IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 June 2023, accepted 3 July 2023, date of publication 12 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294563

RESEARCH ARTICLE

Alternative Relative Discrimination Criterion Feature Ranking Technique for Text Classification

SARAH ABDULKAREM ALSHALIF[1], NORHALINA SENAN[1], FAISAL SAEED[2], WAD GHABAN[3], NORAINI IBRAHIM[1], MUHAMMAD AAMIR[4], AND WAREESA SHARIF[5]

[1]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor 86400, Malaysia
[2]DAAI Research Group, Department of Computing and Data Science, School of Computing and Digital Technology, Birmingham City University, B4 7XG Birmingham, U.K.
[3]Applied College, University of Tabuk, Tabuk 47512, Saudi Arabia
[4]School of Electronics, Computing and Mathematics, University of Derby, DE22 1GB Derby, U.K.
[5]Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan

Corresponding authors: Norhalina Senan (halina@uthm.edu.my) and Faisal Saeed (faisal.saeed@bcu.ac.uk)

This work was supported in part by the Research Management Center, Universiti Teknologi Malaysia, under Grant Q.J130000.21A6.00P48; and in part by the Data Analytics and Artificial Intelligence (DAAI) Research Group, Birmingham City University, U.K.

ABSTRACT The use of text data with high dimensionality affects classifier performance. Therefore, efficient feature selection (FS) is necessary to reduce dimensionality. In text classification challenges, FS algorithms based on a ranking approach are employed to improve the classification performance. To rank terms, most feature ranking algorithms, such as the Relative Discrimination Criterion (RDC) and Improved Relative Discrimination Criterion (IRDC), use document frequency (DF) and term frequency (TF). TF accepts the actual values of a term with frequently and rarely occurring terms used in existing feature ranking algorithms. However, these algorithms focus on the number of terms in a document rather than the number of terms in the category. In this research, an alternative method to RDC, called Alternative Relative Discrimination Criterion (ARDC) was proposed, which aims to improve the accuracy and effectiveness of RDC feature ranking. Specifically, ARDC is designed to identify terms commonly occurring in the positive class. The results obtained were compared to the existing RDC methods, which are RDC and IRDC, and standard benchmarking functions such as Information Gain (IG), Pearson Correlation Coefficient (PCC), and ReliefF. The experimental results reveal that using the suggested ARDC on the Reuters21578, 20newsgroup, and TDT2 datasets provides better performance in terms of precision, recall, f-measure, and accuracy when employing well-known classifiers such as multinomial naïve Bayes (MNB), Support Vector Machine (SVM), Multilayer perceptron (MLP), k-nearest neighbor (KNN), and decision tree (DT). Another experiment was performed to validate the proposed technique, which aims to showcase the novelty of the ARDC approach. The experiment utilized the 20newsgroup dataset and employed the Relevant-Based Feature Ranking (RBFR) technique. Naïve Bayes (NB), Random Forest (RF) and Logistic Regression (LR) classifiers were used in this experiment to demonstrate the effectiveness of the suggested ARDC.

INDEX TERMS Dimensionality reduction, text classification, feature selection, feature ranking, relative discrimination criterion, accuracy 2 metric.

I. INTRODUCTION

Due to the continuous growth of information technology, the abundance of available information has become a significant challenge. Handling big data has captured the attention of researchers in the fields of artificial intelligence and machine learning. Consequently, intelligent models are necessary to analyze this substantial amount of information, specifically designed for data mining tasks [1], [2]. Web pages, news feeds, electronic mail, and digital libraries provide access to an enormous volume of electronic text content. To address the challenges associated with handling such vast amounts

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

of information, text classification has emerged as a fundamental technology for discovering and categorizing text documents [3].

Classification plays a crucial role in machine learning, particularly in text classification, as it involves automatically sorting a set of text documents into predefined categories [4], [5]. Various machine learning classifiers have been utilized in studies to assess the performance of text classification. The most commonly used classifiers for text classification include Naïve Bayes (NB), Support Vector Machine (SVM), Decision Trees (DT), k-Nearest Neighbor (KNN), and Neural Networks (NN) [6]. However, the excessive dimensionality of the feature space hampers the performance of text classification. Therefore, reducing dimensionality is considered one of the most crucial challenges to overcome in text classification tasks. Feature extraction (FE) and feature selection (FS) are two traditional methods used to address this challenge. FS, used for dimensionality reduction, is essential for improving classification accuracy [7], [8].

FS refers to the procedure of obtaining a subset of the original features based on specific FS criteria, selecting the most important and relevant features from the dataset. FS is widely used and important due to its ability to increase learning accuracy, reduce learning time, and simplify learning results [9]. It enhances the effectiveness of classification by reducing data dimensionality through the elimination of irrelevant and redundant features [10]. Feature selection techniques, including filter-based and wrapper-based methods, are commonly used for FS. Filter-based methods evaluate features independently of the classification algorithm, using statistical measures to rank features based on their relevance to the target class. The filter model can be divided into two categories: feature ranking algorithms and subset search algorithms. Feature ranking is a crucial step in text classification, as it helps identify the most relevant and informative features for a given task. On the other hand, wrapper-based methods evaluate features in conjunction with the classification algorithm, selecting a subset of features and training the classifier on that subset. The performance of the classifier is then used to evaluate the quality of the feature subset. Common wrapper-based methods include forward selection, backward elimination, and genetic algorithms [11].

Various feature ranking techniques are employed to reduce the dimensionality of data such as IG [12], PCC [3], [13], ReliefF [14], [15], RDC [8] and RBFR [16]. IG is commonly used in decision tree-based algorithms for feature selection. It quantifies the reduction in entropy or uncertainty of the target class labels provided by the presence of each term (feature). Features with higher information gain are considered more informative and relevant for classification. Information gain assesses the ability of a feature to split the data into more homogeneous subsets based on class labels [12]. PCC is a statistical measure that quantifies the linear relationship between two variables. In feature ranking, PCC is used to assess the association between each feature and the target class labels. It measures the strength and direction of the linear relationship between a feature and the target class. A high PCC score indicates a strong linear association between the feature and the target class, suggesting its relevance and importance for classification [3], [13]. ReliefF is a feature ranking algorithm commonly used in machine learning tasks, including text classification. It evaluates the relevance of each feature based on the concept of nearest neighbors. ReliefF estimates the quality of features by considering the differences between the feature values of the nearest instances belonging to the same and different classes. It assigns higher weights to features that can effectively discriminate between different classes. ReliefF is particularly useful in handling noisy and redundant features, as it focuses on identifying features that contribute significantly to the classification task [14], [15],

The RBFR algorithm addresses the ranking problem by assigning relevance ranks based on the feature's association with the target class. High weights are given to features that fully represent the class, while features present in multiple classes are less likely to be selected. The RBFR algorithm follows several steps, including ranking features based on true positive rate-false positive rate, removing features with low false positive rate, merging selected features from different algorithms, and re-ranking based on class-specific weights. The algorithm considers metrics like true positive, true negative, false positive, and false negative to determine feature ranks. to mitigate the inclusion of negative features, a secondary filtration step based on false positive rate is applied to eliminate weakly represented features [16].

RDC is a new feature ranking method proposed by Rehman et al. [8] for text data, which enhances the rank of frequently occurring terms presented in one class. RDC calculates the rank of a term by weighing the difference between the true positive rate (tpr) and false positive rate (fpr) for every term count. By incorporating (tpr, fpr) while calculating the rank of the term, RDC can select terms more efficiently for text classification. RDC calculates tpr and fpr for each term count, rather than calculating single values for tpr and fpr for every term. These tpr and fpr values are calculated for frequently occurring terms. In the RDC method, information regarding the term count is included to rank the term, which is ignored in other feature ranking methods such as IG, PCC, and ReliefF. In the RDC method, document frequencies of the term are split into DF of term count to rank the term. Each term is given a term count (TC), which is the total number of occurrences in a document [8]. RDC is a technique that ranks the features of a given text dataset based on their relevance to the classification task. The higher the relevance of a feature to the classification task, the higher its rank will be. RDC considers the number of times a term appears in a document and compares it with the number of times it appears in other documents, then uses this information to assign a relevance score to each feature. However, RDC has a limitation when it comes to text classification tasks involving multiple classes. In such tasks, the dataset is typically split into multiple two-class problems,

where one class is considered positive, and all other classes are combined to form the negative class. RDC focuses only on how many times a term appears in each document, and it ignores how many times a term appears in each category, which can result in high-class skew problems [8]. To address this limitation, the proposed ARDC technique improves RDC using the Alternative Accuracy 2 (AAcc2) metric which proposed by Şahin et al. [17], that takes into account the term count per category to solve the RDC high-class skew problem.

In ARDC, features are ranked using RDC with the AAcc2 metric to identify the most significant features and remove the unbalanced ranking of term frequency. To this end, the ARDC computes the *tpr* and *fpr* for every term count based on the category, using AAcc2. The performance of the ARDC was evaluated using three real-world datasets named Reuter21578, 20newsgroup, and TDT2 in several experiments. According to the reported results, ARDC outperforms IG, PCC, ReliefF, RDC, and IRDC in the most cases.

The rest of the paper is structured as follows: Section II provides a brief summary of previous works. Section III presents the details of the suggested ARDC technique. Section IV describes the experimental procedure, and Section V discusses the findings. Finally, Section VI summarizes the paper and suggests future work.

