FEATURE ENGINEERING WITH SENTENCE SIMILARITY USING THE LONGEST COMMON SUBSEQUENCE FOR EMAIL CLASSIFICATION

Aruna Kumara B[1*], Mallikarjun M Kodabagi[2]

[1,2]School of Computing and Information Technology, REVA University, 560064 Bengaluru, India

Email: arunakumara.b@reva.edu.in[1]*(corresponding author), mallikarjun.mk@reva.edu.in[2]

DOI: https://doi.org/10.22452/mjcs.sp2022no2.6

ABSTRACT

Feature selection plays a prominent role in email classification since selecting the most relevant features enhances the accuracy and performance of the learning classifier. Due to the exponential increase rate in the usage of emails, the classification of such emails posed a fitting problem. Therefore, there is a requirement for a proper classification system. Such an email classification system requires an efficient feature selection method for the accurate classification of the most relevant features. This paper proposes a novel feature selection method for sentence similarity using the longest common subsequence for email classification. The proposed feature selection method works in two main phases: First, it builds the longest common subsequence vector of features by comparing each email with all other emails in the dataset. Later, a template is constructed for each class using the closest features of emails of a particular class. Further, email classification is tested for unseen emails using these templates. The performance of the proposed method is compared with traditional feature selection methods such as TF-IDF, Information Gain, Chi-square, and semantic approach. The experimental results showed that the proposed method performed well with 96.61% accuracy.

Keywords: ***Email classification, Feature Engineering, Sentence similarity, Similarity measure, Imbalanced learning, Feature selection.***

1.0 INTRODUCTION

Since the internet evolved, a data explosion happening over the internet, which lead to the enormous amount of natural data being added daily to the internet [1]. Furthermore, human literature in different formats is digitalized and available in digital libraries, and this huge amount of data is getting stored in natural data format. This necessitates the use of natural language processing (NLP) [2] techniques essential in various applications such as classification [3], clustering [4], question-answering [5], prediction [6], information-retrieval [7], and plagiarism checking [8].

Classification plays a crucial role in NLP as many problems are regarded as text classification tasks such as email classification [9], news classification [10], sentiment analysis [11], and so on. Traditional learning models for email classification frequently rely on statistics-based [12] and rule-based [13] characteristics. Feature selection approaches, on the other hand, become expensive when creating artificial feature sets [14]. However, choosing appropriate features for email categorization in a different application area is a difficult challenge, and feature selection is critical in machine learning and natural language processing problems. The use of irrelevant features can cause email to be misclassified and has an impact on the learning classifier's accuracy. It also has the potential to raise the classifier's false positive rate. As a result, the selection of the most relevant features is prominent in classification methods.

Chi-square (CHI-2) [15], latent semantic analysis (LSA) [16], information gain (IG) [17], principle component analysis (PCA) [18], similarity measure [19], and other feature selection approaches are commonly employed in email categorization. The relevant features were chosen using several similarity techniques such as word-based similarity measure [20], pairwise similarity measure [21], and sentence-based similarity measure [22]. Sentence similarity-based feature selection strategies measured the distance between two documents or sequences using existing similarity metrics such as cosine [23], euclidian [24], and Jaccard [25]. However, in the case of multi-class classification issues, these approaches performed poorly [26].

Distinct from such previous research works, the proposed sentence similarity using the longest common subsequence (SSLCS) method, retains the most relevant features which are used to generate templates for each category. Each template stores the emails which are close to each category as a final feature set. The method works in two main phases: i) build the longest common subsequence (LCS) vector of features by comparing each email with all other emails in dataset ii) construct a template for each class based on the most similar

features of emails from that class. Further, these templates are also used to test email classification for unseen emails.

In this paper, a novel feature selection technique based on sentence similarity using the longest common subsequence is proposed. The significant contributions of this work are:

- The performance of classifiers will be affected by the variant features selected. To address the issue of imbalanced class distribution [27], a novel longest common subsequence (LCS) method is proposed to obtain the most relevant sequence as a feature set.
- Due to the exponential increase in the ratio of emails, there is a frequent change in the email content which leads to lesser effectiveness of features during classification. A novel feature selection method SSLCS based on the dynamic programming [28] concept is proposed to enhance the effectiveness of features selected.
- Various email datasets such as Enron [29] and TREC [30] are available publicly for researchers for email classification [31]. However, only one study [32] used the real-time dataset for phishing email classification. In this work, a real-time dataset is constructed and used for email classification to overcome the dataset barriers.

The rest of this paper is structured as follows: section 2 gives the related work carried out on various feature selection techniques used for email classification. section 3 discusses the proposed method SSLCS to select the most relevant features for email classification. Section 4 describes the datasets used in this study and also discussed the performance of various learning classifiers. Finally, the work is concluded in section 5.

2.0 RELATED WORK

Some of the recent studies done on various feature selection techniques in text classification/email classification systems are detailed below:

Shan, Guangxu et al. [33] proposed a novel incremental learning strategy to solve challenges in text classification. The method consists of four components: a student model, a reinforcement learning (RL) module, a teacher model, and a discriminator model. In the student model, extraction of features from the text data is done, then the second model filters the results into multiple models of the student. Later third module reclassifies the results to get the final category. Finally, the fourth model filters the student models based on similarity to avoid increasing an unlimited number of samples. The major advantage of this method takes a shorter training time compared to one - time model, as it needs to train only a new model of the student without changing the existing student models. Also, this method obtains feedback during application. Results showed that this model outperforms various traditional methods by reducing the training time almost by 80%.

Oghbaie and zanjireh [34] introduced a pairwise document similarity measure to find the number of terms that appeared in at least one of the documents. In this method, a pairwise comparison between two documents is done based on the term weights and the number of terms present in both documents. It was observed that, if the number of terms like absence-presence was more in the document, with this note authors concluded that both documents were not similar. Also, the result part of the paper said that the performance was mainly dependent on the type of document collection and the term weight scheme utilized.

An effective and efficient similarity measure based on set theory for text classification and clustering is introduced by amen and Abdalla [35]. A comparative study is carried out to perform classification and clustering using seven similarity measures. The results demonstrated that the proposed set theory-based similarity measure gave the best performance in all classification and clustering techniques used in the work.

