feature extractor that was then sent to the classifier form. In most of cases, the document collections are split into two sets: (Training and Testing set). The training set is used to build a classifier. The testing set is implemented to evaluate the classifier. This is illustrated in Fig. 2.

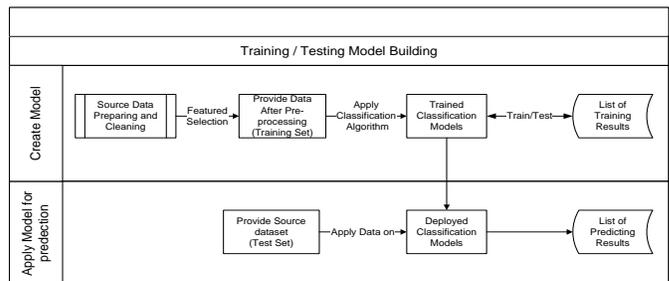


Fig. 2. Training and Testing Data with the Classifier.

Moreover, what is worth to mention, using a large dataset to build the classifier model will improve the performance when classifying new dataset. But this is true up to a specific limit. Using large data can affect the classifier to become slower, since there are too many rules to compare data against. Scale of the datasets is an important factor that related to the task preparation, data quality and selected algorithm. The current experiments are about classifying data set of incoming emails as unstructured data. The reason why we decided to use this dataset was that it is a data set available in a wide range of daily life through the IT unit on Egypt Air lines. The goal was to monitor efficiency of different classification algorithms performed on them, not only by comparing the resulting confusion matrix, but also by comparing running time required to build the model depending on the size of the input data and the number of used attributes. Posteriorly, the best classifier will be implemented in C# code.

D. Waikato Environment for Text Analysis

We used WEKA software that provides all the steps of the text mining process such as pre-processing, vector generation, classification and visualization of the results. The text mining pre-processing steps for pre-processing is shown in Fig. 3. The environment also includes several machine learning algorithms. For task of text categorization, the machine learning algorithms namely, SVM, KNN and NB and J48 evaluated with WEKA platform [19]. Text mining processing and classification using WEKA is shown in Fig. 3.

E. Grievances Classification

We carried four types of classifiers for classifying the new grievances and compared them to select the best classifier in the system. The reason why we choose to use the following data set was that they are based on actual real-life data and that they are both relatively complex. The goal was to see how well the different algorithms performed, by looking into the time required to construct the classification model depending on the size of the input data and number features used of as well as the time required to classify a dataset using the generated classification model [20]. EGYPTAIR dataset are chosen for its grievances system that contains thousands of text messages of different lengths that belong to about 166 different categories. The data collected from April 2017 to March 2018. A dataset of total 5600 grievances were available in the current system. The data set contains 166 classes that describe the groups of them as *Flight Delay category* for grievances of Flight Delay problems, *Baggage category* for lost baggage etc. We decided to classify the grievances of 6 out of the available 166 for study purpose. It reduced the instances to 1004 out of 5600 to train and test our system. The chosen classes are listed in Table 3. The data used in this work were elicited from the SQL Server where the grievances are stored and then applied into WEKA, the tool that used for text mining in this work and machine learning. The only data used for classification was the text describing the grievance that elicited from Email or social media.

F. Steps of Text Pre-Processing using Weka

We depended on WEKA’s most common method to pre-process data (StringToWordsVector), as show in Fig. 4. The StringToWordVector-filter had 16 different settings that you

could adjust to work with classification. The first step is preparing the data to be ready for applying text mining methods, to transform the text messages to a form that is suitable for used algorithms. In current experiments, we used `stringtoWordVector` feature to prepare data as described previously in literature. The result was converting grievances text messages to Word List that contains the occurrence of each word in the category as shown in Table 4. After applying the text mining framework provided by the WEKA, `stringtoWordVector` feature, it executes the pre-processing step; usual techniques of stemming text, removing stop-words, removing less significant words, changing all text to lower case letters, and erasing punctuation and numeric characters. This will produce a list count of each word in dataset, as we mentioned before a term of document matrix TDM is ready to give each word or term its weight in the whole list [21].

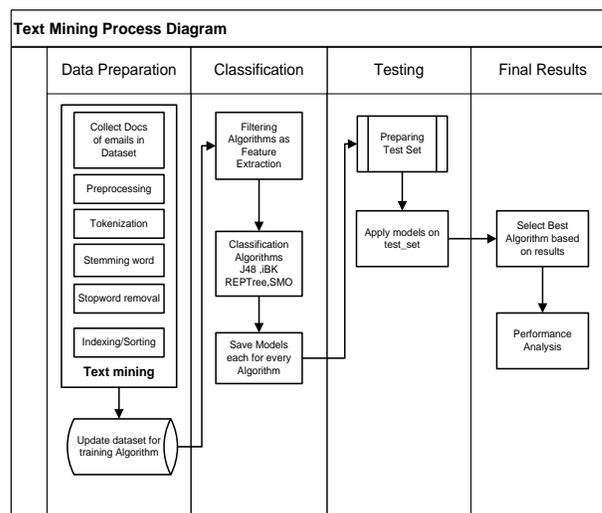


Fig. 3. Text Mining Process of New Model.

