Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

A New Filter Feature Selection Method for Text Classification

RASIM CEKIK[1]

[1] Department of Computer Engineering, Faculty of Engineering, Şırnak University, rasimcekik@sirnak.edu.tr

Corresponding author: R. CEKIK (rasimcekik@sirnak.edu.tr).

ABSTRACT Massively amounts of text data have been created on the Internet due to the widespread use of platforms like social media. Text classification is one of the most frequently used techniques for extracting useful information from text data. One of the most fundamental problems in text classification is high dimensionality. In text classification, high dimensionality greatly reduces the success of classifiers while increasing their computational cost. The most effective way to overcome this problem is to select a subset of features comprising the most distinctive features across the entire feature space, with the help of a feature selector. This study presents a new filter feature selection approach called Multivariate Feature Selector (MFS) for text classification. The proposed approach calculates a score for each feature based on three knowledge structures: class-based, document-based, and document-class-based. These structures have been utilized to reveal hidden information at the class, document, and document-class levels. This enables a more precise and effective scoring calculation for each term. The proposed method (MFS) was tested on four different datasets, and micro-F1 and macro-F1 measures were used as performance evaluators to prove the method's success in feature selection. It has been observed that MFS outperforms the main feature selection methods in the literature. While different classification results were obtained depending on the selected feature size, MFS showed superior performance in all selected sub-feature spaces.

INDEX TERMS Feature Selection, Text Classification, Dimensionality Reduction, Text Mining

I. INTRODUCTION

Technological developments continue rapidly, and technology is used in almost every aspect of life. These developments have brought some difficulties as well as conveniences. Social interactions, shopping, comments, news, etc., on the Internet have led to the formation of a large amount of dirty data. The vast majority of this data generated and shared on the Internet is text data. News on CNN, articles on the Wall Street Journal website, tweets sent by Twitter users, product reviews published on Amazon, and many more are examples of text data on the Internet. However, in these information aggregation and dissemination centers, people only want to see the web pages they are interested in, not the irrelevant ones. To efficiently index web pages (or documents) containing predefined categories, text data needs to be classified according to topics or predefined categories. At this stage, text classification, a machine learning approach, is used for this purpose. Since the volume of text data is very large, some extra processing is required before classification to improve the classification prediction for such text data. One of these operations is the reduction of the text feature space [1]. This process significantly improves the performance of the classifiers used for classification and allows for a more robust classification process. The most common and practical method used for dimensionality reduction is selecting highly discriminative features. In machine learning [53], the dimension reduction and detection of only relevant features is called feature selection [2, 3]. Today, feature selection methods are used for dimension reduction in speech recognition [4], image processing [5, 49], spam filtering [6], sentiment classification [7], medical document classification [8], weather prediction [9], credit risk analysis [10], and cancer detection [11, 48, 50, 51].

Especially in high-dimensional datasets, detecting irrelevant or abundant features is essential for accurately predicting the problem. This is because irrelevant or abundant features significantly reduce prediction accuracy in classification [12, 13]. Therefore, improving the performance of classifiers is possible by preprocessing the data before classification. In this context, overcoming the curse of dimensionality and overfitting for classification problems is possible by identifying the sub-feature space that best represents the dataset [14].

Feature selection methods enhance the performance and success of classifiers while simultaneously reducing their execution time. Feature selection techniques are generally categorized under filter, wrapper, and embedded approaches [15, 16]. Wrapper approaches use a classifier to select the best subset of features. When selecting features, these methods examine the relationship between features and may take longer to run than other approaches. Filter approaches calculate value information for each feature and select a certain number of features with the highest values. They do not use any classifiers or learning algorithms for these operations. The relationship between attributes is not considered in these approaches, and the runtime is faster than in other approaches. Embedded approaches are created by integrating the feature selection process into the classifier, and such feature selection approaches are specific to learning algorithms. The runtime of these methods can be faster than wrapper approaches and slower than filter approaches.

Filter approaches perform the selection function using only statistical information, significantly reducing data processing costs. For this reason, filter approaches provide significant advantages, especially for large volumes of data. As a result, statistical information-based feature selection algorithms are well suited to high-volume and high-redundancy data.

There is a lot of research going on in the area of natural language modelling for text classification. Most of this work has focused on improving classification performance. However, many efforts to increase this performance encounter the problem known as the curse of dimensionality. This curse is a situation that negatively affects the performance of classifiers. In this context, the key to providing an effective solution is to identify and highlight the features that provide the most information among all the features. The primary motivation of this work is to highlight features that provide specific and meaningful information by assigning them a high value. In this way, it will be possible to improve classification performance and reduce the effects of the curse of dimensionality.

In the literature, feature selection strategies are employed to improve classification performance and generate faster results. Filter techniques, which are typically used for feature selection, can also be utilized for this. This study proposes a new feature selection method called Multivariate Feature Selector (MFS) to select highly efficient features in the text data. The proposed approach computes a score for each feature based on three knowledge structures: class-based, document-based, and document-class-based. Each structure provides hidden patterns in text documents. MFS uses these patterns to compute the optimal score using a filter approach. After determining the most discriminative characteristics, classification algorithms are utilized in these filter approaches to examine the methods' performance. SVM (Support Vector Machines), KNN (k-nearest neighbor), and NB (Naive Bayes) classifiers are utilized in this study for this aim. In addition, macro-F1 and micro-F1 evaluation criteria were used in the experimental studies to compare the classification performance of the proposed method with the methods in the literature.

The main contributions of our study can be summarized as follows:

- This study presents an effective feature selector for reducing the computational cost and improving the performance of classifiers in labeled text classification.
- A new FS method, termed the Multivariate Feature Selector (MFS), is proposed, which successfully selects informative features from various datasets.
- The performance of MFS is demonstrated against seven well-known feature selection methods.
- Experiments are conducted on four benchmark datasets using three different classifiers to ensure thorough analysis.

The remainder of this study is organized into four sections. Section II presents the material and the well-known FS methods in the literature. The functionality of the proposed new MFS metric is elucidated in Section III. Section IV discusses the results of the experiments conducted. Finally, Final Section outlines the conclusions of the study.

II. RELATED WORK

In high-dimensional text data, feature selection is critical to improve the performance of classifiers and reduce model complexity. Feature selection methods aim to improve classification accuracy and reduce computational cost by identifying the most informative features in text data. These methods are usually based on factors such as the frequency of words in the text, the distribution of terms across documents and the discrimination between classes.

There are numerous successful filter strategies in the literature, including information gain (IG) [17, 18], Gini index (GI) [19], chi-square (Chi2) [20, 21], normalized difference measure (NDM) [22], and max-min ratio (MMR). Relative discriminant criteria (RDC) [23], variable relative discriminative criterion (MRDC) [24], multi-objective relative discriminative criterion (MORDC) [25], discriminative power measure (DPM) [26], comprehensively measure feature selection (CMFS) [27], rough set-based proportional rough feature selector (PRFS) [28], and improved Gini index (GI) [29] are also popular filter feature selection methods.

In recent years, there has been an intense interest in filtering-based feature selection methods in the field of text classification. Due to the complexity and high dimensionality of text data, there is a need for efficient feature selection methods to improve classification performance. The number of newly proposed methods shows that the work in this field is developing rapidly and is open for further research. For example, some of these studies; Parlak and Uysal [30] proposed a novel filter feature selection method for text classification called the Extensive Feature Selector (EFS). The method is a filter-based approach based on class-based and corpus-based discrimination for each feature in the dataset. The filter-based strategy discriminates each feature in the dataset using class-based and corpus-based methods. Compared to the methods in the literature, they produced features with higher discrimination power and more successful results using this methodology. To compare the methodologies, the micro-F1 and macro-F1 criteria were utilized. A new feature selection technique for text classification is introduced by Mamdouh and Abd [54], focusing on frequent and correlated items. This approach takes into account both relevance and feature interactions, utilizing association as a metric to assess the relationship between the target and features. The ARDEN (Amount of ReDistribution to Establish Neutrality) method developed by Okkalioglu [55] offers a new feature selection approach to the text classification problem. ARDEN determines how discriminative a term is between classes by measuring its statistical distance from a neutral term that is equally distributed among all classes. By quantitatively expressing the degree to which a term deviates from neutrality, this method provides a more objective decision-making mechanism in the feature selection process. Uysal and Gunal [31] offer a new filter-based probabilistic feature selection method called discriminative feature selector (DFS) for text categorization. DFS picks discriminative features by deleting non-informative ones while considering specific term feature constraints. The experimental results reveal that DFS performs competitively in classification accuracy, dimensionality reduction rate, and processing time compared to other techniques in the literature.

Zhou et al. [32] proposed a method based on the term frequency deviation rate for text classification. Combining the suggested method with the CHI, IG, MMR, and TCM methodologies from the literature allows it to account for the effect of term frequency weights. On different datasets, the proposed technique boosted classification success by 7.9%. In [33], Parlak presented a new filter-based method called the bright probabilistic feature selector (BPFS) paper for assigning a fair score and selecting useful features. The BPFS technique seeks to pick unique features while selecting discriminative features by assigning higher scores than the common feature, considering sparse features in the relevant and other classes.

Traditional text classification methods usually work by considering document frequencies. However, this approach may ignore important differences between texts. For example, if a certain term is frequently used in one class and rarely used in other classes, it may be unique to that class. Such special terms can provide important clues to classify texts correctly. Kim and Zang [34] emphasize that considering the document frequencies and relative document frequencies within classes is crucial in text categorization. Therefore, they propose a new approach called Trigonometric Comparison Measure (TCM), which considers the rarity of terms across classes by considering relative document frequencies. The proposed filtering method adopts the idea that the absence of a term in a given class can provide valuable information for text categorization. However, this method uses trigonometric functions to solve the division by zero problem faced by NDM [22].

NDM and TCM may overestimate the importance of unusual and infrequent terms, making it difficult to find a balance between them. Jin et al. [35] developed the maximum difference maximization criteria (MDMC) to solve these challenges. MDMC effectively improves the scores of rare terms while decreasing those of sparse terms without additional parameters. As a result, MDMC can accurately identify distinguishing phrases. According to machine learning research, data from the same class often appears close to each other. Hence, considering a local compactness characteristic improves classification performance. Based on this knowledge, Zhu et al. [36] offer a Compactness Score (CSUFS), a fast, unsupervised feature selection approach, to choose the desired features. Extensive clustering task experiments are carried out to demonstrate the suggested algorithm's superiority, and successful results are obtained.

III. MATERIAL AND METHODS

This section provides information about the study's materials and methods.

A. CLASSIFIERS

Text classification determines which categories a document belongs to according to its content [37, 38]. The text classification problem is also known as text categorization. The main purpose of classification is to predict the target class for the specified data. Classifiers such as support vector machines (SVM) [39], naïve bayes (NB) [40], k-nearest neighbors (KNN) [41], and decision trees (DT) [42] are available to perform the classification task. The main purpose of the classifiers frequently used in the literature is to maximize classification accuracy. Each process used before classification aims for high classification accuracy. Because the proposed feature selection method is filter-based, it is not affected by the learning model. As a result, three distinct classifiers were used to investigate the effect of the chosen features on classification accuracy. Table 1 contains brief descriptions of the classifiers used in the experimental portion of the study.

TABLE I
CLASSIFIERS ARE USED TO EVALUATE THE PERFORMANCE OF THE PROPOSED FEATURE SELECTION METHOD.

Classifier	Description
Support Vector Machines (SVM)	The SVM is regarded as an effective classifier in the literature, and it is based on the margin maximization principle. It also has linear and nonlinear versions, depending on the kernel type.
k-Nearest Neighbors (kNN)	The kNN is a non-parametric method for storing all available cases and classifying new ones using similarity measures. The kNN has seen widespread use in pattern recognition and statistical estimation.
Naïve Bayes (NB)	The NB is based on Bayes' theorem and the independence assumptions between predictors.

B. EXISTING FEATURE SELECTION METHODS

Feature selection selects the most representative feature subset to classify data in the feature space. Generally, feature selection is handled with three different approaches in the literature. These methods are filter, wrapper, and embedded feature selection approaches. Filter approaches calculate a score value for each feature based on a statistical calculation. Each score value represents the power of individual discrimination on the classification for each feature. Filter approaches do not use any classifier or machine learning model for feature selection. Filter approaches calculate a score value for each feature based on a statistical calculation. Each score value represents the power of individual discrimination on the classification for each feature. Filter approaches do not use any classifier or machine learning model for feature selection. However, wrapper approaches try to select the most appropriate subset of features based on a particular metric using a classifier. On the other hand, embedded approaches attempt to select the appropriate subset using the advantageous aspects of both the filter and wrapper approaches. In this study, we aim to develop a new feature selection method for text classification based on the filter approach that has proven successful in text classification.