The article [36] proposed a model to classify Chinese text by enhancing the features. This work first extracts the preliminary semantic features. Later, a non-equilibrium bidirectional long short-term memory was applied to enhance the weight of the text which had essential semantics, and additionally improves the performance of the vital feature during classification. The method substantially improves the precision in Chinese text classification, and also the model shows a strong ability in recognizing Chinese text features.

Usage of Association for Computing Machinery taxonomy (ACM) is proposed [37] to get the similarity between two scientific documents, where each document is made by a group of keywords. A genetic algorithm was used to enhance the index validity of clusters and to find the similarity between documents Floyd's and Warshall's algorithms were used. Based on the data, it can be inferred that the algorithm that performed the best was the one that used uniform crossover.

A study proposed to use the eigenvalues for semantic sentence similarity [38]. Three features (sentence-pair-aware word similarity, cosine similarity of vector representation, and semantic textual similarity) were introduced in the study, and the performance was studied on various datasets. Combining these features with the term frequency-inverse document frequency (TF-IDF) improved the performance of the learning classifier.

Hadeel M. Saleh presented an efficient feature selection algorithm for spam email classification [39]. The algorithm was the combination of two algorithms (particle swarm optimization and artificial bees colony). The results showed better performance over traditional feature selection methods in selecting the subset of features.

T. Poonkodi and Dr.S. Sukumaran proposed an enriched firefly optimization algorithm [40] to filter spam emails by selecting suitable features. The method was used to select suitable features from upper dimensional space using the fitness function. The results showed remarkable improvement compared to sine-cosine algorithms.

By evaluating the semantic qualities of words, an effective email filtering approach based on semantic methods was presented [41] to overcome the problem of the high dimensionality of features. The results of the experiments revealed significant performance with great accuracy in a short amount of time.

After a thorough literature review, it was discovered that many feature selection techniques for email classification have been developed. However, less research has been conducted on the feature selection technique for a multi-class imbalanced [42-43] email classification system on a real-time dataset. As a result, efficient feature selection techniques on real-time datasets are required to classify emails into various categories with improved classification accuracy. In this paper, an efficient feature selection technique based on sentence similarity using the longest common subsequence for email classification is proposed.

3.0 PROPOSED METHODOLOGY

The proposed methodology employs an efficient feature selection method for the classification of email data. The method works in four main phases: Data Collection and Preprocessing, Extraction of Features, Template Construction, and Classification. To begin, raw email data was collected and a dataset was created. Following that, several preprocessing techniques were used to clean up the raw data and prepare it for future processing. The proposed feature selection method was then used to design a template and select the most relevant features. Finally, machine learning classifiers were used to categorize previously unseen data into predetermined classes. The proposed system architecture is illustrated in Fig. 1. The following sections provide a detailed description of the system architecture.

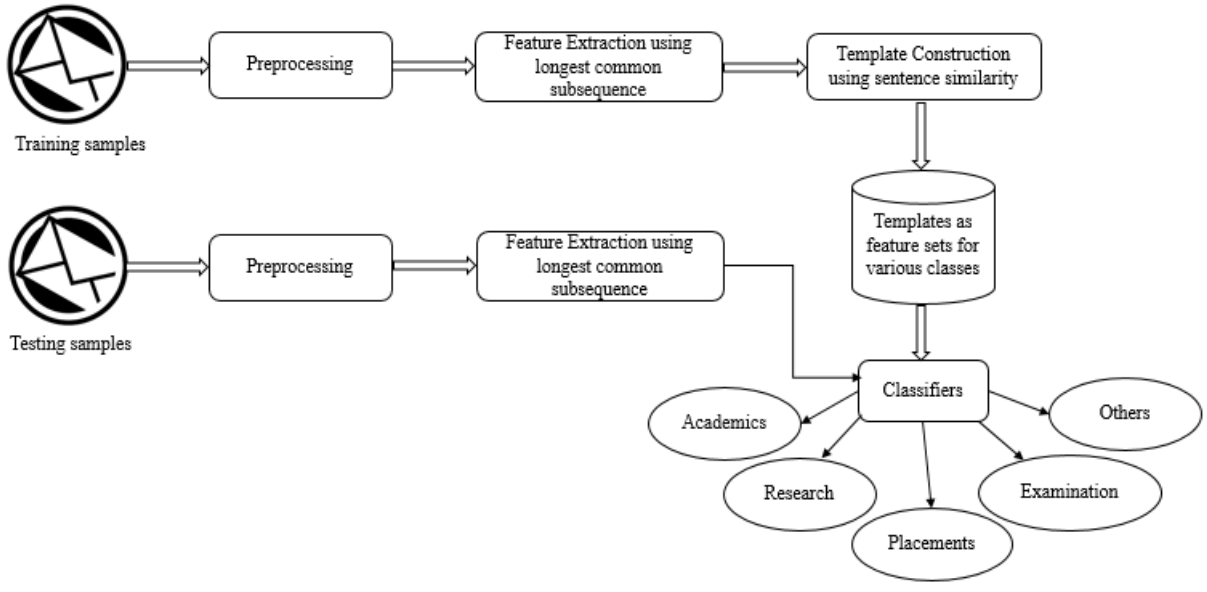


Fig. 1: The proposed system architecture

3.1 Data Collection and Preprocessing

The real-time email samples from the outlook domain of the Rukmini Educational Vision Academy (REVA) University database were collected and constructed into an email dataset referred to as the REVA dataset with five different categories: Examination, Academics, Research, Placements, and Others. These email samples were collected from the 2017–2020 years under the terms and service conditions of the Internal Quality

Assurance Cell (IQAC) of REVA University. Further, data preprocessing methods were applied to clean the raw data following data gathering. Various preprocessing methods such as lowercase conversion, stop-word removal (SWR), removal of email signature (RES), word tokenization, and stemming were applied to remove irrelevant data [44]. A detailed explanation is given below with an example.