II. RELATED WORK

Text classification involves assigning documents to one or more categories. The manual classification of texts takes a long time, particularly for large datasets; consequently, automated text classification is increasingly being used in different applications [3], [18]. A text document is a set of words organized in accordance with the corresponding linguistic grammar rules. However, although word arrangement is required to construct meaningful phrases, for text classifiers, the text document is typically depicted as a 'bag of words', in which the word sequence is not taken into consideration in the classification procedure [19]. Consequently, a document D_i is shown as a $vectorD_i = \{TW_{1i}, TW_{2I}, \ldots, TW_{vi}\}$, where TW_{ji} denotes the weight of the *jth* term based on a vocabulary of words $T = \{t_1, t_2, \ldots, t_v\}$. A general method for weighting terms in documents is TF-IDF, where $tf\ (t, D)$ is the term frequency and $idf\ (t, D)$ is the inverse document frequency of the term t in document D [3], [20]. Issues with text classification may involve thousands of features, making it a high-dimensional problem. Although the average collection of texts contains tens of thousands of words. The vast majority of them have little to no information to predict the text label. The relationship among features defines that the feature is constantly very important for determining the class label, thus feature selection is critical not only to enhance classification performance but additionally to decrease storage requirements [3].

Various methods for selecting text features are found in the literature [2], [3], [7], [8], [16], [17], [21], [22], [23], [24], [25]. The paper by Sahin and Kilic [17] proposed two new filter-based feature selection metrics as alternatives to existing ones. The first metric is the relevance frequency feature selection, which adds new parameters to the relevance frequency approach used in text classification. The second metric is the AAcc2, which modifies the parameters of the accuracy 2 metric. The relevance frequency feature selection and AAcc2 metrics were found to be successful compared to existing ones.

Adeleke et al. [2] proposed a two-step feature selection technique for labeling instances of the input data (Quranic verses). The first step involves minimizing the dimensionality of the feature set using the chi-squared filter-based technique, and in the second step, the wrapper is used to further select the most relevant features from the reduced feature set. This method achieved an accuracy result of 93.6% at 4.17 seconds, outperforming the standard filter-based chi-squared and the wrapper correlation-based technique in terms of accuracy and processing time.

Bahassine et al. [23] introduce ImpCHI, a method for improving chi-squared feature selection to enhance the efficiency of classifying Arabic text. The ImpCHI method outperformed other techniques, with the best f-measure of 90.50% obtained on 900 features.

The study presented in [25] proposes a hybrid approach for text classification of Urdu news articles by combining filter feature selection methods such as chi-squared, information gain, and gain ratio with latent semantic indexing. The study used the Urdu dataset called ''ROSHNI''. The results of the proposed method show a superior classification with significant accuracy and efficiency. The proposed method achieves an accuracy of up to 62.57%, which is relatively satisfactory compared to other techniques.

A research work done by [16] introduces a novel algorithm called Relevant-Based Feature Ranking (RBFR) that aims to identify and select smaller subsets of highly relevant features within the feature space. The performance of RBFR is compared against five existing filter-based feature selection methods on three datasets: 20newsgroup, Reuters, and WAP. The evaluation of the RPFR method involves testing it with five machine learning models, namely SVM, NB, KNN, random forest, and logistic regression. The results indicate that the RBFR method achieves a 25.4305% higher accuracy compared to the existing feature selection methods.

Several research adopted Meta-heuristic technique as feature selection such as the study presented by [26] is a novel Meta-heuristic approach called Binary Multi-objective Chimp Optimization Algorithm (BMOChOA), which incorporates a dual archive and a KNN classifier to extract relevant aspects from medical data. To explore the effectiveness of BMOChOA, twelve different versions are implemented based on group information and the types of chaotic functions employed. The performance of these variations is compared with three benchmark multi-objective FS methods, using 14 popular medical datasets of varying dimensions. The evaluation is carried out using four multi-objective performance metrics, and the results indicate that the proposed FS method

excels in achieving the optimal trade-off between the two objective functions: the number of features and classification.

In the study by [27], a novel Meta-heuristic technique called discrete artificial gorilla troop optimization (DAGTO) is introduced for the first time to handle feature selection (FS) tasks in the healthcare sector. Four variants of the proposed method are implemented, depending on the number and type of objective functions: (1) single-objective DAGTO (SO-DAGTO), (2) bi-objective wrapper DAGTO (MO-DAGTO1), (3) bi-objective filter wrapper hybrid DAGTO (MO-DAGTO2), and (4) tri-objective filter wrapper hybrid DAGTO (MO-DAGTO3) for the identification of relevant features in diagnosing a particular disease. An outstanding gorilla initialization strategy is provided based on label mutual information (MI) to increase population variety and accelerate convergence. To verify the performance of the presented methods, ten medical datasets of variable dimensions are taken into consideration. A comparison is also carried out between the best of the four suggested approaches (MO-DAGTO2) and four established multi-objective FS strategies, and its superiority is statistically proven. Finally, a case study is performed with COVID-19 samples to extract critical factors related to it and to demonstrate its fruitfulness in real-world applications.

The work done in [28], aims to predict the health condition of COVID-19 patients by identifying relevant factors using an improved binary multi-objective hybrid filter-wrapper chimp optimization Meta-heuristic (EBMOChOA-FW) based feature selection (FS) approach. In some cases, the initial version of the chimp optimization algorithm (ChOA) may get trapped in local optima. To address this issue, a novel variant called EBMOChOA is developed by integrating the Harris Hawk Optimization (HHO) into the original ChOA. This integration aims to enhance the search capabilities of the optimizer and expand its applicability across various domains. The location change step in the ChOA optimizer is divided into three parts: modifying the population using HHO to create an HHO-based population, generating hybrid entities based on HHO-based and ChOA-based individuals, and adjusting the search agent using a greedy technique and ChOA's tools. The effectiveness of EBMOChOA-FW is demonstrated by comparing it to five well-known algorithms on nine different benchmark datasets. Additionally, its strengths are applied to three real-world COVID-19 datasets to predict the health condition of COVID-19 patients.

While this section reviews the previous research carried out using RDC feature selection in text classification. Various researchers have improved the RDC method, for example, Normalized RDC (NRDC) [24], Improved RDC (IRDC) [7], Multivariate RDC (MRDC) [3], De-redundancy RDC (DRDC) [22], and hybrid RDC with Ant Colony Optimization (RDCACO) [21]. For RDC, it takes into consideration the document frequency for each term count (tc) to define the rank of the term. In unbalanced datasets, document frequencies are measured by the size of the class. The true positive rate (tpr) of a term in the positive class is its normalized document frequency, while in the negative class, the normalized document frequency is its false positive rate (fpr). RDC calculates tpr and fpr for every term count (tc) rather than just calculating a single number for tpr and fpr for a term. The selection criterion used in RDC is as in Equation 1 [8].

$$RDC_{tc} = \frac{\left|tpr_{tc} - fpr_{tc}\right|}{\min\left(tpr_{tc}, fpr_{tc}\right) \times tc} \tag{1}$$

In NRDC the normalized coefficient N defined as in Equation 2, was utilized to remove the term frequency in unbalanced feature ranking.

$$\mathrm{N} = \frac{Avglength}{Length} \tag{2}$$

where $Avglength$ is the average length of the documents in the datasets, and $Length$ is the current length of the documents. Subsequently, the normalized term count (Ntc) can be presented as in Equation 3 and NRDC is calculated as in Equation 4 [24].

$$Ntc = \mathrm{N} * tc \tag{3}$$

$$NRDC = \frac{|tpr - fpr|}{\min\left(tpr, fpr\right)} * Ntc \tag{4}$$

IRDC assigns a high rank to the rare and informative terms for every class used, for performance improvement and to decrease the computational overhead. It makes a trade-off between terms that occur frequently and rarely. Thus, IRDC does not disregard frequent terms; rather, it tends to reduce the number of these terms while increasing the number of rare ones. To assign a high rank to tpr_{tc} and fpr_{tc} for rarely occurring terms, IRDC divides the document frequency of term count by the total of the document frequency of term counts in the positive class and negative class, as shown in Equations 5 and 6 respectively [7].

$$tpr_{tc} = \frac{tp_{tc}}{\sum_{i=0}^{n} tc} \tag{5}$$

$$fpr_{tc} = \frac{fp_{tc}}{\sum_{i=0}^{n} tc} \tag{6}$$

IRDC multiplies the term count (tc) instead of dividing it as defined in RDC Equation 1 that subsequently increases the ranking of rare terms, as shown in Equation 7 [7].

$$\mathrm{I}RDC_{tc} = \frac{\left|tpr_{tc} - fpr_{tc}\right|}{\min\left(tpr_{tc}, fpr_{tc}\right)} \times tc \tag{7}$$

The MRDC focuses on reducing redundant features using the concepts of minimum redundancy and maximum relevance. Thus, MRDC consider the redundancy between the features besides the maximum relevance using a Pearson

correlation coefficient metric, as defined in Equation 8 [3].

$$MRDC_{f_i} = RDC\,(f_i) - \sum_{f_i \neq f_j, f_j \in S} \times \left| \frac{\sum_{d \in |docs|} \left(f_{i,d} - \bar{f}_i\right)\left(f_{j,d} - \bar{f}_j\right)}{\sqrt{\sum_{d \in |docs|} \left(f_{i,d} - \bar{f}_i\right)^2} \sqrt{\sum_{d \in |docs|} \left(f_{j,d} - \bar{f}_j\right)^2}} \right| \quad (8)$$

where $\bar{f}_i$ and $\bar{f}_j$ are mean values of the f_i and f_j vectors, respectively. $f_{i,d}$ is the value of features i and $f_{j,d}$ is the value of features j for dth document. The value of 1 stands for a perfect positive correlation, whereas the value of -1 stands for a perfect negative correlation.