There are many methods presented as filter approaches in the literature. The most widely used and recently suggested approaches to these methods are presented in this section.

The representation of the document frequency measure is shown in Table 2 as a confusion matrix. Additionally, Table 3 provides definitions of some notations used in the text classification domain.

TABLE 2
CONTINGENCY TABLE OF THE RELATIONSHIP BETWEEN TERM AND CLASS

	t (*containing the term*)	$\bar{t}$ (*not containing the term*)
C_i(*belonging to class C*)	*TP (true positive)*	*FN (false negative)*
$\bar{C_i}$ (*not belonging to class C*)	*FP (false positive)*	*TN (true negative)*

TABLE 3
PRELIMINARIES

NOTATION	VALUE	DESCRIPTION
---	*TP*	*Number of documents belonging to the class C_i and presence of the term t.*
---	*FP*	*Number of documents belonging to the other classes ($\bar{C_i}$) and presence of the term t.*
---	*FN*	*Number of documents belonging to the class C_i and absence of the term t.*
---	*TN*	*Number of documents belonging to the other classes ($\bar{C_i}$) and absence of the term t.*
$P(t\|C_i)$	$tpr = \frac{TP}{TP+FN}$	*The probability of term t given presence of class C_i.*
$P(C_i\|t)$	$\frac{TP}{TP+FP}$	*The probability of class C_i given presence of term t*
$P(C_i\|\bar{t})$	$\frac{FN}{FN+TN}$	*Indicate the conditional probability of passing or not passing the given t term in C_i class.*
$P(t\|\overline{C_i})$	$fpr = \frac{FP}{FP+TN}$	*Denotes the conditional probability of term t given all the classes except C_i*
$P(C_i)$	---	*The probability of the C_i class*
$P(t), P(\bar{t})$	---	*The probability of passing or not passing the t term, respectively.*

1) INFORMATION GAIN (IG)

Information gain (IG) [17] is a widely used and practical feature selection approach in data and text mining. The approach is based on Shannon's theory of information and thermodynamics. This approach measures the classification knowledge of a term in any class. In other words, it can be defined as the inverse of entropy. Entropy is the terminology that expresses the disorder of a system. If the number of distinct values a term can take is high, the IG method's choosing that term as a highly distinctive term can result in system overfitting. This situation is a disadvantage for IG. The mathematical representation of IG is given in equation (1).

$$IG(t) = -\sum_{i=1}^{M} P(C_i)logP(C_i) + P(t)\sum_{i=1}^{M} P(C_i|t)logP(C_i|t) + P(\bar{t})\sum_{i=1}^{M} P(C_i|\bar{t})logP(C_i|\bar{t}) \quad (1)$$

In the equation (1), M is the number of classes.

2) GINI INDEX (GI)

The Gini Index (GI) [18] method is presented to complete the shortcomings of the information gain and gain ratio approaches. In this approach, the entropy value is not used. GI first calculates each term's class information and Gini coefficient. Then, it calculates the Gini gain value for each term, depending on the class information relationship of the Gini coefficient, and the terms are selected according to this value. The mathematical representation of IG is given in equation (2).

$$GI(t) = \sum_{i=1}^{M} P(t|C_i)^2 . P(C_i|t)^2 \quad (2)$$

In equation (2), M represents the total number of classes.

3) CHI-SQUARE (CHI2)

Another common and popular filter feature selection approach, Chi-Square (Chi2) [19], is an efficient feature selection tool based on statistical information. Chi2 determines whether the relationship between two variables is dependent or independent. The Chi2 test is a technique used to analyze two independent observations in statistics. Independent observations for text classification include the formation of terms and classes. Accordingly, the Chi2 information is calculated as follows:

$$CHI2(t, C) = \sum_{t\in\{0,1\}} \sum_{C\in\{\,0,1\}} \frac{(N_{t,C} - E_{t,C})^2}{E_{t,C}} \quad (3)$$

The N and E values represent the observed and expected frequency for each case of term t and class C, respectively. In conclusion, the mathematical background of the Chi2 method used in this study is given in equation (4):

$$CHI2(t) = \sum_{i=1}^{M} P(C_i) * CHI2(t, C) \quad (4)$$

4) DISTINGUISHING FEATURE SELECTOR (DFS)

One of the most effective methods for feature selection in text classification is the distinguishing feature selector (DFS) [20] approach. The mathematical background of the DFS method is given in equation (5).

$$DFS(t) = \sum_{i=1}^{M} \frac{P(C_i|t)}{P(\bar{t}|C_i) + P(t|\bar{C_i}) + 1} \quad (5)$$

In equation (5), M denotes the total number of classes.

5) NORMALIZED DIFFERENCE MEASURE (NDM)

A new method for text classification was proposed by Rehman et al. using the balanced accuracy measure, namely the Normalized Difference Measure (NDM) [21]. It is stated that the proposed NDM method does not work efficiently in a skewed dataset consisting of relatively sparse terms. The formulation of the NDM approach is given in equation (6).

$$NDM = \frac{|TP - FP|}{\min(TP, FP)} \quad (6)$$

In the equation (6), if $\min(TP, FP) = 0$, the denominator is matched to a small value such as 0.001.

6) MAX-MIN RATIO (MMR)

Max-Min Ratio (MMR) gives a high score for each attribute if a term is more frequent in one class but a low score if the term is the same in more than one class. It also completes NDM deficiencies in large and very rare data and assigns the highest value to the relevant term. The mathematical background of the MMR [22] method is given in equation (7).

$$MMR = \max(TP, FP) * \frac{|TP - \mathrm{FP}|}{\min(TP, FP)} \quad (7)$$

7) DEVIATION FROM POISSON DISTRIBUTION (PS)

The deviation from the Poisson distribution (PS) [43] method, derived from the Poisson distribution, is a widely used approach to selecting effective words in information retrieval. The Poisson distribution is a distribution that is widely used in engineering and statistics, including computer science. PS is a Poisson distribution-based text classification method integrated into feature selection problems. Finally, the mathematical foundation of the PS method is as follows:

$$\begin{gathered} PS(t, C) = \frac{(a - \hat{a})^2}{\hat{a}} + \frac{(b - \hat{b})^2}{\hat{b}} + \frac{(c - \hat{c})^2}{\hat{c}} + \frac{(d - \hat{d})^2}{\hat{d}} \\ \hat{a} = n(C)\{1 - \exp(-\mu)\} \\ \hat{b} = n(C)\exp(-\mu) \\ \hat{c} = n(\bar{C})\{1 - \exp(-\mu)\} \\ \hat{d} = n(\bar{C})\exp(-\mu) \end{gathered} \quad (8)$$

$$\mu = \frac{F}{N}$$

F and N represent, respectively, the total frequency of the t term in all documents and the number of documents in the allocated training data. $n(C)$ and $n(\overline{C})$ represent the number of documents that belong to and do not belong to the C class, respectively. The PS method is defined as follows:

$$PS(t) = \sum_{i=1}^{M} P(C_i) * PS(t, C) \qquad (9)$$

8) DISCRIMINATIVE POWER MEASURE (DPM)

DPM analyzes the dataset for both positive and negative discriminative features. The DPM technique seeks to improve classification performance by selecting features that demonstrate more differences between classes. DPM seeks to identify discriminative features with better information value in text classification. The proposed method for feature selection has a very low computational cost [26]. DPM formula can be described as follows:

$$DPM(t) = \sum_{i=1}^{M} \left| P(t|C_j) - P(t|\overline{C}_j) \right| \qquad (10)$$

9) COMPREHENSIVELY MEASURE FEATURE SELECTION (CMFS)

Comprehensively Measure Feature Selection (CMFS) assesses a feature's impact on classification performance by considering both its inter-class and intra-class relevance [27]. CMFS formula can be defined as follows:

$$CFMS(t) = \sum_{i=1}^{M} P(C_j) \cdot P(t|C_j) \cdot P(C_j|t) \qquad (11)$$

IV. PROPOSED METHOD: MULTIVARIATE FEATURE SELECTOR (MFS)

An ideal filter feature selector is expected to assign a low score to features with low discrimination and a high score to features with high discrimination. It should also assign an optimal score to features for which it is challenging to determine whether they exhibit high or low distinctiveness. The distinctiveness of a feature generally depends on the following conditions:

1. If a feature occurs in a single class with a certain frequency, it should receive a high score.
2. If a feature occurs in a single class but only a few times, its distinctiveness is low, and it should receive a low score.
3. If a feature occurs in all classes but with a certain frequency in each class, it has low distinctiveness and should receive a low score.
4. If a feature occurs frequently in some classes and mostly in one class, it should receive a relatively high score; otherwise, it should receive a low score.

In addition to these general cases, some special cases are taken into consideration for features about which it is difficult to obtain information regarding their distinctiveness:

1. If a feature occurs in all classes and is more frequent in some classes, the distribution of the feature between classes is examined. If the frequency of occurrence leans towards one class, it receives an average score. The distribution of the feature by document is important here.
2. If a feature occurs in some classes and predominantly in a single class, it receives an average score. Here, document-based distribution information of the feature is a factor for distinctiveness.

Generally, a feature's high score depends on its occurrence frequency within a class. Therefore, distinctiveness is based on class-wise information. If a feature occurs mostly in one class, it is considered feature distinctive. However, if it is evenly distributed across classes, it is not discriminative. The following structure provides ideal information in this context:

$$MFS_{class-based} = \frac{count(c_i,t)/count(c_i)}{count(c_i,\bar{t})/count(c_i) + count(\bar{c}_i,t)/count(\bar{c}_i) + 1} \qquad (12)$$

Each component in the structure provides separate information, which includes the following:

- $count(c_i,t)/count(c_i)$ structure indicates the intensity of a feature in a class. When it is high, it implies that this feature frequently occurs in the class. This situation allows the feature to receive a high score.
- A high value of the $count(\bar{c}_i,t)/count(\bar{c}_i)$ construct indicates low discrimination. A high value suggests that the feature is predominantly present in classes other than positive classes. Conversely, a low value indicates the opposite situation.
- A low value of the $count(c_i,\bar{t})/count(c_i)$ structure is a sign of high distinctiveness for the term. A low value indicates that the distribution between classes is not homogeneous, suggesting that discrimination may be high.

To fully reveal the distinctiveness of the feature, it is necessary to examine specific cases in addition to general ones. These exceptions involve the distribution of the feature on a document basis. This information can be accessed with the following structure:

$$MFS_{document-based} = \frac{log^{|count(c_i,t)-count(\bar{c}_i,t)|}_{count(document)}}{log^{10}_{count(c_i,t)+count(\bar{c}_i,t)} + 1} \; and, \qquad (13)$$

$$log_{count(document)}^{|count(c_i,t)-count(\bar{c}_i,t)|} = \begin{cases} 0.1 & if\ \ |count(c_i,t)-count(\bar{c}_i,t)| = 0\ or\ 1 \\ log_{count(document)}^{|count(c_i,t)-count(\bar{c}_i,t)|} & other \end{cases} \quad (14)$$

Each component in the structure stores crucial information, enabling MFS to perform precise calculations.

- $|count(c_i,\ t) - count(\bar{c}_i,t)|$ provides information about the distribution of the feature across documents. A high value indicates frequent occurrence in a class, while a low value indicates a balanced distribution across classes. A value close to zero is assigned if it is 0 or 1. In this study, the value of this component is set to 0.1 when it is 0 or 1.
- $count(c_i,t) + count(\bar{c}_i,t)$, the expression, provides information about the distribution of the feature based on the entire document.