EXAMPLE: Let us consider two email sequences A and B to show the working of the proposed methodology:

A = {Below mentioned students have not submitted Exam Application forms} …….. (1)

B = {Congratulations to the below-mentioned students who are selected for the IBM Campus Drive} ……. (2)

The results after preprocessing on equations (1) and (2) were stored in lists A' and B' as follows:

A' = {“below”, “mention”, “student”, “submit”, “exam”, “application”, “form”} ……. (3)

B' = {“congratulation”, “below”, “mention”, “student”, “select”, “ibm”, “campus”, “drive”} ……..…. (4)

For operational purpose, following representations are used in the paper. First element in the list A' (below) is represented as a_1, second element (mention) as a_2 and so on. Similarly, for the list B', first element (congratulation) represented as b_1, second element (below) as b_2 and so on. So the final lists are represented as:

$$A' = \{a_1, a_2, a_3, \ldots a_m\} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (5)$$

$$B' = \{b_1, b_2, b_3, \ldots b_n\} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

3.2 Extraction of Features

In this phase, the common subsequences were determined between every pair of emails and returned the longest among all of them. The algorithm FEATURE_EXTRACTION_USING_LCS was designed to identify the longest subsequence of an email sentence that is common among all sentence sequences in a set of sentences of another email. The subsequences in a sentence in the email are not required to be in the same order as in the sentence of another email that is used for comparison. This algorithm works as follows: it takes each element from an email and compares it with all elements in another email. During the comparison, if both are equal, append it to the previous element in the LCS table. Otherwise, it appends the largest of the upper row element or left side element in the LCS table. This procedure is continued until all elements in one email are compared with another email. Finally, it returns the longest common subsequence between two emails.

Algorithm FEATURE_EXTRACTION_USING_LCS (a[0..m-1], b[0….n-1])

INPUT: Two Email Sequences a [0…..m-1] b [0……n-1]
OUTPUT: Returns the longest common subsequence

```
m←len(a)
n←len (b)
for w← 0 to m-1 do
        for v← 0 to n-1 do
                if (w=0 or v=0) then
                        LCS [w, v] ← 0
                else if (a[w-1] = b[v-1]) then
                        LCS [w, v] ← LCS [w-1, v-1] + 1
                else
                        LCS [w, v] ← max(LCS[w-1, v], LCS[w, v-1])
return LCS[m,n]
```

The computation of the longest common subsequence between two emails sequences (i.e., emails a and b) is illustrated below.

Generate an LCS vector with (m+1)*(n+1) dimension (where m – indicates the length of A' and n – indicates the length of B'). Initialize the a_0 row and b_0 column with Ø to represent an empty sequence. Table 1 is used to store the longest common subsequence for each step of the calculation. When a non-empty sequence is compared with an empty sequence, the LCS will be always an empty sequence.

Table 1:Initial LCS vector

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_0	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
a_1	Ø								
a_2	Ø								
a_3	Ø								
a_4	Ø								
a_5	Ø								
a_6	Ø								
a_7	Ø								

LCS (a_1, b_1) is identified by comparing the first element in the sequence, a_1 (below) and b_1 (congratulation) does not match. Hence, the LCS gets the longest of two sequences using a second property of equation (3) i.e., LCS (a_1, b_0) and LCS (a_0, b_1). By referring to Table 1, both are empty; so LCS (a_1, b_1) is also empty. LCS (a_1, b_2) is identified by comparing the a_1 (below) with b_2 (below). Both a_1 and b_2 match. So "below" is appended to the upper left sequence. LCS (a_0, b_1) is Ø that gives (Øa_1) which in turn gives a_1 (i.e., below). For LCS (a_1, b_3), "below" and "mention" does not match. The sequence in the upper row is empty and the sequence in the left contains only one element (below). Hence, selecting LCS (a_1, b_2) is a_1 (i.e., below). Similarly, for LCS (a_1, b_4) is a_1 (below). For LCS (a_1, b_5) is a_1 (below). For LCS (a_1, b_6) is a_1 (below). For LCS (a_1, b_7) is a_1 (below). For LCS (a_1, b_8) is a_1 (below). All these values are updated in Table 2.

Table 2: LCS vector of a_1 row

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_0	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
a_1	Ø	Ø	a_1	a_1	a_1	a_1	a_1	a_1	a_1
a_2	Ø								
a_3	Ø								
a_4	Ø								
a_5	Ø								
a_6	Ø								
a_7	Ø								

For LCS (a_2, b_1), "mention" and "congratulation" does not match. Hence the LCS (a_2, b_0) is Ø and LCS (a_1, b_1) is Ø. So, LCS (a_2, b_1) is also Ø. For LCS (a_2, b_2), "mention" and "below" does not match. Hence, the LCS (a_2, b_1) is Ø and LCS (a_1, b_2) is a_1. So, LCS (a_2, b_2) is Ø. For LCS (a_2, b_3), "mention" and "mention" match. So, a_2 is appended to the upper left sequence which is a_1 & a_2 ("below", "mentioned").For LCS (a_2, b_4), "mention" does not match with "student". LCS (a_2, b_3) is a_1 & a_2. LCS (a_1, b_4) is a_1. Hence LCS (a_2, b_4) is a_1 & a_2. Similarly, for LCS (a_2, b_5) is a_1 & a_2. For LCS (a_2, b_6) is a_1 & a_2. For LCS (a_2, b_7) is a_1 & a_2. For LCS (a_2, b_8) is a_1 & a_2. All these values of a_2 row are updated in the Table 3.

Similarly, LCS for rows a_3, a_4, a_5, a_6, and a_7 were computed as discussed above. Table 4 gives the LCS vector for email sequences A and B. Calculating the LCS of a particular row requires the values in the previous row and current row. For long sequences, storing all the subsequences becomes a too lengthy and tedious task. So, the length of the subsequence is stored in the LCS table instead of actual subsequences as shown in Table 5, which is later used to construct the template for each class.