Jin et al. [22] proposed a new technique called De-redundancy Relative Discrimination Criterion (DRDC), which takes into account the redundancy between terms when evaluating their importance. DRDC utilizes both RDC and mutual information to measure the relevance of terms to categories and their redundancy between terms. Respectively, during the selection process, the scores of RDC and mutual information are normalized separately to balance them and reduce the impact of mutual information. To find the optimal term subset, DRDC iteratively selects the term with maximum relevance to categories and minimum redundancy with terms already in the feature subset [22].

Hemmati et al. [21] combine the Relative Discrimination Criterion (RDC) and Ant Colony Optimization (ACO) techniques in a two-stage FS technique. In the first stage, RDC is applied to rank features based on their values, and features with lower values than a threshold is removed from the feature set. In the second stage, an ACO-based feature selection method is applied as a wrapper method to select redundant or irrelevant features that were not removed in the first stage. The experimental results demonstrate the effectiveness of the RDC-ACO method in text feature selection [21].

III. THE SUGGESTED FEATURE RANKING TECHNIQUE: ALTERNATIVE RELATIVE DISCRIMINATION CRITERIA (ARDC)

In In this research, an alternative feature ranking method called the alternative relative discriminative criterion (ARDC) is proposed. The ARDC is specifically designed for text classification tasks and consists of three stages: pre-processing, feature selection, and evaluation. Firstly, the raw text documents underwent various pre-processing methods, including tokenization, stemming, and stop-word removal. These methods were applied to transform the documents into a valuable and proper representation. Then, the terms were converted into real-valued vectors using the bag-of-words method. In the second stage, the proposed ARDC feature ranking criterion was employed to obtain the most significant features and address the issue of unbalanced ranking caused by the high-class skew problem in text classification. This helped eliminate the disparity in term frequency rankings. In ARDC, the alternative Acc 2 metric was applied to calculate the difference between *TPR* and *FPR*. Finally, in the last stage, the ranked features were evaluated using several classifiers, including multinomial naïve Bayes (NB), Support Vector Machine (SVM), Multilayer Perceptron (MLP), k-nearest neighbor (K-NN), and decision tree (DT). These stages are illustrated in Figure 1, while Figure 2 provides the definition of the ARDC Algorithm.

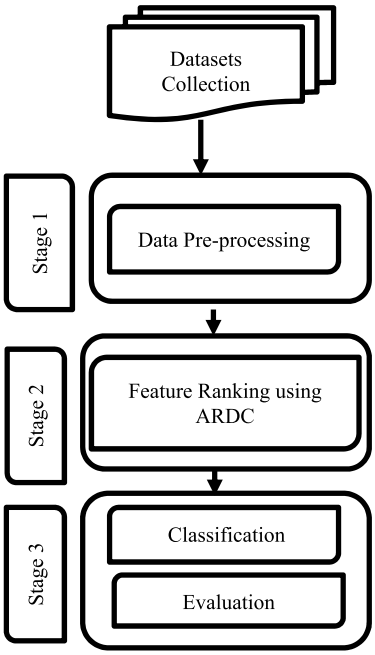


FIGURE 1. The framework of the ARDC.

A. DATASET COLLECTION

The dataset consists of a collection of documents belonging to various categories. These datasets were specifically created to train and evaluate the algorithm's performance when presented with new documents. In this research, three distinct single-labeled datasets were utilized, each characterized by varying dataset sizes and class skews: Reuter21578, 20newsgroup, and TDT2. These datasets, namely Reuter21578, 20newsgroup, and TDT2, are widely regarded as standard datasets for text classification tasks. They are sourced from The UCI Machine Learning Repository, which provides a comprehensive collection of datasets commonly used as benchmarks for various machine learning tasks, including classification. These datasets have been extensively utilized in previous studies [3], [7], [8], [17] within the field. The datasets used are summarized in Table 1.

B. DATA PRE-PROCESSING

In text classification problems, the vector space or bag-of-words model is commonly employed to represent documents. In this model, a document is treated as a collection of its words, disregarding word order and syntax. The frequency of terms is utilized as feature values during classifier training. However, due to the large number of features generated by this model, certain pre-processing methods need to be

Algorithm: alternative relative discrimination criteria (ARDC) algorithm
Begin

Input: Dataset term frequency matrix
Output: 1500 top selected features final list.
POS= documents number in positive class
NEG= documents number in negative class
K= Total number of categories
TC= term counts
TC_MAX=maximum term counts for term T

for TC=1 to TC_MAX **do**

TP_{TC} = positive number of documents containing term T and have term count TC
FP_{TC} = negative number of documents containing term T and have term count TC

$$\mathrm{TPR_{TC}} = \frac{\mathrm{TP_{TC}}}{\mathrm{POS}}$$
$$\mathrm{FPR_{TC}} = \frac{\mathrm{FP_{TC}}}{\mathrm{NEG}}$$
$$\mathrm{D_{TC}} = DF_{TC} \times \left|\mathrm{TPR_{TC}} - \frac{\mathrm{FPR_{TC}}}{\mathrm{K}-1}\right|$$
$$\mathrm{ARDC_{TC}} = \frac{(\mathrm{D_{TC}})}{\min(\mathrm{TPR_{TC}}, \mathrm{FPR_{TC}}) \times TC}$$

end
$AUC_T = 0$
for TC=1 to TC_MAX **do**

$$AUC_T = AUC + \frac{ARDC_{TC} + ARDC_{TC+1}}{2}$$

end
End

FIGURE 2. The algorithm of ARDC.

TABLE 1. Summary of the dataset used.

Dataset	Types of datasets	Total No. of documents	Total No. of features	No. of classes
Reuter21578[7]	Unbalanced	1897	12978	15
20newsgroup[7]	Balanced	3000	9906	10
TDT2[7]	Balanced	500	1000	5

implemented to reduce the high dimensionality of the term space. Tokenization, stop-word removal, and stemming are among the most commonly performed pre-processing tasks in text classification. Tokenization involves breaking down the content of a file into individual words, referred to as tokens. As an initial step, basic filtering is employed to remove various types of characters, such as quotation marks, question and exclamation marks, semicolons, and full stops, in order to standardize the texts. This ensures that the subsequent analysis focuses on meaningful and relevant words. Thereafter, each apostrophe-separated suffix and prefix are eliminated, such as ''it's'' will be ''it'' only. After that, all uppercase characters are converted to lowercase. Finally, the text is tokenized by breaking the stream of text into words. The stop-word removal procedure removes terms from the feature space that are often used but lack discriminatory information. For instance, the words ''a,'' ''the,'' and ''that'' are used frequently in almost all documents and provide little meaningful information for classification. The stemming procedure removes the root forms of the term. As a result, various words with the same root can be recognized in the feature space as the same term. The phrases ''computer'', ''computing'', ''computation'' and ''computes'' for example, are semantically equivalent to their root ''compute.'' Porter's stemmer was used in this study for this purpose [29].

C. FEATURE RANKING USING ARDC

At this point, the suggested ARDC algorithm is utilized to evaluate features, as shown in Figure 2.

The ARDC technique aims to improve feature ranking in text classification by considering the number of times a term appears in a positive class. This is because the frequency of a term in the positive class is crucial for accurate classification. Unlike existing feature ranking algorithms that use document frequency (DF), the ARDC technique uses category count-based measures to rank terms. The key idea of the ARDC technique is to adjust the true positive and false positive rates of the term count in positive and negative classes to assign a high rank to frequent term counts in the positive class. The ARDC technique considers both document frequency (DF) and term count (TC) to determine the rank of a term and boosts the weight of frequent terms in the positive class by dividing the false positive rate (FPR) by the number of categories in the negative class.

In ARDC, as presented in Figure 2, the True Positive Rate (TPR) is the number of documents in the positive class containing the term (T) and having term count (TC) divided by the number of documents in the positive class while the False Positive Rate (FPR) is the number of documents in the positive class containing the term (T) and having the term count (TC) divided by the number of documents in negative class, as shown in Equations 9 and 10 respectively.

$$\mathrm{TPR_{TC}} = \frac{\mathrm{TP_{TC}}}{\mathrm{POS}} \tag{9}$$
$$\mathrm{FPR_{TC}} = \frac{\mathrm{FP_{TC}}}{\mathrm{NEG}} \tag{10}$$

The value of D_{TC} is calculated by dividing FPR_{TC} by the number of categories contained in the negative class, as defined in Equation 11. The resulting term D_{TC} is then used to calculate the ARDC, as shown in Equation 12.

$$\mathrm{D_{TC}} = DF_{TC} \times \left|\mathrm{TPR_{TC}} - \frac{\mathrm{FPR_{TC}}}{\mathrm{K}-1}\right| \tag{11}$$
$$\mathrm{ARDC_{TC}} = \frac{(\mathrm{D_{TC}})}{\min(\mathrm{TPR_{TC}}, \mathrm{FPR_{TC}}) \times TC} \tag{12}$$

where K is the total number of categories (classes) and DF is document frequency, to ensure difference for some terms, because AAcc2 values may be the same in some cases [17].