In addition to this information, it is also important to determine the impact of the feature on the classes in the value space. By examining the average frequency of occurrence for each feature per document and the distribution of this average frequency across each class, the impact of a feature on the classes can be determined. The following structure provides the impact of the average frequency on a class basis.

$$MFS_{class-document-based} = \frac{count(t)/count(class)}{count(document)} \quad (15)$$

As a result, the distinctiveness of a feature depends on multiple variables. These variables consist of three knowledge structures according to their knowledge base: class-based, document-based, and class-document-based. In this study, we compute the effects of these structures and present an ideal feature selector called the Multivariate Feature Selector (MFS). The mathematical background of MFS is as follows:

$$MFS(t) = (\sum_{i=1}^{M} MFS_{class-based} * MFS_{document-based}) * MFS_{class-document-based} \quad (16)$$

M denotes the number of classes, t denotes the passing status of the feature, and $\bar{t}$ denotes the non-passing status. $count(c_i,t)$ denotes the number of times the feature is passed in class c_i and $count(\bar{c}_i,t)$ denotes the number of times the feature is passed in classes other than c_i. Similarly, $count(c_i,\bar{t})$ denotes the number of times the feature is not passed in class ci.

A simple practical example:

A simple collection of documents illustrating the working principle of MFS is given in Table 4. In addition, information about the terms in this collection and the score provided by MFS for each term are given in Table 5.

TABLE 4
SIMPLE EXAMPLE

DOCUMENTS NAME	CONTENTS	LABEL
d_1	fig	A
d_2	fig apple	A
d_3	fig apple grape	B
d_4	fig grape	B
d_5	fig kiwi grape	C
d_6	fig kiwi	C

The MFS score calculation for each term on the simple document collection is given in Table 5:

$$MFS(fig) = \left(\left(\frac{\frac{2}{6}}{\frac{0}{2}+\frac{4}{4}+1}\right) * \left(\frac{log_6^2}{log_6^{10}+1}\right) + \left(\frac{\frac{2}{6}}{\frac{0}{2}+\frac{4}{4}+1}\right) * \left(\frac{log_6^2}{log_6^{10}+1}\right) + \left(\frac{\frac{2}{6}}{\frac{0}{2}+\frac{4}{4}+1}\right) * \left(\frac{log_6^2}{log_6^{10}+1}\right)\right) * \frac{|2-\left(\frac{6}{3}\right)| + |2-\left(\frac{6}{3}\right)| + |2-\left(\frac{6}{3}\right)|}{6} = 0.0000$$

$$MFS(apple) = \left(\left(\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}+1}\right) * \left(\frac{log_6^0}{log_2^{10}+1}\right) + \left(\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}+1}\right) * \left(\frac{log_6^0}{log_2^{10}+1}\right) + \left(\frac{\frac{0}{2}}{\frac{0}{2}+\frac{2}{4}+1}\right) * \left(\frac{log_6^2}{log_2^{10}+1}\right)\right) * \frac{|1-\left(\frac{2}{3}\right)| + |1-\left(\frac{2}{3}\right)| + |0-\left(\frac{2}{3}\right)|}{6} = 0.0030$$

$$MFS(kiwi) = \left(\left(\frac{\frac{0}{2}}{\frac{2}{2} + \frac{2}{4} + 1} \right) * \left(\frac{log_6^2}{log_2^{10} + 1} \right) + \left(\frac{\frac{0}{2}}{\frac{2}{2} + \frac{2}{4} + 1} \right) * \left(\frac{log_6^2}{log_2^{10} + 1} \right) + \left(\frac{\frac{2}{2}}{\frac{0}{2} + \frac{0}{4} + 1} \right) * \left(\frac{log_6^2}{log_2^{10} + 1} \right) \right) * \frac{|0 - \left(\frac{2}{3}\right)| + |0 - \left(\frac{2}{3}\right)| + |2 - \left(\frac{2}{3}\right)|}{6} = 0.0398$$

$$MFS(grape) = \left(\left(\frac{\frac{0}{3}}{\frac{2}{2} + \frac{3}{4} + 1} \right) * \left(\frac{log_6^3}{log_3^{10} + 1} \right) + \left(\frac{\frac{2}{3}}{\frac{0}{2} + \frac{1}{4} + 1} \right) * \left(\frac{log_6^1}{log_3^{10} + 1} \right) + \left(\frac{\frac{1}{3}}{\frac{1}{2} + \frac{2}{4} + 1} \right) * \left(\frac{log_6^1}{log_3^{10} + 1} \right) \right) * \frac{|0 - \left(\frac{3}{3}\right)| + |2 - \left(\frac{3}{3}\right)| + |1 - \left(\frac{3}{3}\right)|}{6} = 0.0075$$

For this simple document collection, the proposed approach (MFS) computes the maximum score for the term kiwi. The lowest score is for the term fig. The others are grape and apple, respectively. Considering the document collection, the term fig appears equally in all documents. This term has no distinctiveness at all, and feature selection approaches are expected to give a low score to this term. MFS gave the term fig a score of "0.0000", meaning the term has no distinctiveness. Again, the same is true for the term apple. However, the term apple occurs equally often in two classes and not in one class. This information indicates that it has relatively little distinctiveness. This is why MFS gave this term a very low score of "0.0030". Another term, grape, occurs in two classes and not at all in one class. Moreover, it occurs more frequently in one of the classes than in the other. This situation makes it more distinctive than apple. Therefore, MFS gave this term a score of "0.0075", which is higher than apple. In the last term, kiwi occurs in only one class. Therefore, it received a higher score than the other terms. MFS gave this term a score of 0.0398.

TABLE 5

DATA ON TERM OCCURRENCES AND SCORES.

feature	document frequency	class occurrences	score
fig	6	3	0.0000
apple	2	2	0.0030
grape	3	2	0.0075
kiwi	2	1	0.0398

The working principle of MFS on a simple document collection is given in this section. As can be seen from this simple example, it is shown that MFS performs an ideal scoring. Experimental studies have also shown that MFS outperforms existing similar works.

V. EXPERIMENTAL RESULTS

In this section, the results of the experimental studies will be presented. First, a brief information about the datasets used in the article is given. In the following sections, the findings and conclusions of the experimental studies are presented.

It is pertinent to note that the raw text data underwent several preprocessing steps to enhance its suitability for subsequent analysis. These steps included tokenization, stop-word removal, lowercase conversion, and stemming. Furthermore, unigram features were extracted, and TF-IDF weighting was employed to assign importance scores to these features. The general flow of the stages applied to the raw text data within the scope of the study is given in Figure 1.

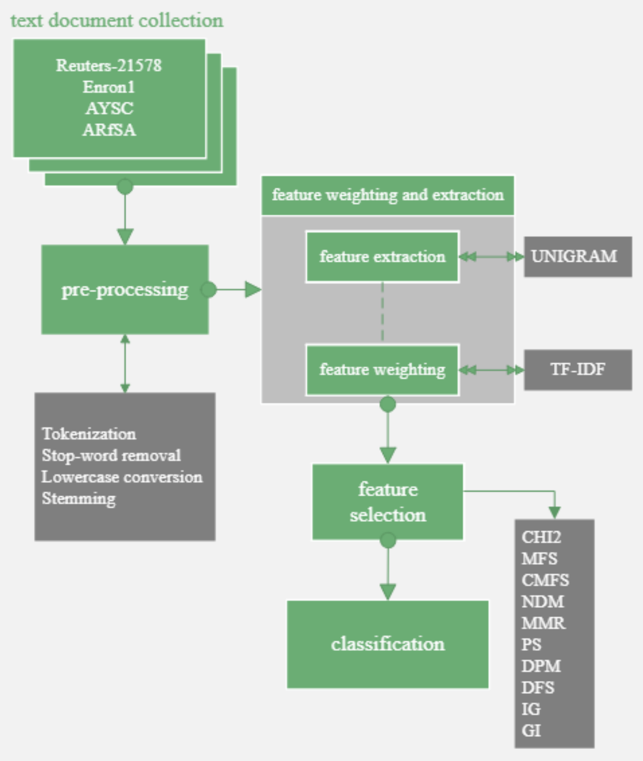


FIGURE 1. **Flowchart of the stages applied.**

1) DATASETS

In this study, four different datasets were used to evaluate the performance of feature selection methods. The first dataset used is the AYSC dataset, known as the YouTube Spam Collection, which combines five similar categories (Psy, KatyPerry, LMFAO, Eminem, Shakira) containing spam and non-spam reviews of various YouTube videos. The dataset comprises five categories with 1,956 actual messages [44]. This dataset is balanced and is used for binary classification as it consists of only two classes. The second dataset is the Amazon Reviews for Sentiment Analysis (ARfSA) [45]. The dataset used in this study is a subset of ARfSA, consisting of text documents containing the emotions of millions of Amazon customers. This dataset is balanced and is used for binary classification. The Enron Email Dataset is another dataset used in this study [46]. The dataset includes email data from approximately 150 users, mostly senior management of the Enron organization. The dataset was collected and prepared by the CALO Project (A Learning and Organizing Cognitive Assistant) and contains approximately 0.5 million messages in total. However, a subset of it, Enron1, was used in this study. This dataset is imbalanced as it contains a different number of documents in each class. Additionally, it is used for binary classification as it consists of only two classes. The recently used Reuters-21578 dataset is a collection of documents containing news articles. The original dataset has 10,369 documents and a vocabulary of 29,930 words. This dataset is a compilation of documents that appeared on the Reuters news portal in 1987. Documents were aggregated and indexed by category [47]. This dataset is termed unbalanced because it contains a different number of documents in each class and is multi-class. Additionally, information about the datasets is provided in Table 6.

TABLE 6
THE DATASETS USED IN EXPERIMENTAL STUDIES

	Label	Training samples	Test samples
Reuters-21578	Earn	2877	1087
	Acq	1650	719
	Money-fx	538	179
	Grain	433	149
	Crude	389	189
	Trade	369	117
	Interest	347	131
	Ship	197	89
	Wheat	212	71
	Corn	181	56
AYSC	Spam	703	302
	non-spam	665	286
ARfSA	Spam	10639	6183
	non-spam	10174	6374
Enron1	Spam	999	500
	non-spam	2449	1224

2) PERFORMANCE MEASURES

The performance metric results evaluate the feature subset classification performances of the filter approaches used in the literature. Micro-F1, macro-F1, and accuracy is frequently preferred in the literature as performance metrics [29, 37]. In this study, micro-F1 and macro-F1 values were used as evaluation metrics.

The precision metric is a metric that shows how many of the positively predicted values are actually positive. This metric value is especially important when the cost of false positive estimates is high. This metric value is obtained by dividing the number of true positive samples (TP) by the sum of the number of true positive samples (TP) and the number of false positive samples (FP).

$$Precision = (TP)/(TP + FP) \tag{17}$$

The Recall metric is a value that shows how many of the situations that should be predicted positively are predicted positively. This value is significant when the cost of estimating false negatives is high. This metric value is obtained by the ratio of the number of true positive samples (TP) to the sum of the number of true positive samples (TP) and the number of false negative samples (FN).

$$Recall = (TP)/(TP + FN) \tag{18}$$

Micro-F1 metric, used to determine classification performance, is calculated using the precision and the recall metric. A harmonic mean of the precision and recall metrics is used to show the performance of the classifier algorithm in a more balanced way.

$$Micro - F1 = 2 \times (Precision \times Recall)/(Precision + Recall) \tag{19}$$

One of the best-known success measures in the literature is the macro-F1 measure. In macro-averaging, the F-measure is calculated for each class in the dataset and then averaged across all classes. Each class is given equal, weight regardless of class frequency. The macro-F1 metric is calculated as follows:

$$p_i = \frac{1}{C} * \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FP_i} \tag{20}$$

$$r_i = \frac{1}{C} * \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FN_i} \tag{21}$$

$$F_i = 2 * \frac{p_i * r_i}{p_i + r_i} \tag{22}$$

$$Macro - F1 = \frac{\sum_{i=1}^{C} F_i}{C}, \tag{23}$$

3) SIMILARITY ANALYSIS

A feature selector needs to be defined and presented differently from other selectors available to see the specific profile of its selected features. It is important to identify and present the specific profile of features that a feature selector selects differently from other selectors. Different feature selection methods employ distinct strategies, leading to varying feature profiles. In this section, it delves into the profiling of the MFS method, examining its distinctiveness and ability to establish a unique profile. To illustrate this, first, Table 7 presents a comparison of the top 10 features selected by MFS and other methods across four datasets: (a) AYSC, (b) ARfSA, (c) Reuters_21578, and (d) Enron1. As observed, DPM consistently selects a higher number of distinct features compared to other methods across all datasets. This observation highlights DPM's unique selection strategy and its consequent formation of a distinctive profile. Beyond the comparisons in Table 7, a deeper analysis of feature selection method profiles can be conducted by examining the shared feature selections and their order. For instance, in the ARfSA dataset, the CHI2, DFS, and IG methods share the same first three selected features. Such insights enable the construction of method-specific profiles. A key takeaway from Table 7 is that MFS predominantly selects features that are also selected by other methods. This implies that MFS possesses an effective selection strategy that captures the important features identified by other methods, leading to a comprehensive profile.