Table 3: LCS vector of a_2 row

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_0	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
a_1	Ø	Ø	a_1	a_1	a_1	a_1	a_1	a_1	a_1
a_2	Ø	Ø	Ø	a_1 a_2	a_1 a_2	a_1 a_2	a_1 a_2	a_1 a_2	a_1 a_2
a_3	Ø								
a_4	Ø								
a_5	Ø								
a_6	Ø								
a_7	Ø								

Table 4: The LCS vector for emails A and B

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_0	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
a_1	Ø	Ø	a_1	a_1	a_1	a_1	a_1	a_1	a_1
a_2	Ø	Ø	Ø	a_1 a_2	a_1 a_2	a_1 a_2	a_1 a_2	a_1 a_2	a_1 a_2
a_3	Ø	Ø	Ø	a_1 a_2	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3
a_4	Ø	Ø	a_1 a_2	a_1 a_2	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3
a_5	Ø	Ø	a_1 a_2	a_1 a_2	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3
a_6	Ø	Ø	a_1 a_2	a_1 a_2	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3
a_7	Ø	Ø	a_1 a_2	a_1 a_2	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3	a_1 a_2 a_3

Table 5: LCS with the length of subsequences

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
a_0	0	0	0	0	0	0	0	0	0
a_1	0	0	1	1	1	1	1	1	1
a_2	0	0	0	2	2	2	2	2	2
a_3	0	0	0	2	3	3	3	3	3
a_4	0	0	2	2	3	3	3	3	3
a_5	0	0	2	2	3	3	3	3	3
a_6	0	0	2	2	3	3	3	3	3
a_7	0	0	2	2	3	3	3	3	3

3.3 Template Construction

For each class, a template is created in this phase. As discussed in section 3.2, an LCS vector is constructed for every two emails in the dataset. As a feature set for a class, the score derived from this technique, as well as real subsequences, are kept in a template is known as Feature Weight (FW[][]) vector. These features are then placed in the non-increasing order of the acquired score. Finally, the top N features for each class are chosen as the final feature set and stored in a template. The algorithm TEMPLATE_CONSTRUCTION_USING_SSLCS was designed to measure the similarity between every two emails in the dataset based on sentence similarity using the longest common subsequence. This algorithm works as follows: first, unique words from each email of the dataset are extracted through word tokenization. Then, the algorithm FEATURE_EXTRACTION_USING_LCS was applied and stored the feature weight or score of each email in the feature weight vector FW. Later, all the

features are arranged in the non-increasing order of their feature weight. Finally, top N features are selected as a feature set for each class

Algorithm TEMPLATE_CONSTRUCTION_USING_SSLCS (E[0…n-1])

INPUT: Email dataset E
OUTPUT: Most relevant feature set FW[]

Step1: Extract unique words from each email of the dataset.

Step2: Compute the LCS between every two emails in the dataset using the Algorithm **FEATURE_EXTRACTION_USING_LCS ( )** and store the score obtained in the feature vector FW[]

Step3: Arrange the features in non-increasing order of their feature weight using the Merge Sort technique

Step4: Select the top N features (above 80% similarity) to construct the template with a final feature set

3.4 Classification

In this phase, email samples were classified into one of the predefined categories using a machine learning classifier. The algorithm E-MAIL_CLASSIFICATION was designed to classify the emails into one of the predefined categories. First, various preprocessing methods on the dataset were applied to remove irrelevant information and to reduce the size of the dataset. Later, the proposed feature selection method (Algorithm FEATURE_SELECTION_USING_SSLCS) was applied to the training dataset to extract the features and construct the template as a feature vector for each class. Finally, unseen emails were classified into the respective predefined categories using the template.

Algorithm E-MAIL_CLASSIFICATION(E[0…n-1])

INPUT: E-mail Dataset E[0…N-1]

OUTPUT: Classifies each incoming email into one of the predefined classes

Step1: Apply Pre-processing techniques on Training and Testing Dataset
- Tokenization
- Lowercase conversion
- Stop word removal
- E-mail Signature removal
- Stemming

Step2: Apply the proposed feature selection technique on training data
(Algorithm TEMPLATE_CONSTRUCTION_USING_SSLCS(E[…n-1])
Step3: Apply machine learning classifiers (NB, LSVC, LR, RF) on training data
Step 4: Apply machine learning classifiers (NB, LSVC, LR, RF) on unseen data.

4.0 RESULTS AND DISCUSSION

In this section, the evaluation of SSLCS was measured in two sets of experiments. In the first experiment, the SSLCS was applied to the real-world email dataset from outlook email users (REVA email dataset) and the performance of various classifiers was measured. In the second experiment, the proposed method was applied to the standard email dataset TREC-2007 corpus, and the performance of different classifiers was measured.

In the first experiment, primary datasets were created using the REVA University email database. The dataset contains 3000 email samples and was categorized by the human expert into five different categories like Academics, Placement, Examination, Research, and Others. For experimental purposes, datasets were distributed in a 70:30 ratio as training and testing samples. The dataset contains the uneven class distribution of emails. The distribution of emails into various classes is depicted in Fig.2.

The performance of the proposed feature selection method SSLCS is compared with traditional feature selection methods used for email classification. The traditional feature selection methods considered are TF-IDF [45], CHI-2 [46], and IG [47]. Linear Support Vector Classifier LSVC [48] was considered to measure the performance of the proposed feature selection method and traditional feature selection methods on multi-class

imbalanced data. Also, accuracy [49], precision [50], recall [51], and f-measure [52] were used to assess the performance of the LSVC classifier over feature selection methods.