1) AN ILLUSTRATIVE EXAMPLE

In many text datasets, it is common to encounter texts from more than two categories. When dealing with a multi-class

TABLE 2. Example dataset having twelve documents and four unique terms.

Documents	Class	Documents Content
Document 1	Positive	Pen Ruler
Document 2	Positive	Pen Notebook Ruler
Document 3	Positive	Notebook Ruler
Document 4	Positive	Notebook Pen Ruler Notebook Ruler
Document 5	Positive	Ruler Pen Ruler Pen
Document 6	Positive	Ruler Notebook
Document 7	Negative	Eraser Notebook
Document 8	Negative	Eraser Eraser
Document 9	Negative	Ruler Notebook Ruler
Document 10	Negative	Notebook Notebook
Document 11	Negative	Pen Ruler
Document 12	Negative	Eraser Ruler

TABLE 3. Documents frequency of the terms with their term count.

Term	Pen		Eraser		Notebook		Ruler	
Term count	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
1	3	1	0	2	3	2	4	2
2	1	0	0	1	1	1	2	1
3	0	0	0	0	0	0	0	0

classification problem, it is often divided into several two-class problems. Each two-class problem consists of one positive class and the remaining classes grouped together to form the negative class. This approach enables the application of binary classification techniques to handle the multi-class scenario effectively [17]. An example dataset was used to explain ARDC. Table 2 displays this dataset with twelve documents and the four unique terms, which are Pen, Eraser, Notebook, and Ruler.

Suppose that this dataset has four multiple categories (classes). Thus, the value of K is first determined, which is equal to four. Table 3 then displays the document frequencies for every term at various term counts in both classes.

According to the proposed criteria, the term '*Pen*' stands out as the most significant among the four terms. It exhibits a high frequency in the positive class and appears rarely in the negative class. '*Ruler*' and '*Notebook*' are also considered to have relatively high scores. On the other hand, '*Eraser*' is regarded as the least important term based on the given criteria.

TPR_{TC} is calculated for every term count in the positive class and FPR_{TC} is calculated for every term count in the negative class, while, TPR_{TC} is the term count document frequency for every term. Consequently, FPR_{TC} is calculated by dividing the term count document frequency in the negative class by the number of categories in this class. Table 4 presents the calculation of TPR_{TC} and FPR_{TC} for every term count of a term in the positive and negative classes.

TABLE 4. Calculation of TPR_{TC} and FPR_{TC} in positive and negative classes.

Term	Term Count	TPR_{TC}	FPR_{TC}
Pen	1	Pen_{TPR_1}=3	Pen_{FPR_1}=1/ (4-1) = 0.3333
	2	Pen_{TPR_2}=1	Pen_{FPR_2}=0/ (4-1) = 0
Eraser	1	$Eraser_{TPR_1}$=0	$Eraser_{FPR_1}$=2/ (4-1) = 0.6666
	2	$Eraser_{TPR_2}$=0	$Eraser_{FPR_2}$=1/ (4-1) = 0.3333
Notebook	1	?otebook$_{TPR_1}$=3	?otebook$_{FPR_1}$=2/ (4-1) = 0.6666
	2	?otebook$_{TPR_2}$=1	?otebook$_{FPR_2}$=1/ (4-1) = 0.3333
Ruler	1	Ruler_{TPR_1}=4	Ruler_{FPR_1}=2/ (4-1) = 0.6666
	2	Ruler_{TPR_2}=2	Ruler_{FPR_2}=1/ (4-1) = 0.3333

TABLE 5. ARDC calculation for terms.

Term and Term Count	Pos. (TP	Neg. (FP	Difference ($\lvert D \rvert$)	Minimum (γ)	(ϵ)	$ARDC = \frac{D}{\gamma * tc}$
Pen						
TC 1	3	0.333	2.667	0.333	0.1	2.667/ (0.333*1) = 80.09
TC 2	1	0	1	0	0.1	1/ (0 *2) = 10
Eraser						
TC 1	0	0.666	0.666	0	0.1	0.666/ (0 *1) = 6.66
TC 2	0	0.333	0.333	0	0.1	0.333/ (0 *2) = 3.33
Notebook						
TC 1	3	0.666	2.334	0.666	0.1	2.334/ (0.6666 *1) = 35.05
TC 2	1	0.333	0.667	0.333	0.1	0.667/ (0.333 *2) = 10.02
Ruler						
TC 1	4	0.666	3.334	0.666	0.1	3.334/ (0.666 *1) = 50.06
TC 2	2	0.333	1.667	0.333	0.1	1.667/ (0.333 *2) = 25.03

Table 5 displays ARDC values for various term counts. A term count is given a high ranking by the suggested ARDC if it frequently appears in the positive class. According to [30], if a term appears in one class, the minimal document frequency of that term is calculated as zero and that divides the difference by zero, causing an undefined value. Thus, in order to avoid dividing by zero, the ARDC divides the difference over TPR_{TC} and FPR_{TC} by a small value that is (ϵ), given a value of 0.1, which was used in previous

TABLE 6. AUC for ARDC calculations for terms.

Term	Area Under Curve (AUC)
Pen	[(80.09+10)/2] + [(10+0)/2] = 45.045+5 = 50.05
Eraser	[(6.66+3.33)/2] + [(3.33+0)/2] = 4.995+ 1.665= 6.66
Notebook	[(35.05+10.02)/2] + [(10.02+0)/2] = 22.535+5.01= 27.545
Ruler	[(50.06+25.03)/2] + [(25.03+0)/2] = 37.545+12.515=50.06

studies [3], [7], [8]. The other essential factor for deciding term rank is the term count. Typically, when the term count values increase, the document frequency of this term count decreases and drops until it reaches zero and a difference in higher term counts will have greater benefits than lower term counts when dividing by the factor (ϵ). To assign a higher weight to the difference between TPR_{TC} and FPR_{TC}, they are divided by multiplying the minimum of TPR_{TC} and FPR_{TC} with the term count (TC). This loop is continued until the TC_{MAX} count is found. The final value of the term, T, is found through AUC_T and then the algorithm stops.

In alignment with the procedures in earlier research [3], [7], [8], [24], ARDC also considers the area under the curve (AUC) for term rank, shown in Table 6. As shown in Table 6, the highest area under the curve (AUC) is assigned to the terms 'Pen' and 'Ruler', which frequently occur in the positive class, followed by the term 'Notebook'; lastly is the term 'Eraser', which is the least important.

D. CLASSIFICATION

The experiments were conducted using widely recognized classifiers for text classification: NB, MNB, SVM, MLP, KNN, DT, RF, and LR. These classifiers were chosen based on their established effectiveness in text classification tasks.

NB is a widely utilized classifier in the domain of text classification. This model operates on the probabilistic principles of Bayes theorem. It categorizes instances based on their similarity and predicts the class of a new sample by assessing its relationship with each class [16]. MNB is a probabilistic classifier that assumes the independence of input features given the target class. It leverages the probabilities of features occurring in different classes to make predictions [3]. SVM, introduced by Cortes and Vapnik [31], is a supervised technique within the realm of statistical learning. It is widely used to distinguish between linear and non-linear data, and it possesses robust predictive capabilities to address non-linear problems. Beyond classification tasks, SVM is also a valuable tool in regression and clustering applications due to its versatility and effectiveness [32]. MLP is a type of neural network that operates using supervised learning. It consists of three fundamental layers: the input layer, hidden layer(s), and output layer. MLP is a self-adaptive and data-driven technique that can organize these layers based on the provided data, without requiring a specific specification for the functional or distributional structure of the underlying model. This flexibility allows MLP to effectively learn complex patterns and relationships within the data [33]. Whereas KNN is a classification algorithm that involves determining the k-nearest training vectors to a given instance and assigning the class of the new instance based on the most frequent category or label among its closest neighbors. The Euclidean distance formula is commonly employed in KNN classifiers to measure the distance between pairs of vectors, which helps determine the proximity of instances in the feature space. By leveraging the concept of proximity, KNN can make predictions based on the characteristics of similar instances in the training data [34], DT is a machine learning algorithm that learns simple decision rules and constructs a hierarchical structure to estimate the target value of a variable based on the provided training data. It recursively partitions the feature space based on the values of input features to create a tree-like structure. At each internal node of the tree, a decision rule is defined based on a specific feature, and the tree branches out accordingly. Ultimately, the leaf nodes of the decision tree provide the estimated target values based on the learned patterns from the training data [3]. While. RF is a classifier that belongs to the ensemble-based family of algorithms. It leverages multiple decision trees in its classification process. The number of decision trees is predetermined prior to the start of classification. Each decision tree is trained independently on a distinct subset of inputs. Subsequently, the outputs of each decision tree are combined through a majority voting scheme to determine the final class assignment [16]. LR is a specialized type of classifier that is specifically designed for classifying linearly separable data. It builds a decision boundary, also known as a margin, to distinguish between different classes. When new instances are encountered, LR assigns them a class based on their position relative to the margin. Instances located on one side of the margin are assigned to one class, while those on the other side are assigned to the opposite class [16].