TABLE III. SELECTED SIX CATEGORIES COUNTS AND DESCRIPTION

Class	count	Class name	description
A3	207	Sales office staff	when customer has a problem in any sales office
B3	165	Reservation Website	when customer has a problem during reservation through website
C4	104	Frequent Flyer Missing miles:	when frequent flyer customer travels from city A to City B , the miles between 2 cities is measured in miles then their distance is added to his card in units of miles , the problem arise when he cannot find these miles after arrival.
D1	210	Flight delay	when customer has a problem due to flight delay
D10	208	Missing Baggage	when customer has a problem due to missing his baggage
I2	112	Lost items	when customer has a problem due to lost items

TABLE IV. WORD LIST AND COUNTS

Word	Occurrence	C4	A3	B3	D1	I2	D10
able	4	1	1	0	0	1	1
access	18	5	4	0	5	3	1
airport	19	3	8	3	2	2	1
allowing	23	3	5	4	5	6	0
arrive	21	5	4	5	4	3	0
attached	23	5	3	4	7	4	0
boarding	14	1	3	2	5	2	1
booking	24	1	3	5	9	6	1
calling	22	3	4	5	4	5	1
carrier	25	2	5	3	5	9	1
check	10	1	0	1	5	1	2
choose	10	2	4	0	3	0	1
confusing	26	4	5	7	5	3	2
counter	30	5	7	3	7	7	1

The resulting feature set of classes are six classes; as shown in Table 4 contains the resulted counts of vectors for each one.

Term-document matrix (TDM): The current data were divided in two parts: training, and test. The first received 66% of the data and the others 33% each. The training data is used for learning the classification models. Finally, the test data is used to evaluate the selected classification model performance, to verify if it can be generalized to unseen data. To ensure the reliability of the results, 5-folds cross validation test was followed [22]. The data set is divided into five equal subsets. Each of them is used once as testing data where the other four subsets are the training data. So we have applied SVM, KNN, Naive Bayes, and Decision Tree methods on our data set and compared them to select the method that achieved the *highest accuracy and suitable running time* to construct the classifier. We used WEKA to establish the selected classification methods to choose suitable classifier in our system, see Fig. 4 to set the classification process in WEKA.

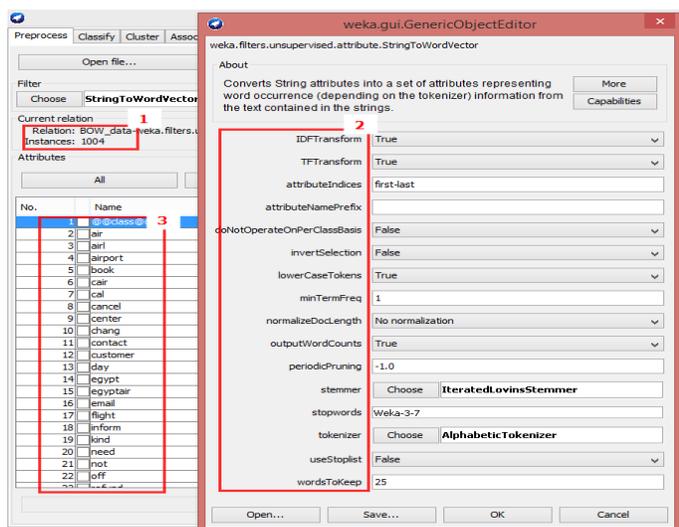


Fig. 4. Command Panel.

Here the panel of setting this parameter:

- Part 1 shows number of instances
- Part 2 shows the parameters setting values,
- Part 3 the resulting list of words from the input file

The full list of parameters and each description is shown in Fig. 5.

G. WEKA File Format

The main file format used in WEKA is their own called (ARFF) Attribute Relationship of File Format, as short notation [23]. It is a normal text file with the structure as shown in Fig. 6. For each email contents, the data was subjected to text mining process (data cleaning, stemming, remove stop words and indexing) the ARFF file to train model has two attributes (Desc, class). Desc is the description of email contents. Class is the one of six chosen classes. The instances of data are separated by comma. On the test set as rest of the file you will see a question mark? Instead of class, here to tell WEKA to deduce the missing class with numeric prediction accuracy Percentage.