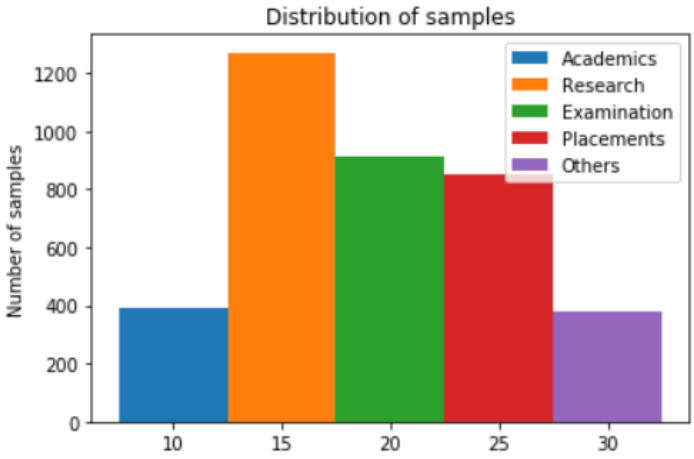
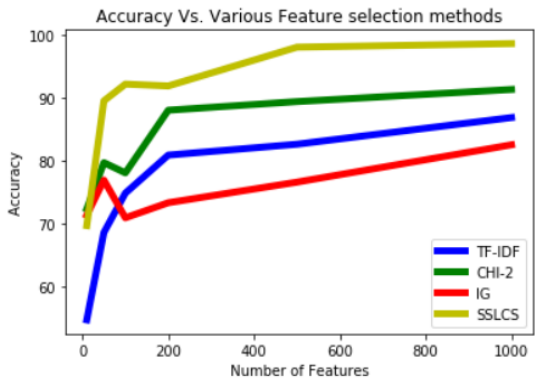
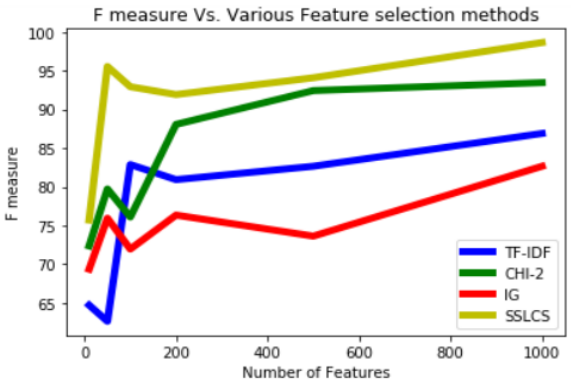


Fig. 2: Email distribution into various classes

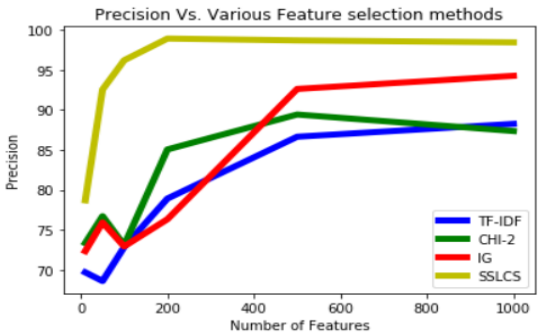
Fig. 3 shows the performance of SSLCS and traditional feature selection methods over various performance measures. IG recorded 91.36% accuracy and SSLCS records an accuracy value of 95.66%. Also, it is observed that the proposed method recorded the accuracy value in increasing order as the number of features selected increased. Fig.3 (a) shows the accuracy of all traditional feature selection methods and proposed methods. The results demonstrated that the SSLCS method dominates over traditional methods with 96.46%. Fig.3 (b) demonstrated the f-measure overall feature selection methods. The proposed method outperforms traditional methods with 94.89%. Fig.3 (c) illustrated the precision of all the feature selection methods used in this work. From the results, it is noted that SSLCS dominates over TF-IDF, CHI-2, and IG with 95.45%. Fig. 3 (d) showed the recall over traditional feature selection methods. It was observed that SSLCS recorded a recall value of 94.81%.



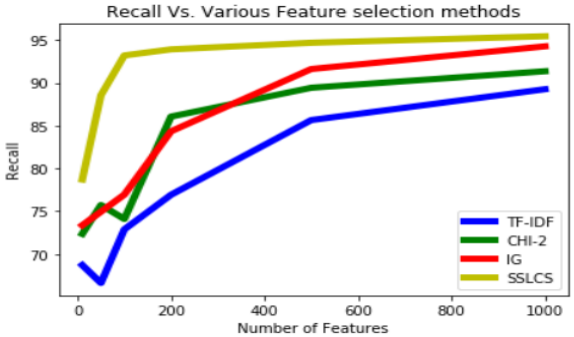
(a): Accuracy over various feature selection methods



(b): F measure over various feature selection methods



(c): Precision over various feature selection methods



(d): Recall over various feature selection methods

Fig. 3: Performance of different feature selection methods over various performance measures

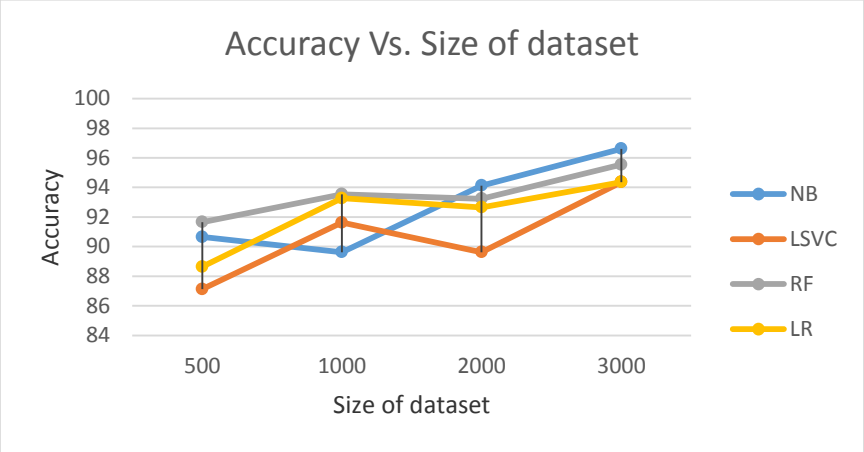


Fig. 4: Accuracy performance of SSLCS over different classifiers on REVA dataset

The proposed feature selection method was evaluated over four learning classifiers such as LSVC, Naïve Bayes (NB) [53], Logistic Regression (LR) [54], and Random Forest (RF) [55]. Fig. 4 shows the performance of the proposed method over different learning classifiers. The experimental results showed that RF dominates over other classifiers with 91.65% accuracy when 500 email samples were considered. LR improved its accuracy by 93.26% compared to other learning classifiers when 1000 email samples were considered for the experiment. NB outperforms other classifiers with 94.12% and 96.61% when 2000 and 3000 email samples were considered respectively. Also, it was observed that the performance of all classifiers was improved as the dataset size increased.