All classifiers were used with default settings and implementation of the machine learning Toolkit WEKA (Waikato Environment for Knowledge Analysis) version 3.8.4 [35]. It is a Java open-source platform that comprises several machine learning algorithms. The default number of iterations in the WEKA toolkit to produce statistically relevant results is ten. In ten cross-validation, datasets are randomly partitioned into ten folds that are not dependent on each other. The training process is repeated ten times, and the testing process is also repeated ten times [7].

E. EVALUATION

Based on confusion metrics, four basic rules are used to evaluate the performance of an algorithm, as follows:

i True-positive (TP): a result where the model correctly predicts the positive class.
ii False-negative (FN): a result where the model incorrectly predicts the negative class.

TABLE 7. Confusion metrics.

Class	T_j	$\hat{T}_j$
Positive Class	TP	FN
Negative Class	FP	TN

iii False-positive (FP): a result where the model incorrectly predicts the positive class.
iv True-negative (TN): a result where the model correctly predicts the negative class.

Table 7 shows some basic guidelines for measuring the performance of the algorithm using the confusion matrix.

In text classification techniques, precision (P), recall (R), f-measure (FM), and accuracy ($ACCU$) are usually used to evaluate performance, where precision (P) is the ratio of TP to the total of TP and FP, The recall is calculated as the ratio of TP to the total of TP and FN, F-measure is based on precision and recall, it is the harmonic mean, which combines recall and precision, while, accuracy is the ratio of the correctly identified objects, TP and TN, to the total number of objects, TP, TN, FN and FP. The following equations give the formal definitions of these measurements respectively.

$$P = \frac{TP}{TP + FP} \tag{13}$$

$$R = \frac{TP}{TP + FN} \tag{14}$$

$$FM = 2 \times \frac{P \times R}{P + R} \tag{15}$$

$$ACCU = \frac{TP + TN}{TP + FP + TN + FN} \tag{16}$$

IV. RESULTS AND DISCUSSION

In this section, the performances of two existing feature ranking algorithms IG, PCC, ReliefF, RDC and IRDC were compared with the proposed ARDC algorithm. In these experiments, ARDC was implemented using Java programming language using Eclipse IDE 2020-09 in the data pre-processing stage. In the classification and evaluation stages, the WEKA tool employed using common classifiers: MNB, SVM, MLP, KNN, and DT. The ARDC algorithm was evaluated on three text datasets (Reuter21578, 20newsgroup, and TDT2) taken from the UCI machine learning repository. They were run consecutively on a computer with an Intel Core i7 processor, 8GB of total main RAM, and Dell OS 64-bit as the operating system. In addition, the experimental results were validated in terms of the number of selected features and the performance of the classifiers using precision, recall, f-measure, and accuracy measurement criteria.

The results obtained for ARDC, RDC, IRDC, IG, PCC, and ReliefF for the different classifiers (MNB, SVM, MLP, KNN, and DT) are shown in Tables 8, 9, 10, 11, and 12 respectively.

The results in Table 8 show that the performance of ARDC in MNB is better than that of RDC, IRDC, IG, PCC, and ReliefF in all cases except in precision and f-measure of the TDT2 dataset, where RDC performed better than ARDC and IRDC. Particularly, the ARDC produced the highest accuracy 68.8 %, which is better than that of the RDC (55.8%), IRDC (55.5%), IG (61.6%), PCC (60%), and ReliefF (60.8%) for the Rueter21578 dataset. While the ARDC achieved the highest accuracy for the 20newsgroup dataset, at 28.3%, it outperformed all the RDC (18.3%), the IRDC (17.4%), IG (26.3%), the PCC (25.6%), and the ReliefF (27.5%). Additionally, the ARDC produced 52% accuracy of the TDT2 dataset, which is higher than the results of the

TABLE 8. Comparison of ARDC with Existing techniques in terms of precision, recall, F- measures and accuracy using the MNB classifier.

Datasets	Feature selection techniques	Precision	Recall	F-measure	Accuracy
Reuter 21578	ARDC	**0.701**	**0.688**	**0.686**	**68.8**
	RDC	0.181	0.558	0.273	55.8
	IRDC	0.097	0.555	0.165	55.5
	IG	0.622	0.616	0.616	61.6
	PCC	0.616	0.608	0.609	60
	ReliefF	0.612	0.608	0.603	60.8
20news group	ARDC	**0.368**	**0.283**	**0.320**	**28.3**
	RDC	0.213	0.183	0.138	18.3
	IRDC	0.226	0.174	0.129	17.4
	IG	0.265	0.263	0.264	26.3
	PCC	0.256	0.256	0.256	25.6
	ReliefF	0.271	0.275	0.273	27.5
TDT2	ARDC	0.499	**0.520**	0.494	**52**
	RDC	**0.552**	0.516	**0.519**	51.6
	IRDC	0.394	0.402	0.361	40.2
	IG	0.417	0.416	0.416	41.6
	PCC	0.437	0.436	0.436	43.6
	ReliefF	0.423	0.422	0.423	42.2

TABLE 9. Comparison of ARDC with existing techniques in terms of precision, recall, F- measures and accuracy using the SVM classifier.

Datasets	Feature selection techniques	Precision	Recall	F-measure	Accuracy
Reuter 21578	ARDC	**0.956**	**0.747**	**0.839**	**74.7**
	RDC	0.870	0.717	0.786	71.7
	IRDC	0.685	0.601	0.640	60.1
	IG	0.695	0.696	0.695	69.6
	PCC	0.671	0.679	0.675	67.9
	FeliefF	0.622	0.683	0.651	68.2
20news group	ARDC	**0.840**	**0.811**	**0.811**	**81.1**
	RDC	0.452	0.437	0.422	43.7
	IRDC	0.433	0.418	0.392	41.8
	IG	0.722	0.700	0.702	70
	PCC	0.702	0.685	0.687	68.5
	FeliefF	0.626	0.616	0.619	61.6
TDT2	ARDC	**0.819**	**0.826**	**0.821**	**82.6**
	RDC	0.797	0.800	0.796	80.0
	IRDC	0.587	0.588	0.580	58.8
	IG	0.709	0.706	0.707	70.6
	PCC	0.732	0.730	0.731	73
	FeliefF	0.726	0.724	0.725	72.4

TABLE 10. Comparison of ARDC with Existing techniques in terms of precision, recall, F- measures and accuracy using the MLP classifier.

Datasets	Feature selection techniques	Precision	Recall	F-measure	Accuracy
Reuter 21578	ARDC	**0.244**	**0.646**	**0.354**	**64.6**
	RDC	0.201	0.483	0.284	48.3
	IRDC	0.177	0.167	0.172	16.7
	IG	0.106	0.131	0.117	13.1
	PCC	0.260	0.353	0.358	35.3
	FeliefF	0.257	0.458	0.330	45.8
20news group	ARDC	0.020	0.100	0.033	10.3
	RDC	0.021	0.106	0.035	10.6
	IRDC	0.171	0.129	0.210	12.8
	IG	0.127	0.114	0.120	11.4
	PCC	**0.375**	**0.131**	**0.194**	**13.1**
	FeliefF	**0.048**	**0.106**	**0.066**	**10.6**
TDT2	ARDC	**0.868**	**0.862**	**0.863**	**86.2**
	RDC	0.790	0.800	0.794	80.0
	IRDC	0.625	0.606	0.613	60.6
	IG	0.444	0.404	0.342	40.4
	PCC	0.658	0.520	0.468	52
	FeliefF	0.444	0.398	0.358	39.8

TABLE 11. Comparison of ARDC with existing techniques in terms of precision, recall, F- measures and accuracy using the KNN classifier.

Datasets	Feature selection techniques	Precision	Recall	F-measure	Accuracy
Reuter 21578	ARDC	**0.777**	**0.772**	**0.773**	**77.2**
	RDC	0.418	0.733	0.532	73.3
	IRDC	0.661	0.671	0.665	67.1
	IG	0.708	0.716	0.712	71.6
	PCC	0.750	0.725	0.737	72.5
	FeliefF	0.743	0.742	0.742	74.2
20news group	ARDC	**0.731**	**0.726**	**0.727**	**72.6**
	RDC	0.569	0.565	0.566	56.5
	IRDC	0.599	0.600	0.598	60.0
	IG	0.568	0.562	0.562	56.2
	PCC	0.574	0.558	0.511	55.2
	FeliefF	0.493	0.479	0.481	47.9
TDT2	ARDC	**0.872**	**0.872**	**0.871**	**87.2**
	RDC	0.870	0.866	0.866	86.6
	IRDC	0.581	0.576	0.578	57.6
	IG	0.657	0.574	0.599	57.4
	PCC	0.711	0.660	0.629	66
	FeliefF	0.735	0.666	0.643	66.6

RDC (51.6%), IRDC (40.2%), IG (41.6%), PCC (43.6%), and ReliefF (42.2%).

Based on the results presented in Table 9, the ARDC method outperforms all techniques for the SVM classifier in terms of overall performance. The accuracy achieved by ARDC was the highest for the Reuter21578 dataset at 74.7%, while RDC, IRDC, IG, PCC, and ReliefF achieved 71.7%, 60.1%, 69.6%, 67.9%, and 68.2%, respectively. In addition,

TABLE 12. Comparison of ARDC with existing techniques in terms of precision, recall, F- measures and accuracy using the DT classifier.