MFS stands out by employing a distinct selection strategy that generates a unique and informative profile. The reduced number of selected features contributes to a clearer and more interpretable profile. This analysis serves as a starting point for understanding MFS profiling. A more comprehensive analysis would involve a detailed examination of each method and a thorough evaluation of the selected features. Therefore, Figures 2-5 were obtained in the next step for analysis.

TABLE 7
TOP-10 FEATURES IN (A) AYSC, (B) ARFSA, (C) REUTERS_21578, AND (D) ENRON1 DATASET

(a)										
CHI2	check	youtub	video	subscrib	song	channel	gui	hey	comment	love
DFS	check	youtub	subscrib	video	channel	song	monei	gui	hey	**call**
IG	youtub	check	video	song	subscrib	Gui	hey	**work**	comment	**start**
GI	check	youtub	video	subscrib	song	channel	love	monei	gui	make
PS	check	youtub	video	br	subscrib	song	channel	gui	hey	love
MMR	googl	type	open	smoke	guruofmovi	uncl	like	check	comment	video
NDM	googl	type	open	smoke	guruofmovi	uncl	check	video	like	youtub
DPM	comment	googl	type	like	open	smoke	guruofmovi	uncl	**stop**	**shakira**
CMFS	check	youtub	video	subscrib	song	channel	love	monei	gui	make
MFS	check	youtub	video	subscrib	channel	Br	song	**quot**	monei	gui
(b)										
CHI2	great	wast	disappoint	love	monei	Bad	excel	worst	return	poor
DFS	great	wast	disappoint	monei	love	Bad	excel	worst	return	poor
IG	great	wast	disappoint	love	monei	Bad	worst	excel	poor	return
GI	great	book	good	love	time	Don	read	work	monei	disappoint
PS	book	great	movi	wast	disappoint	love	monei	bad	album	excel
MMR	fals	huh	trashi	fad	millionair	garbag	hotboi	cash	saver	repetit
NDM	fals	huh	trashi	fad	millionair	garbag	hotboi	cash	saver	repetit
DPM	monei	**horribl**	**product**	**thing**	**expect**	**give**	**amazon**	**call**	**minut**	garbag
CMFS	book	great	good	time	read	love	work	don	**make**	**buy**
MFS	book	movi	great	wast	disappoint	read	monei	don	love	good
(c)										
CHI2	cts	net	shr	qtr	rev	loss	**acquir**	profit	**note**	**dividend**
DFS	cts	wheat	net	oil	shr	tonn	**corn**	barrel	qtr	agricultur
IG	cts	net	wheat	bank	shr	qtr	tonn	**export**	trade	agricultur
GI	cts	net	shr	wheat	oil	barrel	qtr	march	rev	**crude**
PS	mln	dlr	cts	loss	net	bank	**pct**	**billion**	trade	share
MMR	sorghum	wheat	grain	oat	bread	veget	cwt	bu	jul	linoil
NDM	sorghum	wheat	oat	bread	veget	cwt	grain	bu	jul	linoil
DPM	**feb**	**februari**	sorghum	grain	wheat	dlr	**product**	**loan**	**futur**	**board**
CMFS	cts	net	shr	march	qtr	mln	rev	**year**	loss	dlr
MFS	cts	net	march	shr	mln	loss	dlr	share	qtr	profit
(d)										
CHI2	http	cc	enron	gas	ect	pm	meter	forward	hpl	**www**
DFS	enron	cc	hpl	gas	ect	daren	hou	pm	forward	meter
IG	cc	gas	ect	pm	meter	http	corp	**volum**	**attach**	forward
GI	subject	enron	cc	hpl	gas	forward	ect	daren	hou	pm
PS	ect	hou	enron	meter	deal	subject	gas	pm	cc	corp

MMR	medic	sleep	fda	physician	ill	restor	bone	emot	disappear	forti
NDM	sleep	fda	physician	ill	restor	bone	emot	disappear	forti	**thick**
DPM	medic	subject	**click**	**onlin**	**ship**	**secur**	**prescript**	**order**	**low**	**doctor**
CMFS	subject	enron	cc	hpl	gas	forward	ect	daren	hou	pm
MFS	ect	hou	enron	subject	deal	meter	gas	hpl	cc	pm

Figures 2–5 visualize the similarities of MFS with each selection method. These figures show the common features that MFS specifically selects with each method and the features that other methods specifically select. That is, the figures show the features that MFS selects in common with each method, the features that MFS selects exclusively, and the features that the other method selects exclusively. In light of these figures, the profile of MFS compared to other methods can be analyzed.

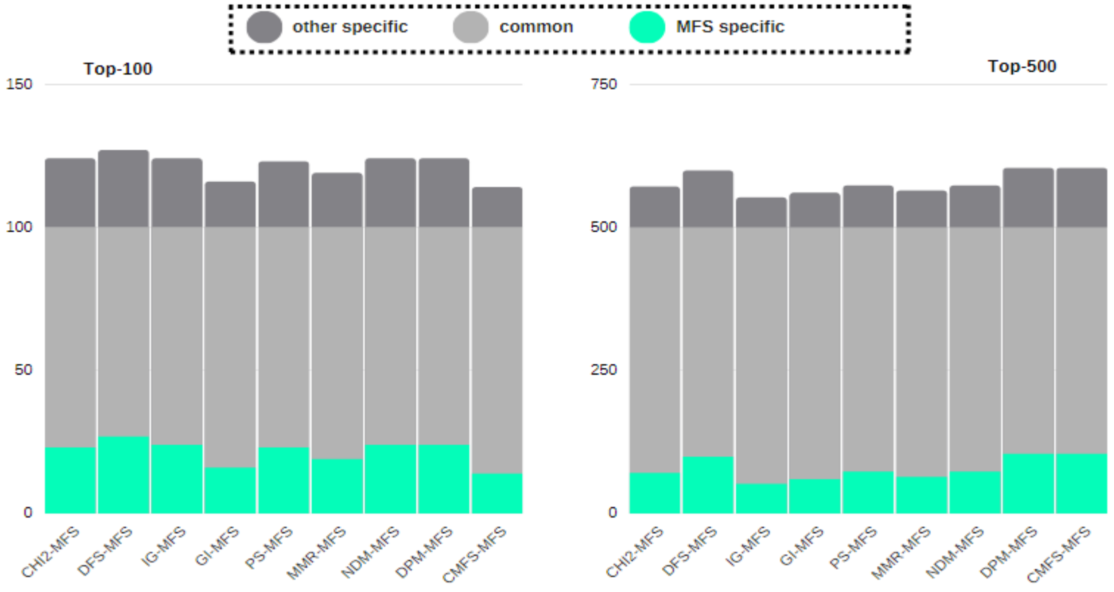


FIGURE 2. **Similarity comparison for the AYSC dataset.**

Figure 2 presents a comparative analysis of the MFS algorithm with other methods on the AYSC dataset. The figure shows that MFS has selected the most similar features with GI and CMFS in the Top-100. In particular, MFS has selected 77% of the features in common with CMFS in the Top-100 dimension. This high rate indicates that MFS shares a similar selection strategy with CMFS for this dataset. In the Top-500-dimension, MFS selected more common features than IG and GI. In particular, it has selected features that are highly similar to IG. This shows that MFS also converges with IG's strategy in feature selection. MFS also has some specially selected features from CMFS and IG in both dimensions. This situation has arisen due to the unique selection criteria of MFS. Another striking point in the figure is that the selection approaches select similar features at a high rate. This shows that the different methods used extract similar information from the dataset. In addition, with the increase in the feature selection dimension, the number of common features also increases. This can be explained by the fact that the distinction between features becomes clearer with the increase in the size of the dataset and different methods can capture this clarity.

Figure 3 demonstrates the outcomes derived from the ARfSA dataset. As depicted in the figure, the MFS method selected features closer to the GI method than the other methods in both feature dimensions in the ARfSA dataset. In addition to GI, MFS also showed close behaviour with PS and CMFS for this dataset. It also showed the same level of similarity with other methods in almost all cases.

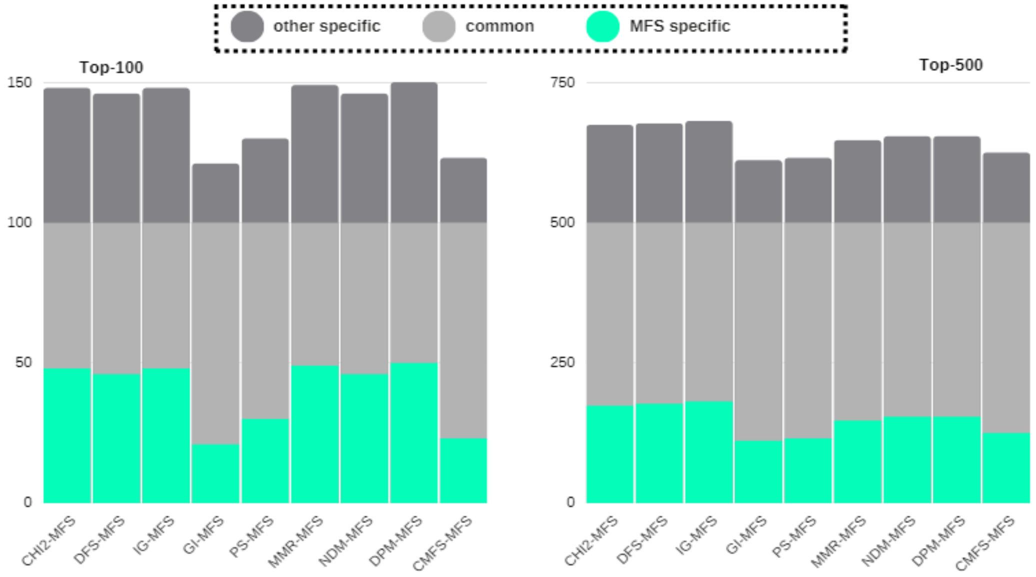


FIGURE 3. **Similarity comparison for the ARfSA dataset.**

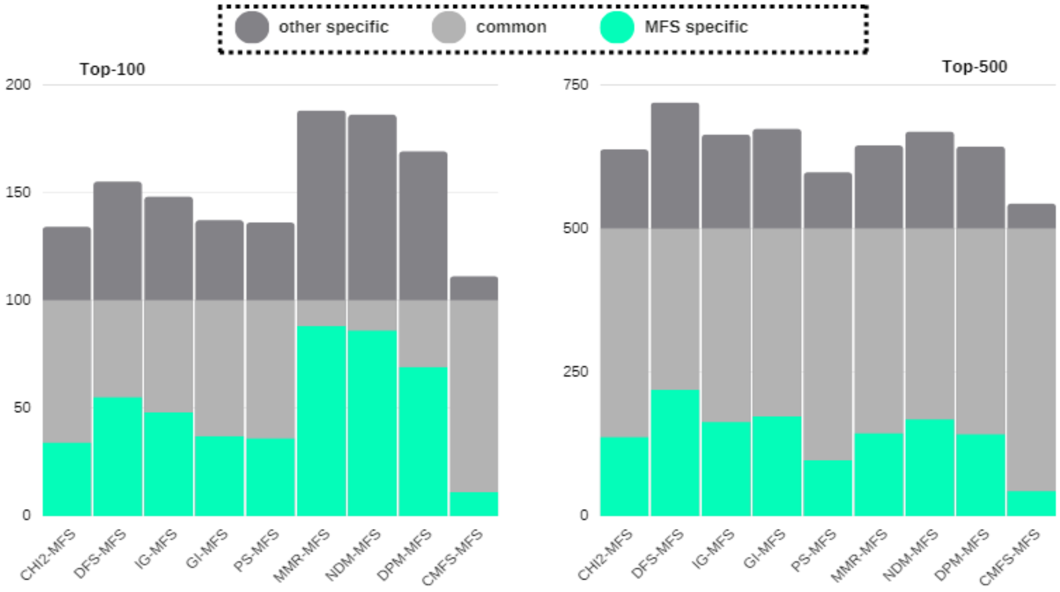


FIGURE 4. **Similarity comparison for the Reuters_21578 dataset.**

Figure 4 is generated for the Reuters_21578 dataset. In the figure, MFS selected a high percentage of features with CMFS. It showed similar behavior on this dataset. It is clearly seen that MMR, NDM and DPM in Top-100 work very differently with MFS. In the Top-500, it showed the least similarity with DFS.