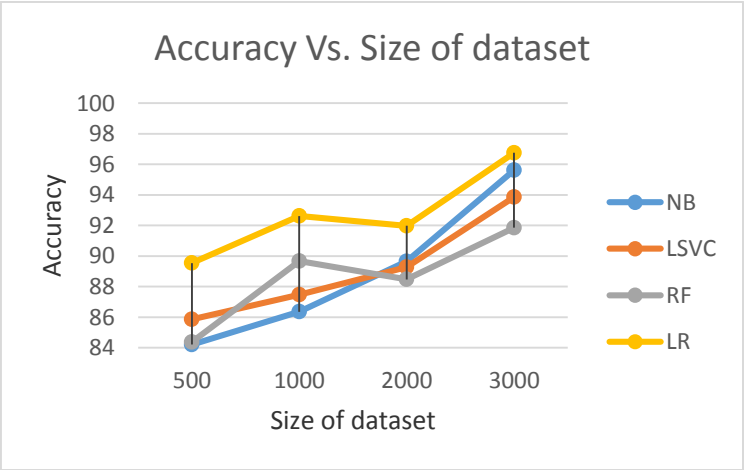


Fig. 5: Accuracy performance of SSLCS over different classifiers on the TREC dataset

In the second experiment, the proposed method was applied to the benchmark dataset TREC-2007 corpus. Fig.5 recorded the performance of SSLCS over different classifiers on the TREC dataset. The experimental results showed that LR performed well compared to other classifiers (NB - 84.21%, LSVC - 85.85%, RF - 84.36%) with 89.54% accuracy when 500 email samples were considered. It was observed that LR achieved 96.61% accuracy when 3000 samples were considered and dominates others. Also, it was observed that the performance of all classifiers was improved as the dataset size increased. Finally, it was noted that the proposed method performed well on the real-world dataset as well as on the standard email dataset.

Fig. 6 shows the accuracy comparison between proposed work and existing work (related work in [14]). From the results, it was observed that, in the existing work, classifiers NB, LSVC, and RF recorded 91.12%, 94.15%, and 91.62% respectively. Also, the results showed that in the proposed work NB recorded the highest accuracy 96.61%, LSVC recorded 94.36%, and RF achieved 95.54%. From the results, it was observed that the proposed work dominated over existing work by performing well for all three classifiers used.

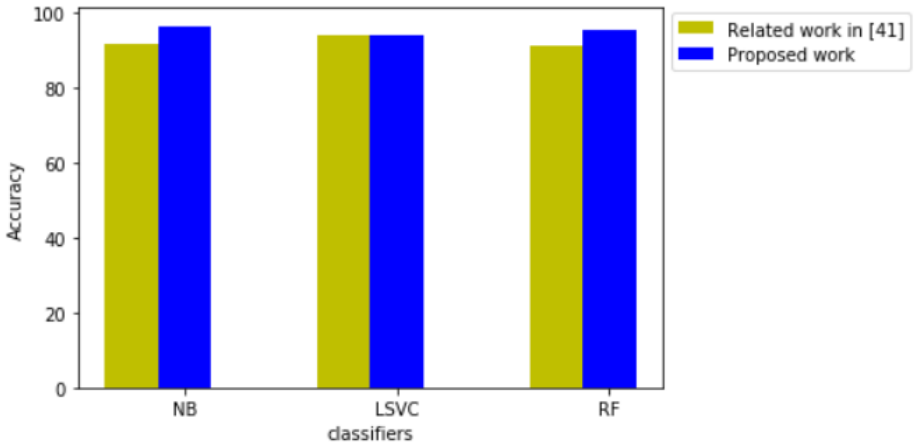


Fig. 6: Accuracy comparison between proposed work and related work in [41]

5.0 CONCLUSION

This paper proposed a novel feature selection method based on sentence similarity using the longest common subsequence. For each email in the dataset, this method constructs the longest common subsequence vector. Later, the template was built for each class using the closest features of that class. Finally, using these templates, email classification was tested on unseen emails. Two sets of experiments were carried out to assess the performance of the proposed work. The proposed method's performance was compared with TF-IDF, CHI-2, IG, and semantic method feature selection methods. The proposed work outperforms the existing method with an accuracy of 96.61%. The classifier LSVC achieved 94.36% accuracy, LR achieved 92.81% accuracy, RF achieved 95.54% accuracy, and NB records 96.61% accuracy. Future work focuses on sentence similarity measures using deep learning computational methods.

REFERENCES

[1] A. Borg, M. Boldt, O. Rosander and J. Ahlstrand, "E-mail classification with machine learning and word embeddings for improved customer support", *Neural Computing and Applications*, No. 33, 2021, pp. 1881-1902. doi: 10.1007/s00521-020-05058-4.

[2] F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López and M. T. Martín-Valdivia, "A Multi-Task Learning Approach to Hate Speech Detection Leveraging Sentiment Analysis", in *IEEE Access*, Vol. 9, 2021, pp. 112478-112489. doi: 10.1109/ACCESS.2021.3103697.

[3] G. Mustafa, M. Usman, L. Yu, M. T. Afzal, M. Sulaiman and A. Shahid, "Multi-label classification of research articles using Word2Vec and identification of similarity threshold", *Scientific Reports*, No.11, 2021, 21900. doi: 10.1038/s41598-021-01460-7.

[4] P. Nerurkar, A. Shirke, M. Chandane and S. Bhirud, "Empirical Analysis of Data Clustering Algorithms", in *Procedia Computer Science*, Vol. 125, 2018, pp. 770-779. Doi: 10.1016/j.procs.2017.12.099.

[5] E. Dimitrakis, K. Sgontzos and Y. Tzitzikas, "A survey on question answering systems over linked data and documents", *The Journal of Intelligent Information Systems*, No.55, 2020, pp. 233–259. doi: 10.1007/s10844-019-00584-7.

[6] S. D. A. Bujang, A. Selamat, R. Ibrahim, O. Krejcar, E. Herrera-Viedma, H. Fujita and N. A. M. Ghani, "Multiclass Prediction Model for Student Grade Prediction Using Machine Learning", in *IEEE Access*, Vol. 9, 2021, pp. 95608-95621. doi: 10.1109/ACCESS.2021.3093563.