parameter	Command description
-C	Output word counts rather than Boolean word presence.
-D	delimiter_characters Specify set of delimiter characters (default: " \t...^WO? !@")
-R first-last	... Specify list of string attributes to convert to words. (default: all string attributes)
-P	attribute_name_prefix Specify a prefix for the created attribute names. (default: "")
-W	number_of_words_to_keep Specify number of word fields to create. Other, less useful words will be discarded. (default: 1000)
-A	Only tokenize contiguous alphabetic sequences.
-L	Convert all tokens to lower grievance before adding to the dictionary.
-S	Do not add words to the dictionary which are on the stop list.
-T	Transform word frequencies to log(1+Fij) where Fij is frequency of word i in document j.
-I	Transform word frequencies to Fij*log(numOfDocs/numOfDocsWithWord) where Fij is frequency of word i in document j.
-N	Normalize word frequencies for each document(instance). The frequencies are normalized to average length of the documents specified in input format.

Fig. 5. The Full List of Parameters and Description.

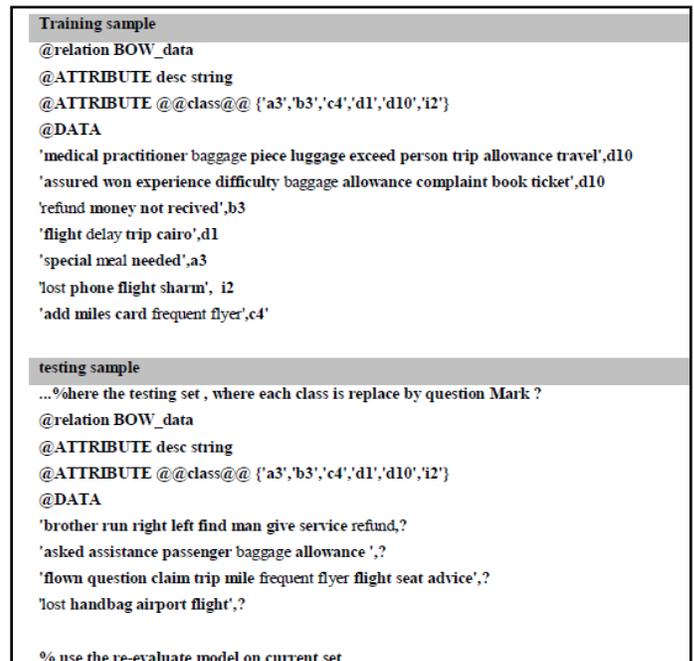


Fig. 6. Weka File Format.

H. Applied Weka Classifiers

The experiments done with four algorithms synchronized together and results will be discussed in full, in the next subsections.

1) *Decision tree (J48) algorithm:* Decision Tree is an algorithm used for classification, it generates a tree with each branch of it represents a decision. By using set samples in training data, it builds the tree. At every node of tree, it selects one attribute of the data that divides set of samples into two subsets located in one class or in the other. Its 6 categories are normalized gain of information that results from choosing any attribute for good splitting the data. The attribute with the highest normalized acquisition of information is chosen to make the decision. Algorithm for decision tree use divide-and-conquer to constructs the tree in a top-down recursive. Hereunder, the brief of the algorithm steps [24]:

- a) Initially, all the samples are at the root level
- b) Samples are separated recursively based on chosen attributes
- c) Test attributes are elected based on a heuristic or statistical measure
- d) The algorithms stop separation in one of the following conditions:
 - IF all the samples belong to same class.
 - IF there were no attributes remained for next separating.
 - IF there were no samples left.

The results of applying decision tree J48 algorithm on our dataset as shown in Fig. 7.

2) *K-Nearest neighbors (KNN) algorithm:* It is a classification method used for classifying objects according to nearest training samples in the set of feature space. *KNN* is a type of lazy learning where the function is only locally, and all computation is deferred until classification. *KNN* is one of the simplest algorithms: when an object is classified by a total vote of its neighbors, (consider *k* is a positive small integer) with the object being selected to the class among its *k*-nearest neighbors. If *k* = 1, then the object is simply assigned to the class of its nearest neighbor [25]. We used WEKA to apply (IKB lazy) algorithm on our dataset as seen in Fig. 8.

```
==== Evaluation result ====
Scheme: iBK_InputMappedClassifier : InputMappedClassifier
Options: -I -trim -L D:\GA-weka\ibk_model.model -W weka.classifiers.lazy.IBk --
Relation: BOW_data-weka.filters.unsupervised.attribute.StringToWordVector-R1-W100-stemmers.NullStemmer-M1-O-tokenizerweka.core.tokenizers.AlphabeticTokenizer-weka.f

Correctly Classified Instances      974          97.012 %
Incorrectly Classified Instances    30           2.988 %
Kappa statistic                    0.9636
Mean absolute error                0.2232
Root mean squared error            0.311
Relative absolute error             81.5386 %
Root relative squared error        84.0654 %
Coverage of cases (0.95 level)    100 %
Mean rel. region size (0.95 level) 84.8938 %
Total Number of Instances         1004
```

Fig. 8. Applying IKB Method using WEKA.