Figure 5 pertains to the Enron1 dataset. MFS has selected the most common features with CMFS in both dimensions. This clearly shows that MFS is the most similar to CMFS in the Enron1 dataset. In addition to CMFS, MFS also selected a high average number of common features with IG and CFS. This indicates that MFS shares a significant similarity with these two methods in the Enron1 dataset. It is observed that MFS selects highly different features from IG and CFS in the Top-100 dimension. This may be due to the structure of the dataset and the working strategy of the feature selection approaches. It is natural that different features are important in different datasets and different methods capture these differences. This is also an important information to see the selection success of feature selectors. As the feature size increased, the common selection rate increased. However, MFS again performed closer selection with GI and CMFS.

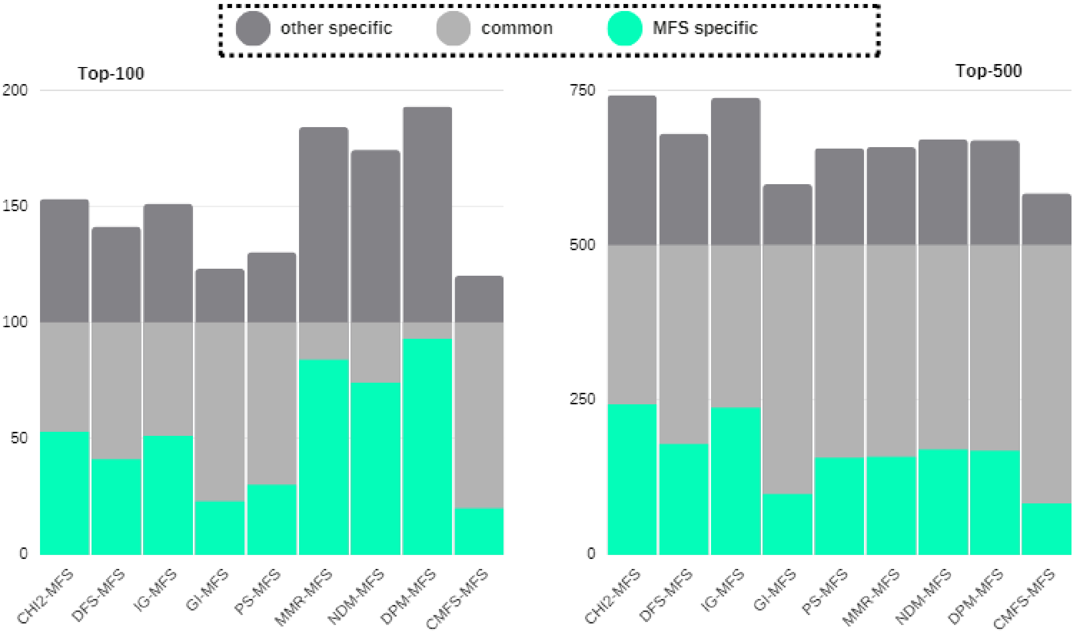
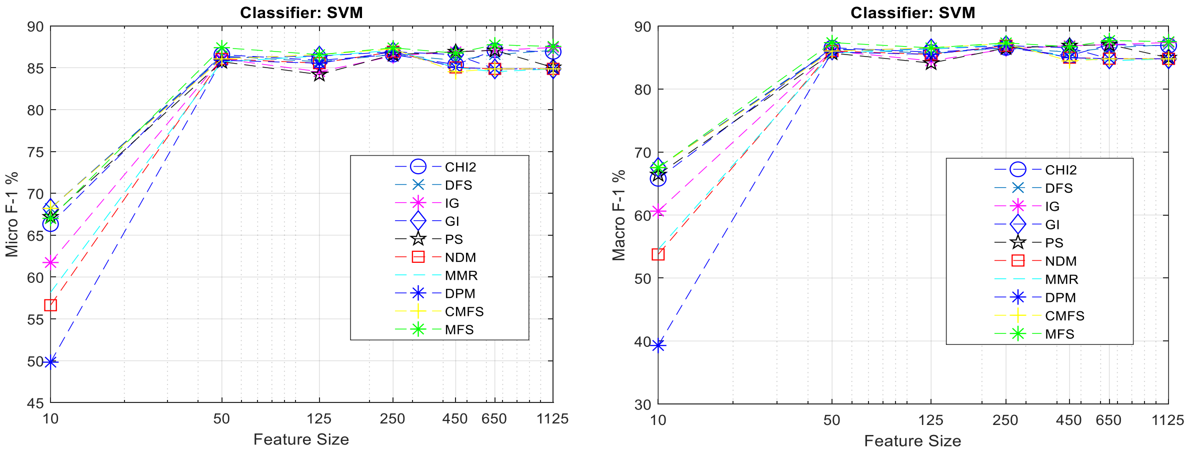


FIGURE 5. Similarity comparison for the Enron1 dataset.

4) ACCURACY ANALYSIS

In this part of the experimental studies, SVM, KNN, and NB classifiers are used to compare MFS with existing feature selection methods according to macro-f1 and micro-f1 values. The experimental investigation was carried out with seven different feature sizes: 10, 50, 125, 250, 450, 650, and 1125. In addition to traditional methods, recently proposed approaches were also included in the comparison. The performance of 10 different feature selection methods was evaluated in total. This made it possible to comprehensively evaluate the performance of MFS on different data sets and to analyse the effect of variations in feature size. SVM, KNN, and NB classifiers were run with varying numbers of features selected by each selection method. The resulting micro-F1 and macro-F1 scores for each dataset are presented in Figures 6-9, and the name of the selection method with the highest scores is added to the table below the figure. The figures show (a), (b) and (c) for SVM, KNN and NB, respectively.

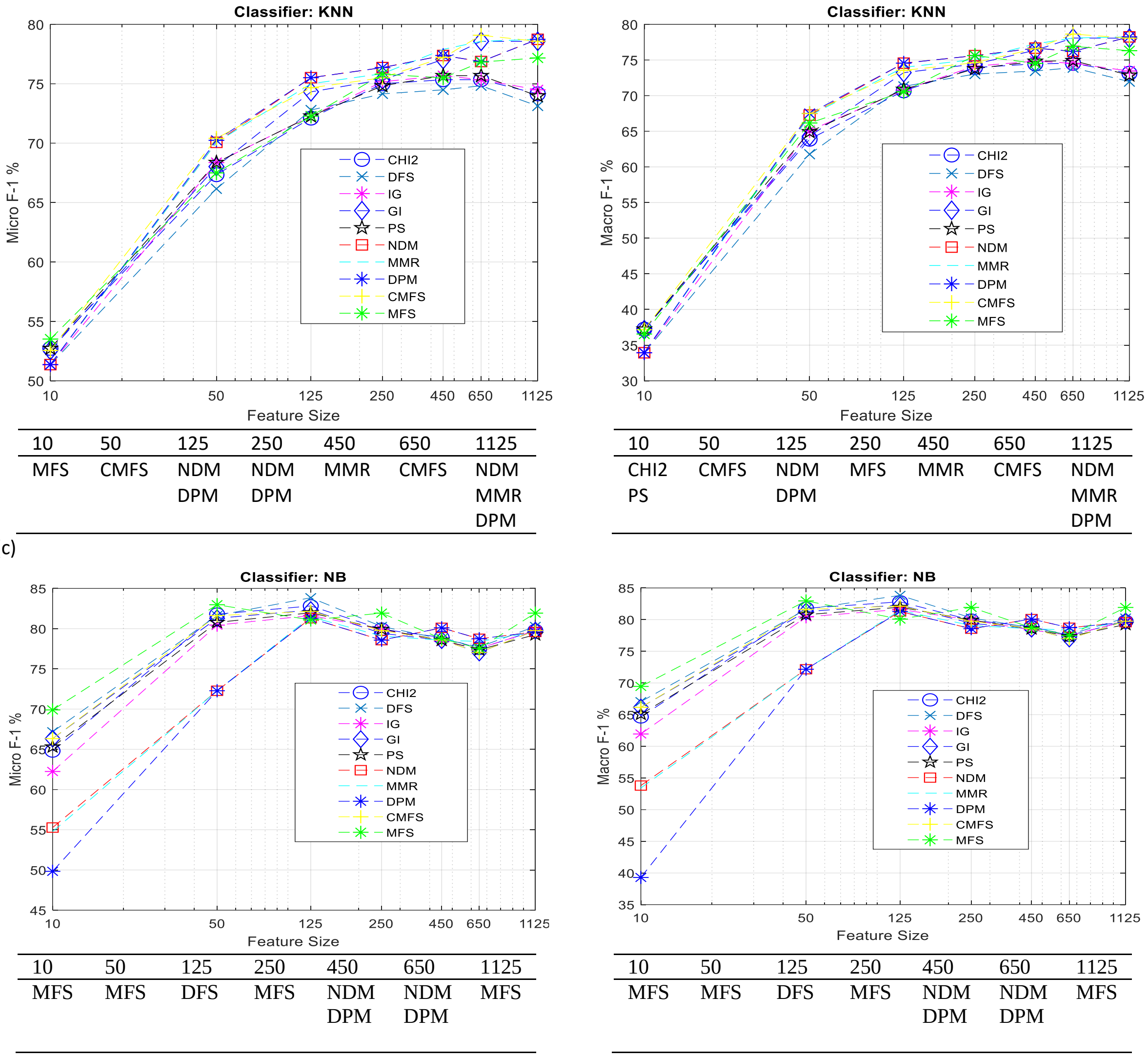
(a)



10	50	125	250	450	650	1125
CMFS	MFS	MFS CMFS	MFS	IG PS	MFS	MFS

10	50	125	250	450	650	1125
DFS GI CMFS	MFS	CMFS	MFS	IG PS	MFS	MFS

(b)



10	50	125	250	450	650	1125
MFS	CMFS	NDM DPM	NDM DPM	MMR	CMFS	NDM MMR DPM

10	50	125	250	450	650	1125
CHI2 PS	CMFS	NDM DPM	MFS	MMR	CMFS	NDM MMR DPM

(c)

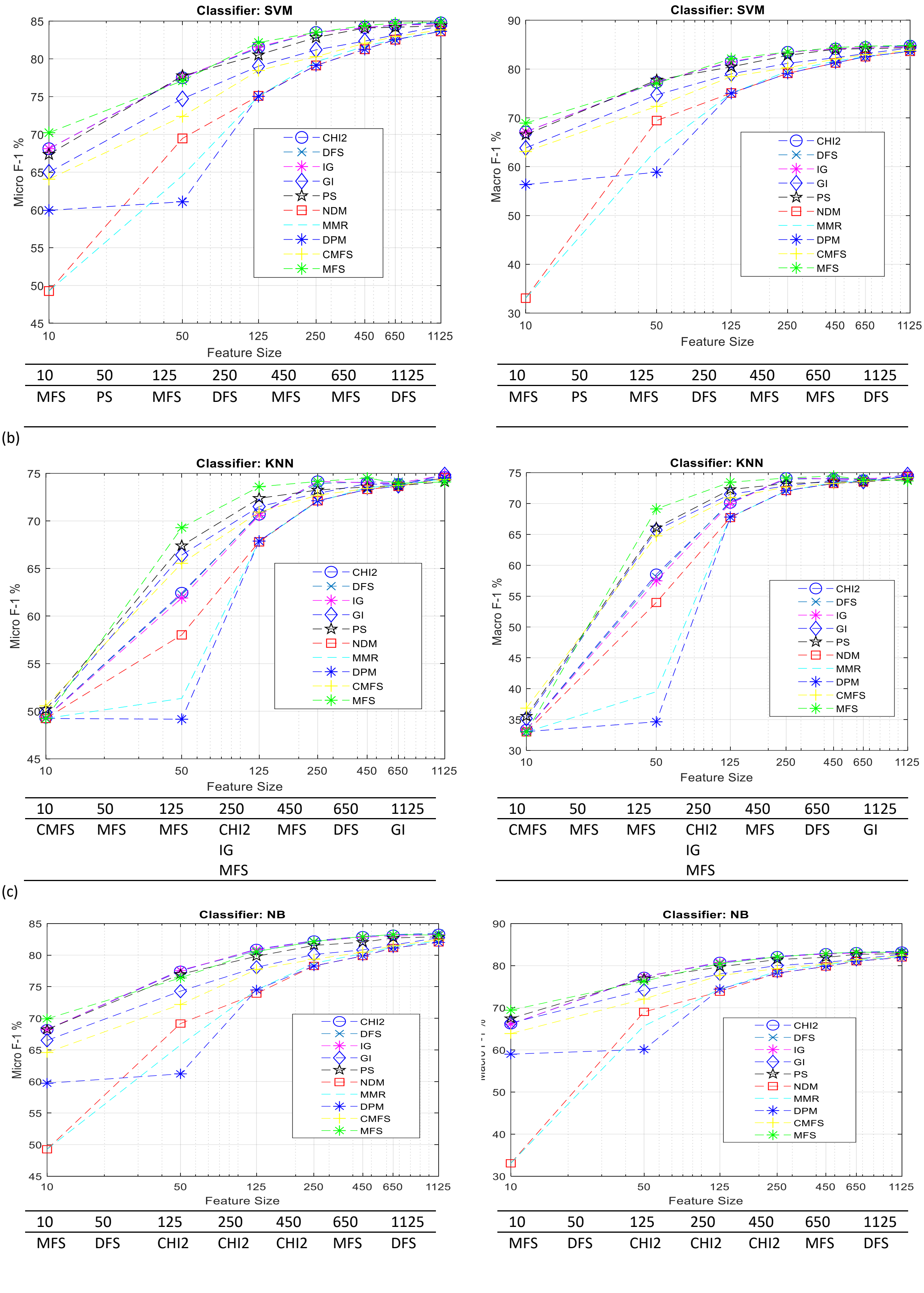
10	50	125	250	450	650	1125
MFS	MFS	DFS	MFS	NDM DPM	NDM DPM	MFS