[7] S. Kadhe, B. Garcia, A. Heidarzadeh, S. E. Rouayheb and A. Sprintson, "Private Information Retrieval With Side Information", in *IEEE Transactions on Information Theory*, Vol. 66, No. 4, 2020, pp. 2032-2043. doi: 10.1109/ALLERTON.2017.8262860.

[8] C.B. Carter and C. F. Blanford, "Plagiarism and detection". *Journal of Material Science*, No.51, 2016, pp. 7047–7048. doi: 10.1007/s10853-016-0004-7.

[9] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi and O. E. Ajibuwa "Machine learning for email spam filtering: review, approaches and open research problems", *Heliyon*, Vol. 5, No. 6, 2019, e01802, doi: 10.1016/j.heliyon.2019.e01802.

[10] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli and Rudy, "News Article Text Classification in the Indonesian Language", in *Procedia Computer Science*, Vol. 116, 2017, pp. 137-143, doi: 10.1016/j.procs.2017.10.039.

[11] A. Ghose and P. G. Ipeirotis, "Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics", *in IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 10, 2011, pp. 1498-1512, doi: 10.1109/TKDE.2010.188.

[12] J. C. Gomez, E. Boiy & M. -F. Moens, "Highly discriminative statistical features for email classification", *Knowledge Information Systems*, No.31, 2012, pp. 23–53, doi: 10.1007/s10115-011-0403-7.

[13] W. Hadi, Q. A. Al-Radaideh and S. Alhawari, "Integrating associative rule-based classification with naïve Bayes for text classification", *Applied Soft Computing*, No. 69, 2018, pp. 344–356, doi: 10.1016/j.asoc.2018.04.056.

[14] M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue and R. Guan, "Text classification based on deep belief network and softmax regression", *Neural Computing and Applications*, No. 29, 2018, pp. 61–70, doi: 10.1007/s00521-016-2401-x.

[15] S. Bahassine, A. Madani and M. Kissi, "An improved Chi-square feature selection for Arabic text classification using decision tree", in *11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 2016, pp. 1-5, doi: 10.1109/SITA.2016.7772289.

[16] P. Symeonidis, A. Nanopoulos and Y. Manolopoulos, "A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis," in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 2, 2010, pp. 179-192, doi: 10.1109/TKDE.2009.85.

[17] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization", *Information Processing & Management*, Vol. 42, No. 1, 2006, pp. 155-165, doi: 10.1016/j.ipm.2004.08.006.

[18] Z. Fan, W. Zuo, J. Yang, J.Tang, Z. Lai and D. Zhang, "Modified Principal Component Analysis: An Integration of Multiple Similarity Subspace Model", in *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 25, No. 8, 2014, pp. 1538-1552, doi: 10.1109/TNNLS.2013.2294492.

[19] C. Xiao, W. Wang, X. Lin, J. X. Yu and G. Wang, "Efficient similarity joins for near-duplicate detection", *ACM Transactions on Database Systems*, Vol.36, No.3, 2011, pp. 1-41, doi: 10.1145/2000824.2000825.

[20] D. Bollegala, Y. Matsuo and M. Ishizuka, "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words", in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 7, 2011, pp. 977-990, doi: 10.1109/TKDE.2010.172.

[21] Z. Zhang, Q. Zou, Y. Lin, L. Chen and S. Wang, "Improved Deep Hashing With Soft Pairwise Similarity for Multi-Label Image Retrieval", in *IEEE Transactions on Multimedia*, Vol. 22, No. 2, 2020, pp. 540-553, doi: 10.1109/TMM.2019.2929957.

[22] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics", in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 8, 2006, pp. 1138-1150, doi: 10.1109/TKDE.2006.130.

[23] Y. Gao, Y. Xu, H. Huang, Q. Liu, L. Wei and L. Liu, "Jointly Learning Topics in Sentence Embedding for Document Summarization", in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 32, No. 4, 2020, pp. 688-699, doi: 10.1109/TKDE.2019.2892430.

[24] B.M. Mathisen, A. Aamodt, K. Bach and H. Langseth, "Learning similarity measures from data", *Progress in Artificial Intelligence*, No. 9, 2020, pp. 129–143, doi: 10.1007/s13748-019-00201-2.

[25] V. Verma and R. K. Aggarwal, "A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: an empirical and theoretical perspective", *Social Network Analysis and Mining*, No. 10, 2020, 43, doi: 10.1007/s13278-020-00660-9.

[26] H. -Y. Lin, "Efficient classifiers for multi-class classification problems", *Decision Support Systems*, Vol. 53, No. 3, 2012, pp. 473-481, doi: 10.1016/j.dss.2012.02.014.

[27] Q. Li, Y. Song, J. Zhang and V. S. Sheng, "Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering", *Expert Systems with Applications*, Vol. 147, 2020, 113152, doi: 10.1016/j.eswa.2019.113152.

[28] H. Zhang, L. Cui, X. Zhang and Y. Luo, "Data-Driven Robust Approximate Optimal Tracking Control for Unknown General Nonlinear Systems Using Adaptive Dynamic Programming Method," in *IEEE Transactions on Neural Networks*, Vol. 22, No. 12, 2011, pp. 2226-2236, doi: 10.1109/TNN.2011.2168538.

[29] K. Xu, C. Wen, Q. Yuan, X. He and J. Tie, "A MapReduce based parallel SVM for email classification", *Journal of Networks*, Vol. 9, 2014, pp. 1640–1647.

[30] J. Pujara, H. Daumé and L. Getoor, ''Using classifier cascades for scalable e-mail classification'', in *Proceedings 8th Annual Collaboration, Electron. Messaging, Anti–Abuse Spam Conf. (CEAS)*, Perth, WA, USA, 2011, pp. 55–63, doi: 10.1145/2030376.2030383.

[31] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed and M. A. Al-Garadi, "Email Classification Research Trends: Review and Open Issues", *IEEE Access*, Vol. 5, 2017, pp. 9044–9064, doi: 10.1109/ACCESS.2017.2702187.

[32] M. D. del Castillo, A. Iglesias and J. I. Serrano, "Detecting Phishing E-mails by Heterogeneous Classification", *International Conference on Intelligent Data Engineering and Automated Learning – IDEAL 2007*, Vol. 4881, 2007, pp. 296-305, doi: 10.1007/978-3-540-77226-2_31.