3) *Naïve Bayesian algorithm:* While Bayes theorem calculates the probability of one event occurring given that another event has already occurred, *Naïve Bayesian* modifies the method and naively assumes that each event is conditionally independent of each other.

Naïve Bayesian makes it a fast and scalable algorithm that performs surprisingly well compared more complex models if your data set doesn't grow too much. The results of applying WEKA on *Naïve Bayesian* algorithm on our dataset is seen in Fig. 9.

Naïve Bayesian will run into a problem if you encounter data with a variable having zero probability since it will ruin your equation when multiplied with the other variables. However, this can be fixed if you smooth the data beforehand where zero probabilities are removed [26].

4) *Support vector machines (SMO) algorithm:* It is a set of related methods of supervised learning that analyze incoming data and recognize outcome patterns used for classification and regression. It's namely (SMO) in WEKA. If we have a set of training items, each one has a previous category, SVMs training algorithm create a model that predicts either a new object lay into one category or the other [27]. Initially, SVM model resample objects as points in vector space, with big gap between the objects of the separate categories they are divided that is as big as possible. A SVM put one or set of hyper-planes in the m-dimensional space. So, a fair separation is gained by the hyper-plane which has the largest space to the nearest training data points of any class. In general, the bigger the margin the smaller is the generation error of the classifier. We used WEKA to apply SVM algorithm, named as SMO, on our dataset as shown in Fig. 10.

```
==== Evaluation result ====
Scheme: J48_InputMappedClassifier : InputMappedClassifier
Options: -I -trim -L D:\GA-weka\j48_model -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation: BOW_data-weka.filters.unsupervised.attribute.StringToWordVector-R1-W100-prune-rate-1.0-C-T-I-W0-L-S-stemmerweka.stemmers.NullStemmer-M1-O-tokenizerweka.core.tokenizers.AlphabeticTokenizer-weka.filters.unsupervised.attribute.ClassAssigner-D-C

Correctly Classified Instances      802          79.8805 %
Incorrectly Classified Instances    202         20.1195 %
Kappa statistic                    0.7535
Mean absolute error                0.1022
Root mean squared error            0.2254
Relative absolute error             37.3553 %
Root relative squared error        60.9376 %
Coverage of cases (0.95 level)    98.008 %
Mean rel. region size (0.95 level) 45.3353 %
Total Number of Instances         1004
```

Fig. 7. Applying Decision Tree Method using WEKA.

```
==== Evaluation result ====
Scheme: NB_InputMappedClassifier : InputMappedClassifier
Options: -I -trim -L D:\GA-weka\nb_model -W weka.classifiers.bayes.NaiveBayes --
Relation: BOW_data-weka.filters.unsupervised.attribute.StringToWordVector-R1-W100-prune-rate-1.0-C-T-I-W0-L-S-stemmerweka.core.tokenizers.NullStemmer-M1-O-tokenizerweka.core.tokenizers.AlphabeticTokenizer-weka.filters.unsupervised.attribute.ClassAssigner-D-C

Correctly Classified Instances      589          58.6653 %
Incorrectly Classified Instances    415         41.3347 %
Kappa statistic                    0.4938
Mean absolute error                0.1381
Root mean squared error            0.3678
Relative absolute error             50.4551 %
Root relative squared error        99.4211 %
Coverage of cases (0.95 level)    60.259 %
Mean rel. region size (0.95 level) 17.7623 %
Total Number of Instances         1004
```

Fig. 9. Naïve Bayesian Classifier using WEKA.

TABLE VI. SYSTEM RULES AND RESPONSIBILITY IN THE CURRENT SYSTEM IN EGYPT AIRLINES

Roles	Responsibilities
Admin (A-M)	Customer Services team Admin is responsible to download grievances from mailbox, distribute them among team and review team performance. Also can follow up grievances, manage users, dashboard, security.
Member (CS-M)	Customer Services member is responsible to review and categorize grievance sent by admin, divert the grievance requests coming from the different channels to the responsible department (handlers).
grievance handler member (H-M)	grievance handler is normally one or more person at each department and is responsible to handle and resolve the grievance assigned to him by the team