10	50	125	250	450	650	1125
MFS	MFS	DFS	MFS	NDM DPM	NDM DPM	MFS

FIGURE 6. Success measures (%) for the AYSC dataset using (a) SVM, (b) KNN, and (c) NB

Figure 6 shows the macro-f1 and micro-f1 results obtained by the classifiers for the feature sizes offered by the selected methods on the AYSC dataset. In the figure, when SVM classifier is used, according to macro-F1 and micro-F1 criteria, the highest values are reached for MFS in 50, 250, 650 and 1125 dimensions, while MFS and CMFS are reached for micro-f in 125 dimensions. In the figure, when the SVM classifier is employed, the highest values for MFS are observed in dimensions 50, 250, 650, and 1125, based on both macro-F1 and micro-F1 criteria, while MFS and CMFS achieve the highest scores for micro-F in dimension 125. It can be inferred that MFS exhibits significant success when utilized with the SVM classifier for this dataset. In the case of the KNN classifier applied to the dataset, the highest values are attained with MFS at dimension 10 for micro-F1 and dimension 250 for macro-F1. Moreover, MFS generally secures the second position across other feature sizes. On employing the NB classifier for this dataset, the highest scores are recorded in dimensions 10, 50, 250, and 1125 for both micro-F1 and macro-F1. This underscores the effective performance of MFS and the NB classifier on the dataset.

(a)



10	50	125	250	450	650	1125
MFS	PS	MFS	DFS	MFS	MFS	DFS

10	50	125	250	450	650	1125
MFS	PS	MFS	DFS	MFS	MFS	DFS

(b)

10	50	125	250	450	650	1125
CMFS	MFS	MFS	CHI2 IG MFS	MFS	DFS	GI

10	50	125	250	450	650	1125
CMFS	MFS	MFS	CHI2 IG MFS	MFS	DFS	GI

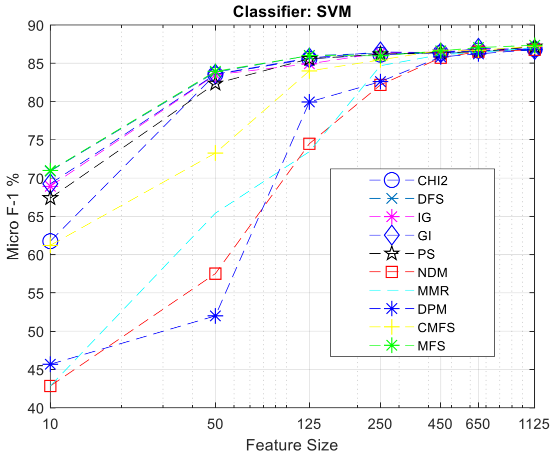
(c)

10	50	125	250	450	650	1125
MFS	DFS	CHI2	CHI2	CHI2	MFS	DFS

10	50	125	250	450	650	1125
MFS	DFS	CHI2	CHI2	CHI2	MFS	DFS

FIGURE 7. Success measures (%) for the ARfSA dataset using (a) SVM, (b) KNN, and (c) NB

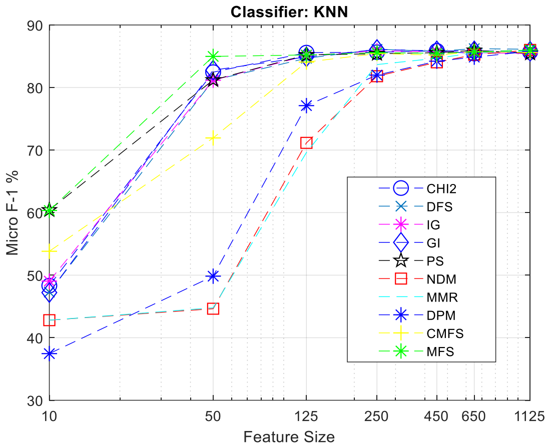
(a)



10	50	125	250	450	650	1125
MFS	MFS	DFS	GI	CMFS	MFS	MFS

10	50	125	250	450	650	1125
IG	DFS	DFS	MFS	DFS	MFS	MFS

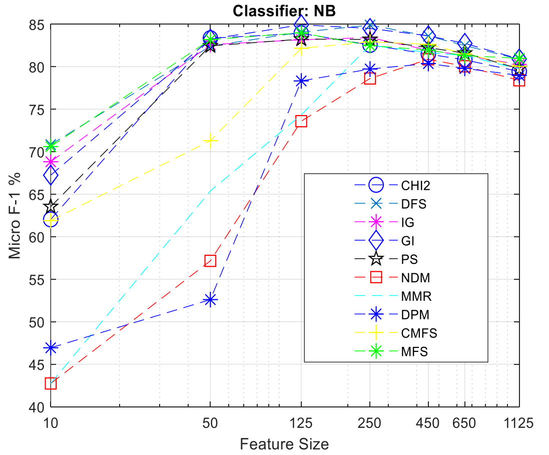
(b)



10	50	125	250	450	650	1125
PS	MFS	CHI2	DFS	CHI2 GI	DFS	DFS

10	50	125	250	450	650	1125
MFS	GI	MFS	GI	DFS	DFS	MFS

(c)



10	50	125	250	450	650	1125
DFS	CHI2	GI	DFS	GI	GI	MFS

10	50	125	250	450	650	1125
DFS	DFS	DFS	DFS	DFS	GI	MFS

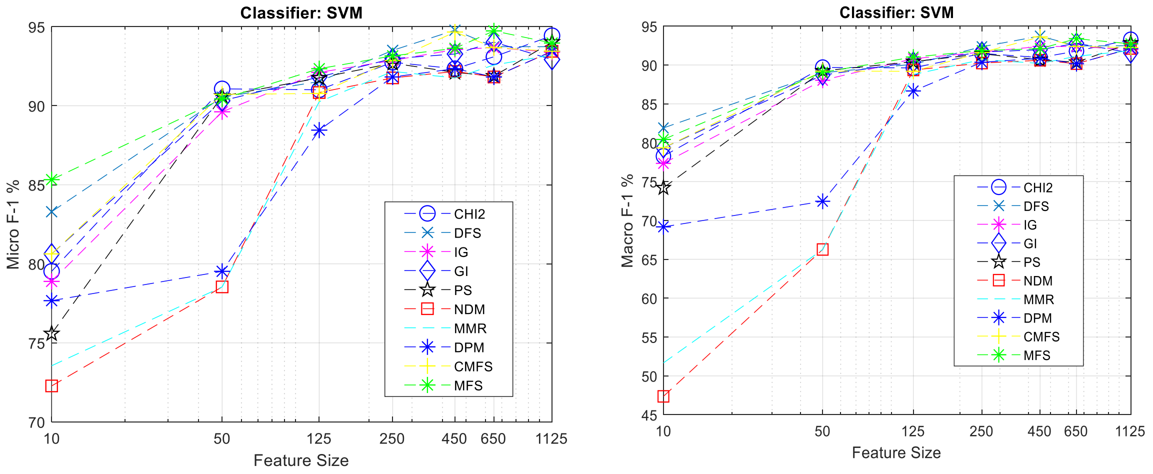
FIGURE 8. **Success measures (%) for the Reuters_21578 dataset using (a) SVM, (b) KNN, and (c) NB**

Figure 7 presents the performance comparison of the feature selectors used with different feature sizes and classifiers on the ARfSA dataset. When Figure 7 is analyzed, it is observed that MFS gives better results when the SVM classifier is used according to both criteria in dimensions 10, 50, 450, and 650. This observation underscores the efficacy of MFS in conjunction with the SVM classifier. Across various dimensions, MFS consistently demonstrates superior performance compared to alternative methodologies. Moreover, for the KNN classifier, the best results in terms of micro-F1 and macro-F1 have been achieved at dimensions 50, 125, 250, and 450. In these dimensions, MFS exhibits higher performance compared to other methods. The highest results are obtained in 10 and 650 dimensions when the BN classifier is used. In these dimensions, MFS performs better than the other methods. In other dimensions, MFS shows an average performance in general. In this dataset, MFS shows more consistent and high performance in SVM and KNN classifiers compared to other methods. In the NB classifier, MFS performs better than the other methods in some dimensions, but in general, it shows an average performance.

Figure 8 presents a performance comparison of feature selectors utilized across various dimensions and classifiers on the Reuters_21578 dataset. In Figure 7, for micro-F1 criterion, MFS achieves the highest scores in dimensions 10, 50, 650, and 1125. This indicates that MFS is highly effective when used with the SVM classifier. For macro-F1 criterion, it reaches 250, 650, and 1125 dimensions. This demonstrates that MFS also performs well for the macro-F1 criterion across all dimensions. Similarly, for the KNN classifier, MFS gave the best values in 10, 125, and 1125 dimensions according to the macro-F1 value. This suggests that MFS is quite effective for the macro-F1 criterion when used with the KNN classifier. According to both micro-F1 and macro-F1 values for BN, MFS achieved the highest score at dimension 1125.

Figure 9 illustrates the performance comparison of feature selectors employed in different dimensions and classifiers on the Enron1 dataset. When the SVM classifier was used, the MFS method reached dimensions 10, 125, and 650 for the micro-F1 value and 125 and 650 for the macro-F1 value. KNN reached 125, 250, and 650 in micro value and 125, 250, 650, and 1125 in macro-F1 value. Likewise, NB achieved the best result in the micro value in dimensions 10, 125, 450, and 650 and in the macro-F1 value in dimensions 50, 125, 450, and 650.