[33] G. Shan, S. Xu, Li Yang, S. Jia and Y. Xiang, "Learn#: A Novel incremental learning method for text classification", *Expert Systems with Applications*, Vol. 147, 2020, 113198, doi: 10.1016/j.eswa.2020.113198.

[34] M. Oghbaie and M. M. Zanjireh, "Pairwise document similarity measure based on present term set", *Journal of Big Data*, No. 5, 2018, 52, doi: 10.1186/s40537-018-0163-2.

[35] A. A. Amer, and H. I. Abdalla, "A set theory based similarity measure for text clustering and classification", *Journal of Big Data*, No. 7, 2020, 74, doi: 10.1186/s40537-020-00344-3.

[36] H. Huan, J. Yan, Y. Xie, Y. Chen, P. Li and R. Zhu, "Feature-Enhanced Nonequilibrium Bidirectional Long Short-Term Memory Model for Chinese Text Classification", in *IEEE Access*, Vol. 8, 2020, pp. 199629-199637, doi: 10.1109/ACCESS.2020.3035669.

[37] A. Diaz-Manríquez, A. B. Ríos-Alvarado, J. H. Barrón-Zambrano, T. Y. Guerrero-Melendez and J. C. Elizondo-Leal, "An Automatic Document Classifier System Based on Genetic Algorithm and Taxonomy", in *IEEE Access*, Vol. 6, 2018, pp. 21552-21559, doi: 10.1109/ACCESS.2018.2815992.

[38] A. Vardasbi, H. Faili and M. Asadpour, "Eigenvalue based features for semantic sentence similarity", *Artificial Intelligence and Signal Processing Conference (AISP)*, 2017, pp. 184-189. doi: 10.1109/AISP.2017.8324078.

[39] Hadeel M, "An Efficient feature selection algorithm for the spam email classification", *Periodicals of Engineering and Natural Sciences*, Vol. 9, No. 3, 2021, pp.520-531, doi: 10.21533/pen.v9i3.2202.

[40] T. Poonkodi and D. S. Sukumaran, "E-Mail Spam Filtering Through Feature Selection Using Enriched Firefly Optimization Algorithm", *Turkish Journal of Computer and Mathematics Education*, Vol.12, No. 5, 2021, pp. 1248-1255, doi: 10.17762/turcomat.v12i5.1791.

[41] E. M. Bahgat, S. Rady, W. Gad and I. F. Moawad, "Efficient email classification approach based on semantic methods", *Ain Shams Engineering Journal*, Vol. 9, No. 4, 2018, pp. 3259-3269, doi: 10.1016/j.asej.2018.06.001.

[42] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers", *Lecture. Notes in Computer Science*, Vol. 2709, 2003, pp. 125–134, doi: 10.1007/3-540-44938-8_13.

[43] S. Wang and X. Yao,, "Multiclass imbalance problems: Analysis and potential solutions", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 42, No. 4, 2012, pp. 1119–1130, doi: 10.1109/TSMCB.2012.2187280.

[44] B. A. Kumara, M. M. Kodabagi, T. Choudhury and J., -S. Um, "Improved email classification through enhanced data preprocessing approach", *Spatial. Information. Research*, No. 29, 2021, 247–255, doi: 10.1007/s41324-020-00378-y.

[45] W. Zhang, T. Yoshida and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification", *Expert Systems with Applications*, Vol. 38, No. 3, 2011, pp. 2758-2765, doi: 10.1016/j.eswa.2010.08.066.

[46] H. Ahmed and A. K. Nandi, "Compressive Sampling and Feature Ranking Framework for Bearing Fault Classification with Vibration Signals", in *IEEE Access*, Vol. 6, 2018, pp. 44731-44746, doi: doi: 10.1109/ACCESS.2018.2865116.

[47] Z. Zhu, Y. -S. Ong and M. Dash, "Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework", in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics*), Vol. 37, No. 1, 2007, pp. 70-76, doi: 10.1109/TSMCB.2006.883267.

[48] J. Suriya Prakash, K. Annamalai Vignesh, C. Ashok and R. Adithyan, "Multi-class Support Vector Machines classifier for machine vision application", in *2012 International Conference on Machine Vision and Image Processing (MVIP)*, 2012, pp. 197-199, doi: 10.1109/MVIP.2012.6428794.

[49] U. Naqvi, A. Majid and S. A. Abbas, "UTSA: Urdu Text Sentiment Analysis Using Deep Learning Methods", in *IEEE Access*, Vol. 9, 2021, pp. 114085-114094, doi: 10.1109/ACCESS.2021.3104308.

[50] C. Dewi, S. -W. Huang and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods", *Journal of Big Data*, No.7, 2020, 52, doi: 10.1186/s40537-020-00327-4.

[51] E. O. Abiodun, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif & R. S. Alkhawaldeh, "A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities", *Neural Computing and Applications*, No. 33, 2021, pp. 15091–15118, doi: 10.1007/s00521-021-06406-8.

[52] S. S. Patil and S. P. Sonavane, "Improved classification of large imbalanced data sets using the rationalized technique: Updated Class Purity Maximization Over_Sampling Technique (UCPMOT)", *Journal of Big Data*, No. 4, 2017, 49, doi: 10.1186/s40537-017-0108-1.

[53] P. S. Parmar, P. K. Biju, M. Shankar and N. Kadiresan, "Multiclass Text Classification and Analytics for Improving Customer Support Response through different Classifiers", in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 538–542, doi: 10.1109/ICACCI.2018.8554881.

[54] G. -B. Huang, H. Zhou, X. Ding and R. Zhang, "Extreme learning machine for regression and multiclass classification", in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics*), Vol. 42, No. 2, 2012, pp. 513–529, doi: 10.1109/TSMCB.2011.2168604.

[55] G. Mujtaba, L. Shuib, R. G. Raj and R. Gunalan, "Detection of suspicious terrorist emails using text classification: A review", *Malaysian Journal of Computer Science*, Vol. 31, No. 4, pp. 271-299, 2018, doi: 10.22452/mjcs.vol31no4.3.