TABLE VII. SYSTEM WORKFLOW STEPS

ser	Step	Description
start	Download email inbox	This step is where mail box of WECARE account is downloaded and accumulate messages to be saved in mails Data base with tagged NEW
1	Review , distribute grievance to CS member	The CS team Admin should review the new mail or form then sets its criticality level. Distribute it to one of CS member s with extra comment.
2	start grievance case from mail	Check if the mail is a new grievance then, open as new case. Or related mail of old case, new cases are given a new unique reference number and sends a confirmation email to the customer.
3	Contact Customer	The CS member contact customer for more information concerning his case., if needed
	Receive response from customer	The CS member receives response from customer (by mail) with the extra information about his issue.
4	Review then assign to Handler(s)	The CS member should review the new case details and review CS Admin 's comment and assign it to one of Handler(s) And checks if this grievance is a single grievance or multiple
5	Receive by Handler(s)	The Handler(s) checks the grievance content and try to solve it with his department. The Handler(s) receives response from customer , colleague or CS team with the extra information about case , their responds are added to flow of case
6	If response can Resolve	Handler(s) finish working and solves the case
	not Resolved	Handler(s) keep working in step 6 until he solves the case
7	If action(s) Approved	The CS member approves the Handler(s) decision , case is closed
	If not Approved	The CS member reassign it to another Handler(s) (back to step 4)
	Review case resolution(s)	The CS member -contact customer with the grievance resolution
8	Case closed	The CS member -receives feedback from customer
		if positive response is obtained , system close the case if negative response is obtained it is routed to the CS Team Admin to escalate it
9	case is archived	If case closed, the data is accumulated into data list to be analyzed and get statistical charts from it

VI. METHODOLOGY

A. Suggested Model for Improving WECARE System

Customer grievances handling system in WECARE will be modified by adding more machine intelligence to it by building a new model of text mining to collect all previous grievances data as shown in Fig. 13 and check the new grievances against it, to achieve automatic grievance categorization, and automated solution hint. The proposed part of the system is to include set of text mining techniques written in c# with MS-SQL 2012 tool, to analyze and classify incoming email automatically based on the previous grievances learning datasets update the KB library.

A new module in C# have been applied to extract important terms out of new incoming email text data , this method is applied on EGYPTAIR data set to classify the new grievances. SQL statements were used to search our dataset and apply text pre-processing to make the text documents suitable to search it, it includes **tokenization** to convert input text to list of tokens, **stop word removal** to remove unnecessary words, **stemming** to remove suffixes of the resulted features and **weight-evaluation** to select the important terms, based on their TF-IDF weight in each category. The pre-processing steps are shown in Fig. 4 and the C# code is shown in the following section.

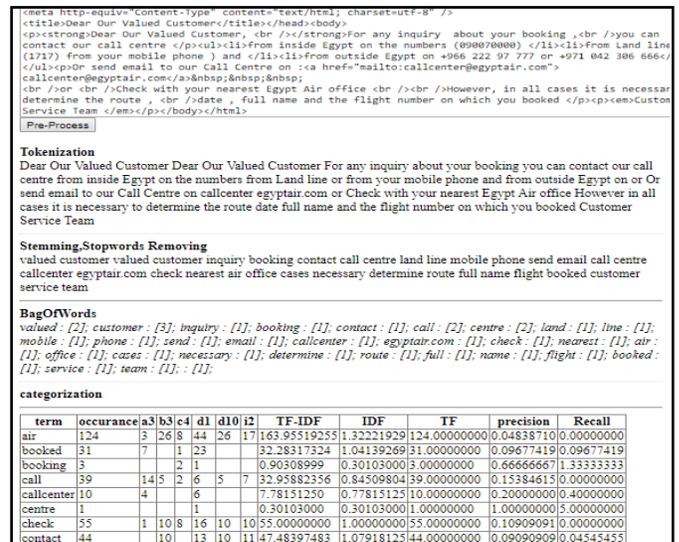


Fig. 13. Text Mining Running Process in Improved WECARE.