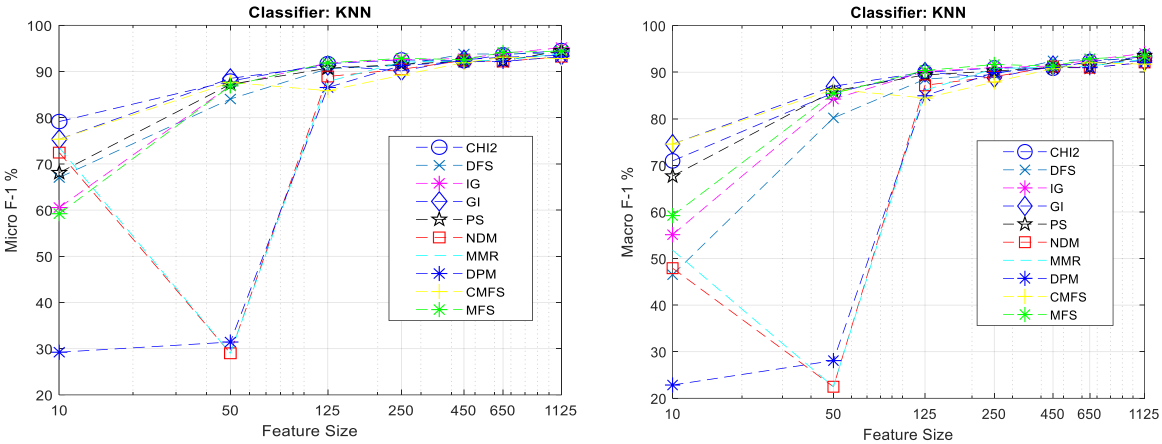
(a)



10	50	125	250	450	650	1125
MFS	CHI2	MFS	DFS	DFS	MFS	CHI2

10	50	125	250	450	650	1125
DFS	CHI2	MFS	DFS	DFS	MFS	CHI2

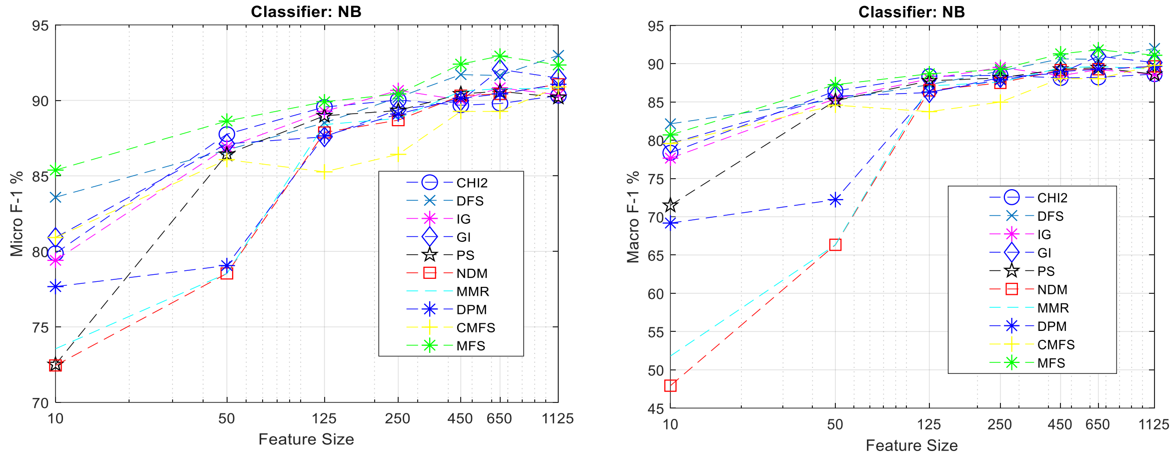
(b)



10	50	125	250	450	650	1125
CHI2	GI	MFS	MFS	DFS	MFS	IG

10	50	125	250	450	650	1125
GI CMFS	GI	MFS	MFS	DFS	MFS	MFS

(c)



10	50	125	250	450	650	1125
MFS	MFS	MFS	IG	MFS	MFS	DFS

10	50	125	250	450	650	1125
DFS	MFS	MFS	IG	MFS	MFS	DFS

FIGURE 9. Success measures (%) for the Enron1 dataset using (a) SVM, (b) KNN, and (c) NB

From the analysis of Figures 6, 7, 8 and 9, several important insights into the effectiveness of the MFS method with different classifiers and feature sizes can be extracts.

- The consistent high performance of MFS across different datasets, classifiers, and feature sizes underscores its efficacy in feature selection for classification tasks.
- MFS generally outperforms alternative methods, showcasing its robustness and versatility in various contexts.
- MFS consistently performs well with the SVM classifier, achieving high scores for both micro-F1 and macro-F1, highlighting its effectiveness.
- MFS demonstrates strong performance with the KNN classifier, particularly in some dimensions for micro-F1 and macro-F1.
- MFS also exhibits effective performance with the NB classifier, achieving high scores in several dimensions for both micro-F1 and macro-F1 criteria across all datasets.

5) STATISTICAL ANALYSIS

Some statistical analyses are performed with the MFS approach, examining whether the proposed method obtains meaningful patterns. One of them is to look at the average scores of the selection approaches across all datasets. Table 8 for the micro-F1 criterion and Table 9 for the macro-F1 criterion are prepared to analyse the average scores of the selection approaches in all datasets.

TABLE 8

AVERAGE MICRO-F1 SCORES FOR ALL FEATURE SIZES USING (A) SVM, (B) KNN, AND (C) NB CLASSIFIERS ON ALL DATASETS.

(A)										
f.size	CHI2	DFS	IG	GI	PS	NDM	MMR	DPM	CMFS	MFS
10	68.9268	72.6245	69.4134	70.7664	69.3745	55.2529	55.9546	58.2793	68.5261	**73.3847**
50	84.6643	84.5889	84.1752	83.7049	84.1054	72.8942	73.5523	69.6204	80.6061	**84.7146**
125	86.0044	86.1622	85.7843	85.7308	85.5109	81.4915	81.2829	82.2444	84.9513	**86.7664**
250	87.2298	87.4732	87.2742	86.8872	87.0680	84.9906	85.8800	85.1128	86.4582	**87.5019**
450	87.0588	87.7456	87.7629	87.1385	87.3765	86.0481	86.1448	86.1185	86.9833	**87.8479**
650	87.8199	87.8319	88.0018	87.2504	87.4008	86.4344	86.5149	86.3626	87.0448	**88.5587**
1125	88.2314	88.0829	88.0492	87.2336	87.6603	87.1780	87.1937	87.1869	87.3444	**88.4261**
(B)										
10	57.3910	53.7600	52.5754	56.2467	57.8572	53.9634	54.2390	41.8235	**58.0704**	55.5791
50	75.0429	73.4149	74.5951	76.4342	76.0504	50.4324	48.7998	50.1687	73.8859	**77.1640**
125	80.0169	79.7111	79.9624	80.5068	80.1493	75.8465	75.0776	76.7609	78.8790	**80.7420**
250	81.8287	81.3910	81.9396	81.1143	81.2519	80.1938	80.9023	80.4417	80.7155	**82.1002**
450	81.8902	82.0402	82.0554	**82.3515**	81.7815	81.7843	82.0921	81.7711	81.9347	82.0124
650	82.0658	82.2428	82.2000	82.8109	81.9011	81.9833	82.5786	81.8846	**82.9046**	82.6554
1125	82.2441	81.9343	82.4836	83.1852	82.0159	83.1274	**83.2341**	83.0736	82.8349	82.9257
(C)										
10	68.7308	72.4518	69.6724	70.2451	67.3954	54.9293	55.0773	58.5375	68.4287	**73.9488**
50	82.5984	82.0975	81.8763	81.4881	81.6601	69.2850	70.4911	66.2854	77.7974	**82.7960**
125	**84.3126**	84.2709	83.7342	83.2245	83.5019	79.1681	79.6728	80.4375	81.8256	83.8816
250	83.6975	**84.3633**	84.0464	83.3533	83.4998	81.0396	82.2834	81.4192	82.1129	84.2552
450	83.2575	**84.2358**	83.3017	83.2496	83.3204	82.7876	82.7684	82.6185	82.7348	84.0092
650	82.8337	**83.7757**	83.1705	83.3598	83.0740	82.5867	82.9346	82.5240	82.3269	83.7617
1125	83.2118	84.4051	83.4108	83.7243	83.0265	82.7672	83.0883	82.9107	83.2804	**84.6072**

TABLE 9

AVERAGE MACRO-F1 SCORES FOR ALL FEATURE SIZES USING (A) SVM, (B) KNN, AND (C) NB CLASSIFIERS ON ALL DATASETS.

(A)										
f.size	CHI2	DFS	IG	GI	PS	NDM	MMR	DPM	CMFS	MFS
10	56.9300	61.9981	59.5854	59.7771	59.9308	36.4983	37.7744	44.7041	57.0970	**62.2033**
50	77.3157	78.2486	77.8969	77.4303	76.9414	60.9186	60.6630	59.3013	71.6116	**78.3762**
125	79.3233	80.3128	79.6538	79.5458	78.9540	73.6694	73.4206	74.9202	77.6273	**80.7047**
250	81.0712	81.6640	81.3043	80.7419	80.7959	78.8240	79.9225	78.8683	80.0187	**81.8068**
450	80.6803	**81.8326**	81.3297	80.7823	81.0374	79.8139	79.9600	79.8826	80.8422	81.6590
650	81.4459	81.6246	81.6562	80.9832	81.0916	80.1654	80.1742	80.1199	80.9471	**82.4263**
1125	81.9043	81.6599	81.7088	80.9372	81.3583	80.9104	80.9056	80.8388	81.0700	**82.2410**
(B)										
10	38.1667	33.8335	36.7575	41.8191	**42.5898**	31.6492	32.6183	26.3925	41.8156	39.6723
50	65.7783	64.6830	66.0935	69.0842	67.7090	39.8548	36.0861	38.3276	64.3488	**69.7334**
125	72.6523	72.9071	72.7439	73.6900	73.0483	67.6056	66.6195	69.0346	71.2012	**74.1075**
250	74.7429	74.5501	75.0944	74.5296	74.3076	72.9458	74.1707	73.3405	73.6538	**75.6546**
450	75.0367	75.2415	75.2231	**75.5091**	74.8881	74.8851	75.1975	74.8395	74.8528	74.9455
650	75.2714	75.4716	75.2541	76.1013	75.0922	75.1609	75.8911	74.9909	76.0927	**76.1532**
1125	75.4480	75.0119	75.6437	76.4826	75.2608	76.5399	**76.6862**	76.4759	76.1677	76.2515
(C)										
10	56.4588	**63.3374**	60.5358	59.9300	58.8259	37.8405	38.7134	47.5046	56.9108	63.1174
50	77.3748	**77.5764**	77.0169	76.7733	76.6955	59.1071	59.5144	58.2930	69.4277	77.3548
125	79.2567	**79.6664**	78.6534	78.2355	78.3904	72.7711	73.6349	75.0734	75.9676	78.5878
250	78.6687	79.5203	79.2275	78.3427	78.3799	75.8340	77.0688	76.2264	77.0511	**79.6687**
450	78.2730	**79.4324**	78.4445	78.2706	78.3862	77.9719	77.9244	77.6491	77.7892	79.0600
650	77.8224	78.9420	78.2728	78.4832	78.2778	77.9167	78.2208	77.8538	77.4067	**79.1047**
1125	78.0649	79.5440	78.4047	78.7507	77.9590	77.9657	78.2204	78.0751	78.3161	**79.7585**

In Table 8, in the micro-F1 criterion, MFS reaches the highest value in all dimensions for the SVM classifier, while it reaches the highest value in dimensions 50, 125 and 250 for KNN. Similarly, the highest values are reached in dimensions 10, 50 and 1125 for BN.

In Table 9, in the macro-F1 criterion, SVM obtained the highest average result in all dimensions except dimension 450. This shows that MFS works more efficiently with SVM and increases the efficiency of the classifier in the features selected. In addition, MFS produced the best results on average with the KNN classifier in both micro-F1 and macro-F1 dimensions 50, 125 and 250. For BN, macro-f1 presented the best average at 250, 650 and 1125 dimensions. These analyses, which examine the performance of the MFS method with different classifiers, reveal that it is particularly compatible with the SVM classifier and generally performs well with the NB and KNN classifiers. These findings suggest that the MFS method can achieve remarkable results when used with different datasets and dimensions. The MFS method achieved the highest average scores for both Mikro-F1 and macro-F1 criteria when used with the SVM classifier. This implies that MFS significantly enhances the effectiveness of the SVM classifier by selecting the most relevant features.

T-test was also used as a statistical analysis to demonstrate the validity of the proposed best-performing MFS. For this purpose, Table 10 and Tablo 11 have been created. Table 10 and Table 11 show the result of the P-values obtained from a one-tailed, paired t-test.

TABLE 10

T-TEST RESULTS OF MFS FOR MICRO-F1 FOR SVM, KNN, AND NB CLASSIFIERS

Classifier	CHI2 & MFS	DFS & MFS	IG & MFS	GI & MFS	PS & MFS	MMR & MFS	NDM & MFS	DPM & MFS	CMFS & MFS
SVM	0.062	0.008*	0.054**	0.001*	0.020**	0.022**	0.024**	0.021**	0.005*
KNN	0.207	0.019**	0.0311**	0.375	0.255	0.105	0.118	0.064	0.223
NB	0.064	0.188	0.0385**	0.013**	0.043**	0.026**	0.034**	0.028**	0.003*

*Significance at 99%.
**Significance at 95%.