Datasets	Feature selection techniques	Precision	Recall	F-measure	Accuracy
Reuter 21578	ARDC	**0.989**	**0.989**	**0.989**	**98.9**
	RDC	0.707	0.900	0.792	90.0
	IRDC	0.768	0.902	0.830	90.2
	IG	0.856	0.853	0.853	85.3
	PCC	0.847	0.843	0.844	84.3
	FeliefF	0.836	0.832	0.833	83.2
20news group	ARDC	**0.999**	**0.999**	**0.999**	**99.9**
	RDC	0.761	0.759	0.760	75.9
	IRDC	0.882	0.882	0.882	88.2
	IG	0.671	0.666	0.667	66.6
	PCC	0.665	0.656	0.659	65.6
	FeliefF	0.606	0.602	0.601	60.2
TDT2	ARDC	**0.949**	**0.948**	**0.948**	**94.8**
	RDC	0.917	0.914	0.915	91.4
	IRDC	0.794	0.792	0.792	79.2
	IG	0.893	0.892	0.891	89.2
	PCC	0.902	0.900	0.900	90
	FeliefF	0.891	0.890	0.890	89

ARDC produced a higher accuracy of 81.1% for the 20newsgroup dataset compared to RDC, IRDC, IG, PCC, and ReliefF, which achieved 43.7%, 41.8%, 70%, 68.5%, and 61.6% respectively. Furthermore, the accuracy achieved by ARDC for the TDT2 dataset was 82.6%, which was higher than that of RDC (80.0%), IRDC (58.8%), IG (70.6%), PCC (73%), and ReliefF (72.4%).

Table 10 shows that the ARDC method outperforms RDC, IRDC, IG, PCC, and ReliefF methods in terms of overall performance for MLP except for the 20newsgroup dataset. The Reuter21578 dataset achieved the highest accuracy with ARDC at 64.6%, while RDC, IRDC, IG, PCC, and ReliefF achieved 48.3.3%, 16.7%, 13.1%, 35.3%, and 45.8% respectively. While, PCC achieved a higher accuracy of 13.1% for the 20newsgroup dataset, whereas ARDC, RDC, IRDC, IG, and ReliefF achieved 10.3%, 10.6%, 12.8%, 11.4%, and 10.6% respectively. Moreover, the TDT2 dataset achieved an accuracy of 86.2% with ARDC, which is higher than RDC (80.0%), IRDC (60.6%), IG (40.4%), PCC (52%), and ReliefF (39.8%).

According to the results of KNN in Table 11, the overall performance of ARDC is better than that of RDC, IRDC, IG, PCC, and ReliefF. Moreover, the ARDC provided the highest accuracy in the Reuter21578 dataset, at 77.2%, outperforming the RDC (73.3%), IRDC (67.1%), IG (71.6%), PCC (72.5%), and ReliefF (74.2%). The ARDC produced a higher accuracy of the 20newsgroup dataset, at 72.6%, than did the RDC, IRDC, IG, PCC, and ReliefF (56.5%, 60%, 56.2%, %%.2%, and 47.9%, respectively). The ARDC also gave the highest accuracy on the TDT2 dataset, exceeding the RDC (86.6%), IRDC (57.6%), IG (57.4%), PCC (66%), and ReliefF (66.6%) with a performance of 87.2%.

TABLE 13. Comparison of ARDC with RDC and IRDC in terms of precision, recall using the Reuter21578 dataset.

classifiers	Performance measurements	Number of selected features							
		10	20	50	100	200	500	1000	1500
MNB	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
SVM	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
MLP	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
KNN	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
DT	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**

In addition, ARDC outperformed RDC, IRDC, IG, PCC, and ReliefF in all cases using the DT classifier, as shown in Table 12. For the Reuter21578, the ARDC produced the highest accuracy 98.9 %, which is considerably better than RDC (90%), IRDC (90.2%), IG (85.3%), PCC (84.3%), and ReliefF (83.2%). Whereas the ARDC accomplished the highest accuracy for the 20newsgroup dataset, at 99.9%, it greatly outperformed both the RDC (75.9%), IRDC (88.2%), IG (66.6%), PCC (65.6%), and ReliefF (60.2%). Moreover, the ARDC produced 94.8% of the TDT2 dataset, which is higher than the results produced by the RDC (91.4%), IRDC (79.2%), IG (89.2%), PCC (90%), and ReliefF (89%).

The aim of feature selection ranking methods is to assign a rank value to each feature indicating its level of significance. To objectively evaluate the performance of the ARDC, an equal number of features were selected and compared with existing methods such as RDC and IRDC. The performance of each technique was assessed by varying the number of features considered, ranging from the top 10, 20, 50, 100, 200, 500, 1000, to 1500 features. This evaluation allowed for a comprehensive analysis of how well the ARDC approach performed in comparison to other techniques across different feature subset sizes [3], [7], [8].

Tables 13, 14 and 15 show the number of times ARDC produced superior results compared to RDC and IRDC in terms of precision, recall, and F-measure values when used on the Reuter21578, 20newsgroup and TDT2 datasets respectively. Table 13 shows that ARDC obtained better results for all classifiers for all the numbers of features using Reuter21578 dataset. Similarly, ARDC outperformed RDC and IRDC in most of the cases when used on the 20newsgroup datasets (Table 14) except for MLP classifier on the top 1500 shows that IRDC perform better than ARDC and RDC. Table 15 show that, for the TDT2 dataset, ARDC produced good results for the SVM, KNN, and DT classifiers. However, for MNB, ARDC produced better results for the top 20 features, in terms of precision and F-measure. In terms of recall, RDC and ARDC obtained the same results in term of 20 feature and ARDC produced good results in terms of recall for the top 100, 200, 500 and 1000. While for the MLP classifier ARDC produced better result in most of the cases except in the top of 10 features RDC produced better results in term of precision and F-measure.

For the accuracy evaluation metric, it has been graphically depicted for the Reuter21578, 20newsgroup, and TDT2 dataset respectively in the Figures 3-17 as detailed in the following.

Figures 3-7 present the accuracy of the proposed ARDC in comparison to the RDC and IRDC on the Reuter21578 dataset. The results revealed that the proposed ARDC had superior accuracy in almost all cases than both the RDC and IRDC. Figure 3 showed that the ARDC with the MNB classifier reached the highest accuracy on the top 100 features with a performance of 69.3%, the ARDC outperformed both the RDC (36.6%) and the IRDC (30.6%). Furthermore, the ARDC produced 74.7% in Figure 4 of the SVM classifier while the RDC (71.7%) and the IRDC (60.1%). Otherwise, related to the MLP classifier, RDC (78.9%) and IRDC (70.9%), the ARDC produced a higher accuracy in Figure 5 of the MLP classifier at 82.6%. For the KNN classifier, the ARDC also provided the top in Figure 6 with the highest accuracy, beyond the RDC (79.1%) and IRDC (75%) with a performance of 89.2%. Finally, the DT classifier in Figure 7

TABLE 14. Comparison of ARDC with RDC and IRDC in terms of precision, recall using the 20newsgroup dataset.

classifiers	Performance measurements	Number of selected features							
		10	20	50	100	200	500	1000	1500
MNB	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
SVM	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
MLP	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**IRDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**IRDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**IRDC**
KNN	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
DT	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**

TABLE 15. Comparison of ARDC with RDC and IRDC in terms of precision, recall, F- measures and accuracy using the TDT2 dataset.

Classifiers	Performance measurements	Number of selected features						
		10	20	50	100	200	500	1000
MNB	Precision	RDC	**ARDC**	RDC	RDC	RDC	RDC	RDC
	Recall	RDC	**ARDC/ RDC**	RDC	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	RDC	**ARDC**	RDC	RDC	RDC	RDC	RDC
SVM	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
MLP	Precision	**RDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC/ RDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**RDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
KNN	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
DT	Precision	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	Recall	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**
	F-measure	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**	**ARDC**

the ARDC outperformed the RDC and IRDC (92.2% and 91.7%, respectively) in terms of accuracy with a result of 99.1%.

Figure 8-12 compares the accuracy of the proposed ARDC to the RDC and IRDC with MNB, SVM, MLP, KNN, and DT using 20newsgroup dataset. The results indicate that for all classifiers, the proposed ARDC's accuracy is higher than that of the RDC and IRDC. Except for 1500 features for the MLP classifier IRDC is higher than ARDC and RDC as well. The MNB classifier in Figure 8 achieved the highest accuracy with a performance of 42.3% for the ARDC which outperformed both the RDC (22.4%) and the IRDC (18.4%). In addition, the ARDC generated 81.3% in Figure 9 of the SVM classifier, outperforming the RDC (43.8%) and

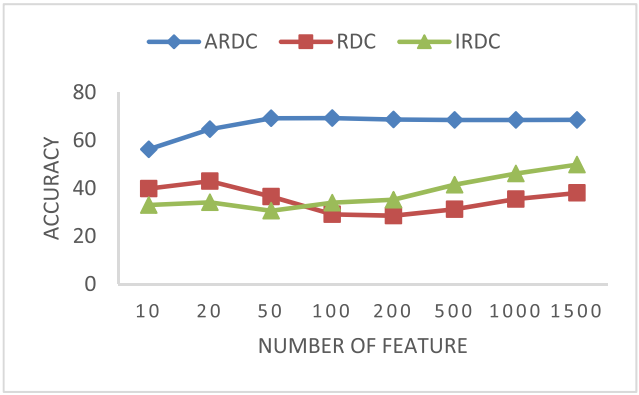


FIGURE 3. Accuracy comparison of ARDC, RDC and IRDC using Reuter21578 dataset and MNB.