C# Code used in text mining improvement in WECARE system

```
<html dir="ltr" xmlns="http://www.w3.org/1999/xhtml"><head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script runat="server" language="c#">
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class CaseManager : System.Web.UI.Page
```

```

string category;
string html = "";
string _mailBody;
string _taskId;
string summery="";
protected void Page_Load(object sender, EventArgs e)
{
    List<string> doc1 = new List<string>();
}

private string Tokenizing( string src )
{
    //replace html tags trminaror directive &gt; into <
    src = src.Replace("&gt;",">");
    src = src.Replace("&lt;","<");
    src = src.Replace("&nbsp;"," ");
    src = src.Replace("_","");

    //remove html tags as <b>,<b> etc
    Regex regEx = new Regex("<[^>]*>","RegexOptions.IgnoreCase |
    RegexOptions.Multiline);
    string afterTags=regEx.Replace(src, " ");

    //Change your regex to ^[a-zA-Z0-9_@.-]+$. Here ^ denotes the beginning of a string,
    $ is the end of the string.
    string remove = Regex.Replace(afterTags, @"^[a-zA-Z0-9_@.-]+$", "");
    string result = Regex.Replace(remove, @"^[a-zA-Z0-9_@.-]+", " ");
    return result ;
}

private string removeStopWords(string src)
{
    string[] strSrc = src.ToLower().Split(' ');
    string[] stopwrds={
        "about", "after", "all", "also", "yes", "and", "another", "any", "are", "she", "why", "be",
        "because", "been", "before", "being", "between", "already", "they", "both", "but", "dear",
        "came", "can", "come", "could", "did", "sir", "does", "each", "else", "for", "from", "get",
        "got", "had", "has", "however", "ever", "have", "numbers", "her", "hello", "here", "him",
        "himself", "his", "how", "thank", "please", "into", "you", "number", "its", "just", "like",
        "make", "many", "this", "might", "more", "most", "much", "must", "my", "never", "now",
        "new", "anyway", "only", "madam", "other", "our", "out", "over", "greeting", "said",
        "same", "see", "should", "since", "regards", "some", "someone", "something", "still",
        "such", "take", "than", "that", "the", "their", "them", "then", "there", "these",
        "through", "whom", "too", "under", "yours", "use", "very", "want", "was", "way",
        "thanks", "well", "were", "what", "when", "where", "which", "while",
        "who", "those", "will", "with", "would", "you", "your", "between", "centimeter", "info",
        "com", "regarding", "one", "two", "three", "four", "five", "may", "best", "description"
    };
    string result="";
    bool wrdfnd=false;

    for (int i = 0; i <= strSrc.Length-1; i++)
    {
        //omit string length <=2
        if (strSrc[i].Length > 2)
        {
            for (int j = 0; j <= stopwrds.Length-1; j++)
            {
                //compare text word with All StopWords array items ,
                //if found break loop
                if(String.Compare( strSrc[i],stopwrds[j],true ) == 0)
                {
                    wrdfnd=true;
                    break;
                }
                else
                {
                    wrdfnd=false;
                }
            }
            if (!wrdfnd)
            {
                result+= strSrc[i]+" ";
            }
        }
    }
    return result;
}

private string BagOfWords(string src)

```

```

string result="";
string[] source = src.ToLower().Split(' ');
var frequencies = new Dictionary<string, int>();
string highestWord = null;
int highestFreq = 0;

foreach (string word in source)
{
    int freq;
    frequencies.TryGetValue(word, out freq);
    freq += 1;
    // frequencies.Add(freq, word);
    if (freq > highestFreq)
    {
        highestFreq = freq;
        highestWord = word;
    }
    frequencies[word] = freq;
}

int max = frequencies.Values.Max();
foreach (KeyValuePair<string, int> kvp in frequencies)
{
    if (kvp.Value <= max)
    {
        result+=kvp.Key + " : " + kvp.Value.ToString()+"<br>";
    }
    else break;
}

return result;
}

protected void btn_click(object sender, EventArgs e)
{
    string html = Tokenizing(TextBox1.Text);
    Label1.Text =afterTokenizing;
    summery =removeStopWords(afterTokenizing);
    Label2.Text =summery;
    category =BagOfWords(summery);
    Label3.Text =category;
}

</script>
</head>
<body>
<form id="form1" runat="server">
<asp:TextBox runat="server" id="TextBox1" TextMode="MultiLine" Width="494px"
Height="225px"></asp:TextBox>
<br />
<asp:Button runat="server" Text="Pre-Process" id="Button1"
OnClick="btn_click"></asp:Button>
<br /><br><strong>Tokenization</strong><br>
<asp:Label runat="server" ID="Label1" Text="-"/>
<br /><hr><strong>Summerization</strong><br>
<asp:Label runat="server" Font-Bold="true" ID="Label2" Text="-"/>
<br /><hr><strong>BagOfWords</strong><br>
<asp:Label runat="server" Font-Italic="true" ID="Label3" Text="-"/>
</form>
</body>
</html>

```

VII. SOLUTION HINT MODULE FOR SUGGESTION OF ANSWER

A. Getting Similarity Issues by LEVENSHTAIN MODULE in C#.Net Package

This module provides all of the functionality for coding lines of text, making comparisons and calculating the edit distance and similarity score. The actual Levenshtein functionality is called from C#.net package for text mining and the Natural language processing .In LEVENSTEIN module two-dimensional arrays to store the distances of prefixes of the words compared, and return the amount of difference between the two strings based on the minimum number of operations needed to transform one string into the

other, where an operation is an insertion, deletion, or substitution of a single character. The program starts by displaying received grievance documents [27].

1) The main steps of answers suggestion part:

a) First select a grievance.

b) Then compare it with the stored in the data grievances base and return the similarity score

c) If the similarity score matches the determined similarity score e.g. 0.5, add it to similar grievances list to display them in *similar cases suggestion* area. Note: For each grievance document, apply preprocessing steps on it before passing it to similarity method.