In Table 10, the results show that the performance gains from MFS compared to DFS, GI, and CMFS with respect to the micro-F1 metric are statistically significant, with a very high confidence level of 99% for the SVM classifier. Moreover, with the with the same classifier, almost all P-values are above the 95% confidence level, except CHI2. For the other classifiers, this confidence level is at NB, where almost all P-values are above the 95% confidence level, except in a few cases. KNN gave the same confidence level against the DFS and IG methods.

TABLE 11

T-TEST RESULTS OF MFS FOR MACRO-F1 FOR SVM, KNN, AND NB CLASSIFIERS

Classifier	CHI2 & MFS	DFS & MFS	IG & MFS	GI & MFS	PS & MFS	MMR & MFS	NDM & MFS	DPM & MFS	CMFS & MFS
SVM	0.025**	0.025**	0.012**	0.002*	0.000*	0.029**	0.030**	0.023**	0.007*
KNN	0.015**	0.027**	0.019**	0.401	0.215	0.070	0.089	0.065	0.121
NB	0.070	0.148	0.019**	0.017**	0.018**	0.033**	0.039**	0.029**	0.006*

*Significance at 99%.
**Significance at 95%.

The results according to the macro-F1 criterion are given in Table 11. As seen in Table 11, the confidence level for the SVM classifier is 99% for GI, PS, and CMFS, while it is 95% for the other methods. For the KNN and NB classifiers, the confidence level is 99% for NB against CMFS and 95% for the others except for CHI2 and DFS. In KNN, it is 95% against CHI2, DFS, and IG methods. These results verify the superiority of the proposed MFS against other feature selection methods used for comparison.

CONCLUSION

Text data consists of a large number of different words. Classification methods are used in machine learning to categorize such large volumes of text data containing many words. In text classification, each feature does not have the same weight. Some features are more significant in text classification, while some words may be irrelevant or abundant. Selecting class-determining features is important to improve prediction accuracy in text classification. At this point, feature selection approaches have significant advantages in improving classification accuracy.

In this study, a new feature selection approach called MFS is presented as a solution to the problem of high dimensionality, which is one of the main problems in text classification. The main feature of MFS is that it extracts hidden patterns due to multiple variables in text documents and provides an effective selection from them. Experimental results show that MFS works more effectively than existing methods. Moreover, statistical confidence tests of MFS compared to other methods were performed, and it was found to provide high confidence rates of 95% and 99% in general. At the end of the study, an effective feature selection approach for efficient dimensionality reduction in text classification has been introduced to the literature. It will make significant contributions to future studies in this field.

Data accessibility: The Reuters-21578 and AYSC datasets, which support the study's findings, are freely available on Machine Learning Repository-UCI in (Reference 46) and (Reference 47), respectively. ARfSA dataset is available at https://www.kaggle.com/datasets/bittlingmayer/amazonreviews in (Reference 45). Enron1 is a binary-class dataset in (Reference 30).

DECLARATION

Conflict of interest: No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

REFERENCES

[1] Deng, X., Li, Y., Weng, J., & Zhang, J. (2019). Feature selection for text classification: A review. Multimedia Tools and Applications, 78, 3797-3816.

[2] Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. Computers in biology and medicine, 112, 103375.

[3] Kaya, M., Bilge, H. Ş., & Yildiz, O. (2013, April). Feature selection and dimensionality reduction on gene expressions. In 2013 21st Signal Processing and Communications Applications Conference (SIU) (pp. 1-4)

[4] Özseven, T. (2019). A novel feature selection method for speech emotion recognition. Applied Acoustics, 146, 320-326.

[5] Wang, Y. Y., Peng, W. X., Qiu, C. H., Jiang, J., & Xia, S. R. (2019). Fractional-order Darwinian PSO-based feature selection for media-adventitia border detection in intravascular ultrasound images. Ultrasonics, 92, 1-7.

[6] Sanghani, G., & Kotecha, K. (2019). Incremental personalized E-mail spam filter using novel TFDCR feature selection with dynamic feature update. Expert Systems with Applications, 115, 287-299.

[7] Halim, Z., Waqar, M., & Tahir, M. (2020). A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. Knowledge-based systems, 208, 106443.

[8] Parlak, B., & Uysal, A. K. (2020). On classification of abstracts obtained from medical journals. Journal of Information Science, 46(5), 648-663.

[9] Moon, S. H., & Kim, Y. H. (2020). An improved forecast of precipitation type using correlation-based feature selection and multinomial logistic regression. Atmospheric Research, 240, 104928.

[10] Arora, N., & Kaur, P. D. (2020). A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment. Applied Soft Computing, 86, 105936.

[11] Sun, L., Zhang, X., Qian, Y., Xu, J., & Zhang, S. (2019). Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification. Information Sciences, 502, 18-41.

[12] Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on knowledge and data engineering, 17(4), 491-502.

[13] Şenol, A. (2023). Comparison of Performance of Classification Algorithms Using Standard Deviation-based Feature Selection in Cyber Attack Datasets. International Journal of Pure and Applied Sciences, 9(1), 209-222.

[14] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. Computational Statistics & Data Analysis, 143, 106839.

[15] Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. Knowledge-based systems, 86, 33-45.

[16] Kaya, M., & Bilge, H. Ş. (2016, May). A hybrid feature selection approach based on statistical and wrapper methods. In 2016 24th Signal Processing and Communication Application Conference (SIU) (pp. 2101-2104)

[17] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in Icml, 1997, vol. 97, no. 412-420: Nashville, TN, USA, p. 35.

[18] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," Expert Systems with Applications, vol. 33, no. 1, pp. 1-5, 2007.

[19] Y. Li, C. Luo, and S. M. Chung, "Text clustering with feature selection by using statistical data," IEEE Transactions on knowledge and Data Engineering, vol. 20, no. 5, pp. 641-652, 2008.

[20] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," Knowledge-Based Systems, vol. 36, pp. 226-235, 2012.

[21] A. Rehman, K. Javed, and H. A. Babri, "Feature selection based on a normalized difference measure for text classification," Information Processing & Management, vol. 53, no. 2, pp. 473-489, 2017.

[22] A. Rehman, K. Javed, H. A. Babri, and M. N. Asim, "Selection of the most relevant terms based on a max-min ratio metric for text classification," Expert Systems with Applications, vol. 114, pp. 78-96, 2018.

[23] A. Rehman, K. Javed, H. A. Babri, and M. Saeed, "Relative discrimination criterion–A novel feature ranking method for text data," Expert Systems with Applications, vol. 42, no. 7, pp. 3670-3681, 2015.

[24] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili, "A novel multivariate filter method for feature selection in text classification problems," Engineering Applications of Artificial Intelligence, vol. 70, pp. 25-37, 2018.

[25] M. Labani, P. Moradi, and M. Jalili, "A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion," Expert Systems with Applications, vol. 149, p. 113276, 2020.

[26] Chen, C. M., Lee, H. M., & Chang, Y. J. (2009). Two novel feature selection approaches for web page classification. Expert systems with Applications, 36(1), 260-272.

[27] Yang, J., Liu, Y., Zhu, X., Liu, Z., & Zhang, X. (2012). A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. Information Processing & Management, 48(4), 741-754.

[28] Cekik, R., & Uysal, A. K. (2020). A novel filter feature selection method using rough set for short text data. Expert Systems with Applications, 160, 113691.

[29] M. Asim, K. Javed, A. Rehman, and H. A. Babri, "A new feature selection metric for text classification: eliminating the need for a separate pruning stage," International Journal of Machine Learning and Cybernetics, vol. 12, no. 9, pp. 2461-2478, 2021.

[30] Parlak, B., & Uysal, A. K. (2023). A novel filter feature selection method for text classification: Extensive Feature Selector. Journal of Information Science, 49(1), 59-78.

[31] Uysal, A. K., & Gunal, S. (2012). A novel probabilistic feature selection method for text classification. Knowledge-Based Systems, 36, 226-235.

[32] Zhou, H., Ma, Y., & Li, X. (2021). Feature selection based on term frequency deviation rate for text classification. Applied Intelligence, 51, 3255-3274.

[33] Parlak, B. A novel feature ranking algorithm for text classification: Brilliant probabilistic feature selector (BPFS). Computational Intelligence.

[34] Kim, K., & Zzang, S. Y. (2019). Trigonometric comparison measure: a feature selection method for text categorization. Data & Knowledge Engineering, 119, 1-21.

[35] Jin, L., Zhang, L., & Zhao, L. (2023). Max-difference maximization criterion: a feature selection method for text categorization. Frontiers of Computer Science, 17(1), 171337.

[36] Zhu, P., Hou, X., Tang, K., Wang, Z., & Nie, F. (2023). Compactness score: a fast filter method for unsupervised feature selection. Annals of Operations Research, 1-17.

[37] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in Proceedings of the seventh international conference on Information and knowledge management, 1998, pp. 148-155.

[38] Y. Yang, "An evaluation of statistical approaches to text categorization," Information retrieval, vol. 1, no. 1, pp. 69-90, 1999.

[39] B. Scholkopf and A. J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2018.

[40] I. Rish, "An empirical study of the naive Bayes classifier," in IJCAI 2001 workshop on empirical methods in artificial intelligence, 2001, vol. 3, no. 22, pp. 41-46.

[41] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," Computers & security, vol. 21, no. 5, pp. 439-448, 2002.

[42] O. Z. Maimon and L. Rokach, Data mining with decision trees: theory and applications. World scientific, 2014.

[43] Ogura, H., Amano, H., & Kondo, M. (2009). Feature selection with a measure of deviations from Poisson in text categorization. Expert Systems with Applications,, 36(3),6826-6832

[44] Alberto, T. C., Lochter, J. V., & Almeida, T. A. (2015). Tubespam: Comment spam filtering on youtube. Paper presented at the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA).

[45] Kaggle. https://www.kaggle.com/datasets/bittlingmayer/amazonreviews

[46] UCI machine learning repository. https://archive.ics.uci.edu/dataset/137/reuters+21578+text+categorization+collection.

[47] UCI machine learning repository. https://archive.ics.uci.edu/dataset/380/youtube+spam+collection.

[48] Yaqoob, A., Musheer Aziz, R., & verma, N. K. (2023). Applications and techniques of machine learning in cancer classification: a systematic review. Human-Centric Intelligent Systems, 3(4), 588-615.

[49] Joshi, A. A., & Aziz, R. M. (2024). Deep learning approach for brain tumor classification using metaheuristic optimization with gene expression data. International Journal of Imaging Systems and Technology, 34(2), e23007.

[50] Mahto, R., Ahmed, S. U., Rahman, R. U., Aziz, R. M., Roy, P., Mallik, S., ... & Shah, M. A. (2023). A novel and innovative cancer classification framework through a consecutive utilization of hybrid feature selection. BMC bioinformatics, 24(1), 479.

[51] ul Haq, I., Khan, D. M., Hamraz, M., Iqbal, N., Ali, A., & Khan, Z. (2023). Optimal-k nearest neighbours based ensemble for classification and feature selection in chemometrics data. Chemometrics and Intelligent Laboratory Systems, 240, 104882.

[52] Hamraz, M., Khan, Z., Khan, D. M., Gul, N., Ali, A., & Aldahmani, S. (2022). Gene selection in binary classification problems within functional genomics experiments via robust Fisher Score. IEEE Access, 10, 51682-51692.

[53] Khan, S., Khan, M., Iqbal, N., Dilshad, N., Almufareh, M. F., & Alsubaie, N. (2023). Enhancing Sumoylation Site Prediction: A Deep Neural Network with Discriminative Features. Life, 13(11), 2153.

[54] Mamdouh Farghaly, H., & Abd El-Hafeez, T. (2023). A high-quality feature selection method based on frequent and correlated items for text classification. Soft Computing, 27(16), 11259-11274.

[55] Okkalioglu, M. (2024). A novel redistribution-based feature selection for text classification. Expert Systems with Applications, 246, 123119.

RASIM CEKIK received the M.S. and the Ph.D. degree in computer engineering, Eskişehir Technical University, Turkey, in 2015 and 2020, respectively. He completed his PhD in the field of text mining on feature selection in text classification. He has undertaken important work in these areas. He also has important work on machine learning, recommendation systems, and soft computing. He is currently an academician with Şırnak University, Computer Engineering Department, Turkey.