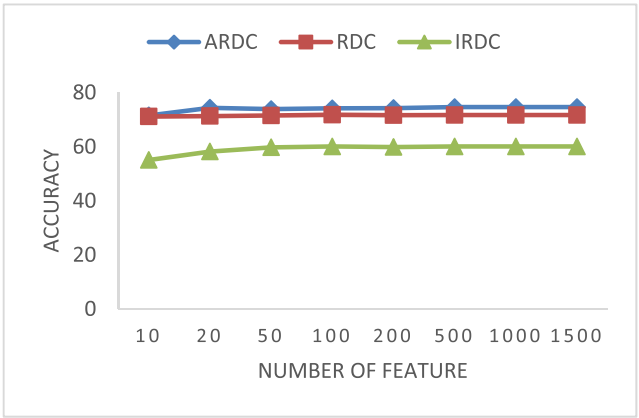


FIGURE 4. Accuracy comparison of ARDC, RDC and IRDC using Reuter21578 dataset and SVM.

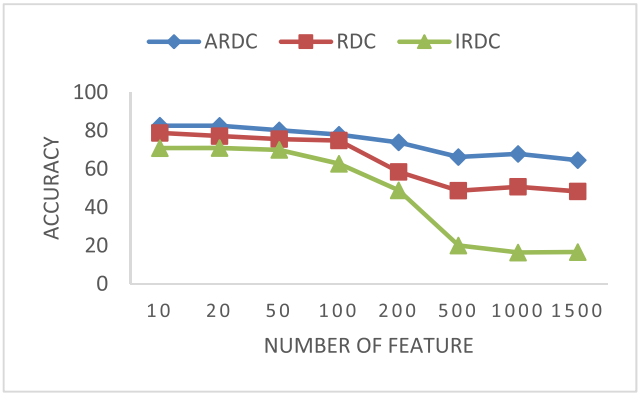


FIGURE 5. Accuracy comparison of ARDC, RDC and IRDC using Reuter21578 dataset and MLP.

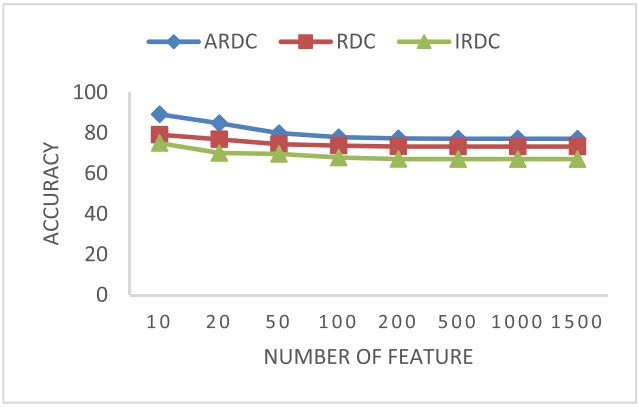


FIGURE 6. Accuracy comparison of ARDC, RDC and IRDC using Reuter21578 dataset and KNN.

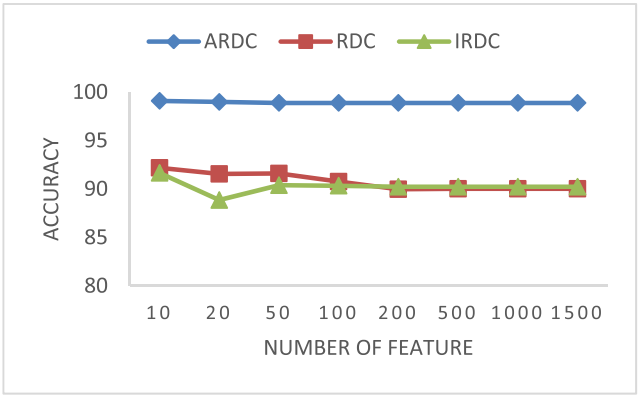


FIGURE 7. Accuracy comparison of ARDC, RDC and IRDC using Reuter21578 dataset and DT.

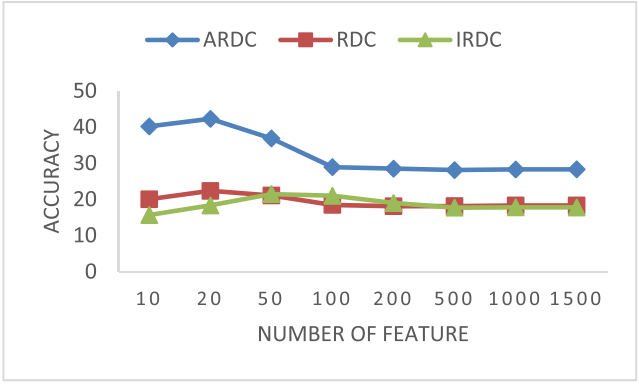


FIGURE 8. Accuracy comparison of ARDC, RDC and IRDC using 20newsgroup dataset and MNB.

the IRDC (41.6%). Otherwise, comparing the RDC (47.4%) and IRDC (49.3%), the ARDC produced a higher accuracy in Figure 10 of the MLP classifier at 94.7%. However, the ARDC also gave the top in Figure 11 with the highest accuracy, exceeding the RDC (59.4%) and IRDC (66.7%) with a performance of 97.7%. Finally, The ARDC outperformed the RDC and IRDC (77.7% and 89.7%, respectively) in terms of accuracy on the DT classifier, with a result of 99.9% showed in Figure 12.

Using the TDT2 dataset, Figures 13-17 present the accuracy of the proposed ARDC in comparison to the RDC and IRDC. The results revealed that the proposed ARDC had superior accuracy in almost all cases than both the RDC and IRDC except for some cases in MNB. Figure 13 showed that the MNB classifier achieved the highest accuracy for

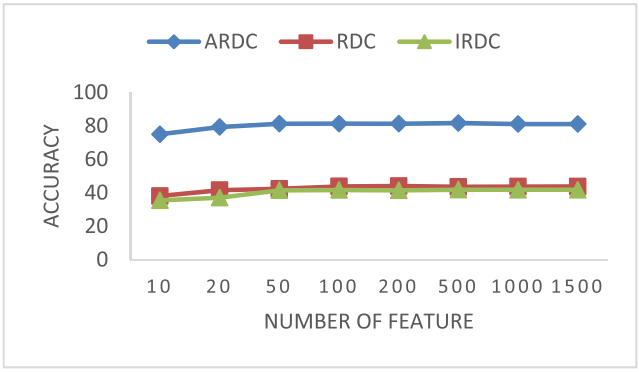


FIGURE 9. Accuracy comparison of ARDC, RDC and IRDC using 20newsgroup dataset and SVM.

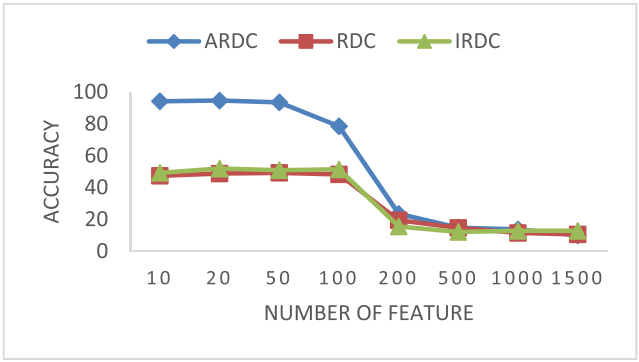


FIGURE 10. Accuracy comparison of ARDC, RDC and IRDC using 20newsgroup dataset and MLP.

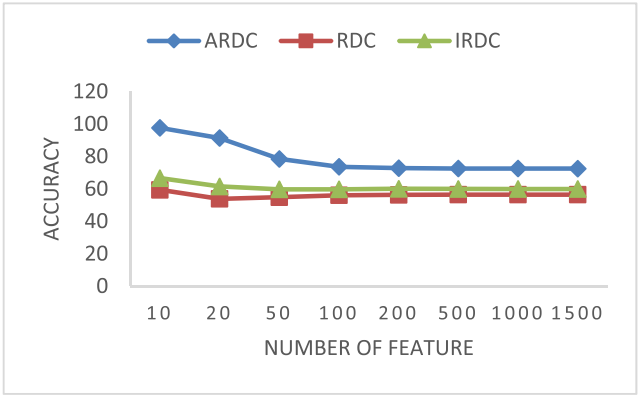


FIGURE 11. Accuracy comparison of ARDC, RDC and IRDC using 20newsgroup dataset and KNN.

both ARDC and RDC with a performance of 55.2%, the ARDC and RDC outperformed the IRDC (43.6%). In addition, Figure 14 of the SVM classifier, shows that the ARDC outperformed the RDC and IRDC with an accuracy rate (ARDC 83.2%, RDC 80.8%, IRDC 58.4%). For the MLP classifier, the ARDC produced a higher accuracy in Figure 15 at 90.4% compared to the RDC (86.4%) and IRDC (66.4%). As well as, the ARDC gave the high performance in Figure 16 with the highest accuracy, exceeding the RDC (87.4%) and

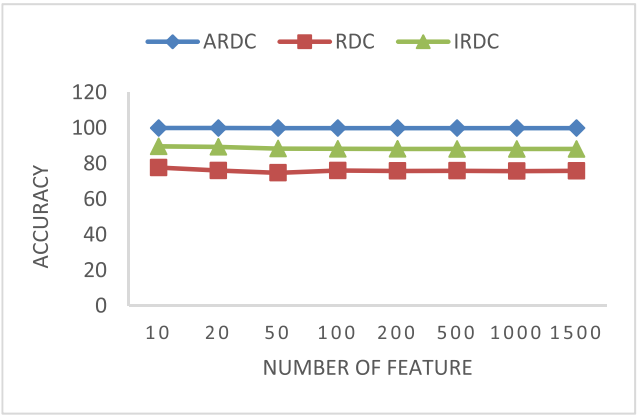


FIGURE 12. Accuracy comparison of ARDC, RDC and IRDC using 20newsgroup dataset and DT.