B. Evaluating Text Similarity and Classifier Modules

The second part of our performance improvement, is to give a hint to the system user, about how to solve this Grievance issue based on previous issues saved in Knowledge base (KB). When a Grievance is coming, it is analyzed to be compared we calculated recall, precision and F-measure to evaluate our modules, and determined what is the best F-Measure based on similarity score [19].

- **Precision:** is the number of correct results divided by the number of all returned results Equation (3).

$$\text{precision} = \frac{| \{ \text{relevant documents} \} \cap \{ \text{retrived documents} \} |}{| \{ \text{retrived documents} \} |} \quad (4)$$

- **Recall:** is the number of correct results divided by the number of results that should have been returned.

$$\text{recall} = \frac{| \{ \text{relevant documents} \} \cap \{ \text{retrived documents} \} |}{| \{ \text{relevant documents} \} |} \quad (5)$$

- **F-measure:** is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: F-Score (F1 score) is the harmonic mean of precision and recall:

$$F_{\text{measure}} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

For Example: Assume we have a dataset contains 160 records on a specific issue. A search query was running on that issue and 90 records were retrieved. And of that 90 records retrieved, 55 were relevant. We calculate the precision and recall scores for the search.

The number of relevant records retrieved=55

The number of relevant records not retrieved=160-55

The number of irrelevant records retrieved=90-55

Recall=(55 / (55 + 105)) * 100% => 55/160 * 100%=34%

Precision=(55 / (55 + 35)) * 100% => 55/90 * 100%=61%

F-measure=2 * 34*61/34+61=43.55

C. Solution Hint Module Results

Here is the automatic hint for solution given by system by using three similarities score Range values [0.40, 0.50 and 0.60] and view the results as shown in Table 8.

TABLE VIII. SIMILARITY SCORE FOR EACH GRIEVANCE

TASKID	MESSAGE	SCORE
6300	Iam an Egypt and Ilove and prefer using....	0.57
5938	Iam contacting you regarding the return of luggage	0.55
1870	Iam .and my wife is .and we had two distant seats	0.65
4125	Iam writing regarding a delayed /cancelled jou....	0.76
7570	I booked two Tickets from Luxor to Cairo....	0.5
7141	I contacted this airline about the bug that...	0.33
3283	I need some help I had a flight with you with pooping	0.45
4721	I recently returned to (Istanboul) from (Kanada)	0.65
6164	I traveled from London to Lagos via Cairo	0.57
4625	I want to complain about stolen goods	0.67
6650	I was on flight number MS 985 from cairo....	0.51
6302	I was on MS 996 from Tornto to Cairo o....	0.53
5779	I was on MS 800 Paris- Cairo on 30th Nov,t...	0.54
2612	I was on the flight back from Amstrdam....	0.47
5959	I was on Yestrdays flight from Cairo to R....	0.61
5959	I was on Yestrdays flight from Cairo to R....	0.62
6319	I was recently due to fly from Alexandria t....	0.58
6439	know this may not be the correct email for Luggage	0.55
2176	Lun 19/6/17,.....,the scritto:}.....	0.48
7580	MS778 London-Cairo MS20 Cairo-Sharm....	0.54
6511	My flight from Egypt was delayed . I had detriment...	0.73
7010	My flight was delayed for 3+ hours which effect.....	0.75
7713	My money was stolen out of my backpac....	0.64

If the similarity scores smaller than 0.4, the result may get irrelevant answers. If similarity score greater than 0.6, the result may get less answers similar. However, we examined it by using the best similarity scores (0.4, 0.5 and 0.6) results were compared as shown in the following table (See Table 9).

According to results of our experiments

1) If similarity score was 0.60, the precision increased and recall decreased,

2) If similarity score equals 0.50, the precision decreased and recall increased. For gaining best F-Measure (69.45%) at similarity score (0.50), you find many statements in the short incoming message than long one. The results are shown in Table 9.

TABLE IX. SIMILARITY SCORE CALCULATION

Score	Precesion	Recall	F-Measure
0.60	82.25%	39.63%	51.34%
0.50	70.59%	69.33%	69.45%
0.40	39.64%	71.95%	55.37%

VIII. CONCLUSION AND FUTURE DIRECTIONS

In this work, we implemented an automated grievances system that integrates some text mining techniques. EGYPTAIR data set were used in this work. All of them came from the previous grievances submitted in the period from 2017 to 2018. The data set included one thousand grievances that belong to 6 categories used for learning. This work examined automatic text categorization of grievances documents by using set of grievances methods (SVM, KNN and decision tree) and according to the results we noticed that KNN achieved the best average classification accuracy and then SMO. Final recall and precision results were 94.69% and 94.96% respectively. Also we conducted several experiments to test solution hint module by similarity score calculation. Opinion grievances is a future direction that can help to discover and extract useful and profound knowledge resources using the concept level sentiment analysis, improving customer loyalty by providing a customer behavior model based on data mining algorithms. Moreover, analyze sentiments (positive or negative) from social datasets and automatically predict sentiment intensity scores to improve services. We will work in the future also to improve the tool in order to enlarge its features, such as covering pdf, and other file format. Also, give the system the ability to detect the type of device from which the cases has been sent in order to handle the request in effective manner and give the user the ability to browse this web tool based on the capabilities of such device. Moreover, we must make our website secure by limiting access for some features in the website to the anonymous user and allow these features to the granted users only. The tendency of using neural network method for text categorization and measuring similarity issues is very high in the new articles. In the future, it can be more focused on identifying neutral comments and improving the performance of the models by using the convolution neural network method on huge corpus.

ACKNOWLEDGMENT

Special thanks to the members of IT department and customer service department in Egypt Air holding company for providing us with customer grievances data.

REFERENCES

- [1] Barbier, G., & Liu, H. (2011). Data mining in social media. In *Social network data analytics* (pp. 327-352). Springer, Boston, MA.
- [2] McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: the management revolution. *Harvard business review*, 90(10), 60-68.
- [3] Yee Liau, B., & Pei Tan, P. (2014). Gaining customer knowledge in low cost airlines through text mining. *Industrial Management & Data Systems*, 114(9), 1344-1359.
- [4] Chen, C. K., Shie, A. J., & Yu, C. H. (2012). A customer-oriented organizational diagnostic model based on data mining of customer-complaint databases. *Expert Systems with Applications*, 39(1), 786-792.
- [5] ElMessiery, AM(2016). Natural Language Techniques for Decision Support Based on Patient Complaints.
- [6] Maia, P., Carvalho, R. N., Ladeira, M., Rocha, H., & Mendes, G. Application of text mining techniques for classification of documents: a study of automation of complaints screening in a Brazilian Federal Agency.

- [7] Shehata, S., Karray, F., & Kamel, M. S. (2010). An efficient model for enhancing text categorization using sentence semantics. *Computational Intelligence*, 26(3), 215-231.
- [8] Al Najjar, M., & Alaa, E. H. (2013). Automated Complaint System Using Text Mining Techniques (Doctoral dissertation MS Thesis, IT Dept, IUG univ, Gaza).
- [9] Yakut, I., Turkoglu, T., & Yakut, F. (2015). Understanding customers' evaluations through mining airline reviews. *arXiv preprint arXiv:1512.03632*.
- [10] Tang ,H., Tan ,S.,et. Al.,(2009), Asurvey on sentiment Detection of Reviews. *Exper Systems*.
- [11] Niharika, S., Latha, V. S., & Lavanya, D. R. (2012). A survey on text categorization. *International Journal of Computer Trends and Technology*, 3(1), 39-45.
- [12] Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press
- [13] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- [14] Song, M. (Ed.). (2008). *Handbook of research on text and web mining technologies*. IGI global
- [15] Kopackova, H., Komarkova, J., & Sedlak, P. (2008). Decision making with textual and spatial information. *WSEAS Transactions on Information Science and Applications*, 5(3), 258-266.
- [16] Hakim, A. A., Erwin, A., Eng, K. I., Galinium, M., & Muliady, W. (2014, October). Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In *2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 1-4). IEEE
- [17] Thada, V., & Jaglan, V. (2013). Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology*, 2(4), 202-205.
- [18] Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454). Springer Science & Business Media.
- [19] Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., & Scuse, D. (2013). *Waikato Environment for Knowledge Analysis (WEKA) Manual for Version 3-7-8*. The University of Waikato, Hamilton, New Zealand. ISO 690.
- [20] Mathew, K., & Issac, B. (2011, December). Intelligent spam classification for mobile text message. In *Computer Science and Network Technology (ICCSNT), 2011 International Conference on* (Vol. 1, pp. 101-105). IEEE.
- [21] Thornton, E. A. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms." *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.
- [22] Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review*, 41(4), 70-73C.
- [23] Bresfelean, V.P. (2007, June). Analysis and predictions on students' behavior using decision trees in Weka environment. In *Proceedings of the ITI* (pp. 25-28).
- [24] Anyanwu, M. N., & Shiva, S. G. (2009). Comparative analysis of serial decision tree classification algorithms. *International Journal of Computer Science and Security*, 3(3), 230-240
- [25] Spyromitros, E., Tsoumakas, G., & Vlahavas, I. (2008, October). An empirical study of lazy multilabel classification algorithms. In *Hellenic conference on artificial intelligence* (pp. 401-406). Springer, Berlin, Heidelberg.
- [26] Wu, J., Pan, S., Zhu, X., Cai, Z., Zhang, P., & Zhang, C. (2015). Selfadaptive attribute weighting for Naive Bayes classification. *Expert Systems with Applications*, 42(3), 1487-1502.
- [27] Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3), 27.