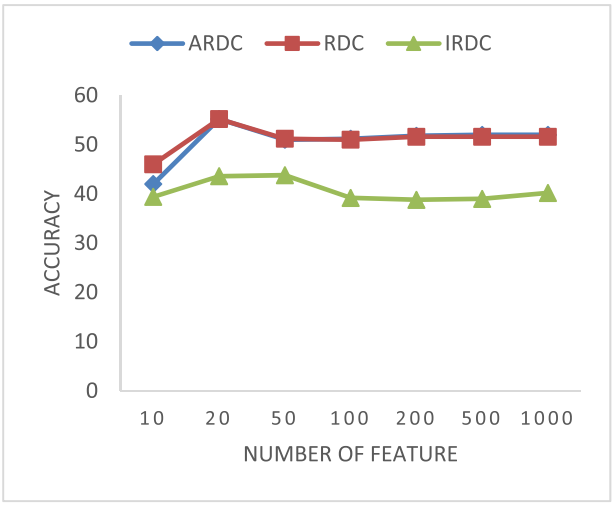


FIGURE 13. Accuracy comparison of ARDC, RDC and IRDC using TDT2 dataset and MNB.

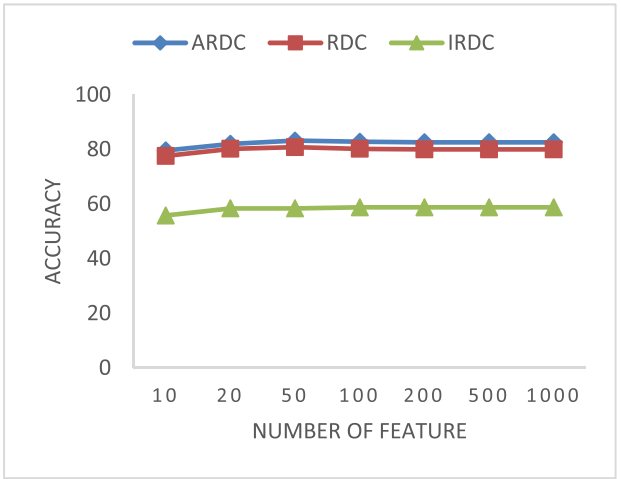


FIGURE 14. Accuracy comparison of ARDC, RDC and IRDC using TDT2 dataset and SVM.

IRDC (68%) with a performance of 91.6% for the KNN classifier. Finally, the ARDC outperformed the RDC and

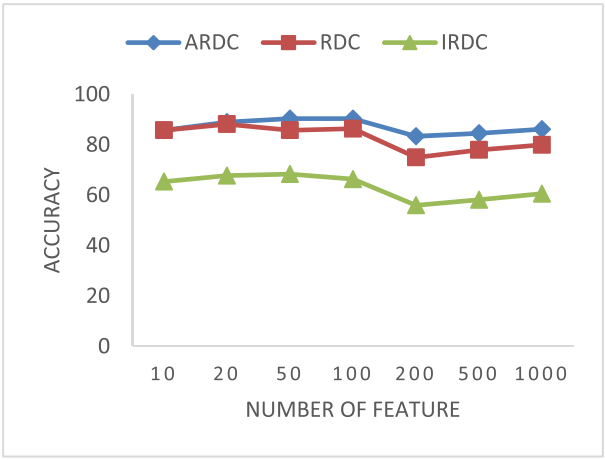


FIGURE 15. Accuracy comparison of ARDC, RDC and IRDC using TDT2 dataset and MLP.

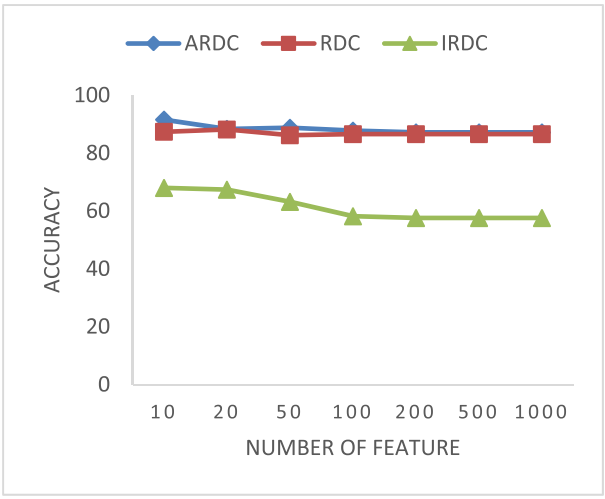


FIGURE 16. Accuracy comparison of ARDC, RDC and IRDC using TDT2 dataset and KNN.

IRDC (93.2% and 81%, respectively) in terms of accuracy on the DT classifier, with a result of 95.4%.

It is concluded that ARDC produced better results with almost all the cases especially for the large dataset having a large number of features. However, overall results of the proposed technique ARDC are better than that of RDC and IRDC in term of precision, recall, F-measure, and accuracy using five classifiers: MNB, SVM, MLP, KNN and DT and three different datasets. Besides, in order to validate the proposed ARDC approach and demonstrate its novelty, an experiment was performed using the 20newsgroup dataset. The experiment specifically employed the RBFR technique. To assess the effectiveness of the ARDC approach, three different classifiers, namely NB, RF, and LR, were utilized in the experiment. This allowed for a comprehensive evaluation of how well the ARDC approach performed with different classifiers, providing insights into its effectiveness and

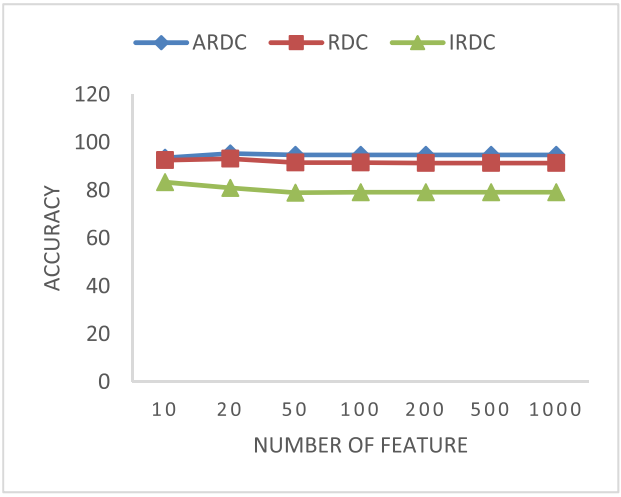


FIGURE 17. Accuracy comparison of ARDC, RDC and IRDC using TDT2 dataset and DT.

TABLE 16. Accuracy comparison.

Classifiers	RPFR[16]	ARDC
NB	**93.69**	92.73
RF	92.47	**94.87**
LR	*87.01*	**96.4**

applicability. Table 16 demonstrate the comparison of ARDC with RBFR in term of accuracy.

The experimental results indicated that the ARDC approach consistently achieved higher accuracy in the RF and LR classifiers compared to the RPFR approach, with the exception of NB classifier. These findings demonstrate that, in general, the ARDC approach outperforms the RPFR approach in terms of accuracy.

V. CONCLUSION

Text data with a high dimensionality is a difficult algorithmic problem for machine learning. Therefore, the focus in this paper was to rank the features and decrease the number of unnecessary and duplicated features to improve performance of the classifier. The main contribution of the proposed ARDC technique is to modify the true positive and false positive rates for terms counts in the positive and negative classes to ensure a high rank for frequently occurring terms count in positive class. The ARDC technique examines both the term count and document frequency in order to rate the ranking of the term and increases the weight of the frequent terms count in a positive class by dividing the false positive rate by the number of categories in the negative class. The experiments demonstrated that among term count information improved feature ranking algorithms, ARDC produces better precision, recall, f-measure, and accuracy values in the majority of classification cases, thus it is an effective technique for feature ranking. The performance of ARDC was compared with that of the existing IG, PCC, ReliefF, RDC and IRDC algorithms

by applying them on three datasets (Reuters21578, 20newsgroups, and TDT2), using MNB, SVM, MLP, KNN and DT classifiers. The results revealed that the ARDC algorithm achieved the highest performance in almost all cases. As well as, the experiment validated the proposed ARDC and demonstrated its effectiveness by achieving higher accuracy in RF and LR classifiers compared to RPFR, indicating the general superiority of ARDC in terms of accuracy, with the exception of the NB classifier. As a future work, the evaluation of the efficiency of ARDC on a variety of other non-text datasets can be carried out and uses the contemporary classifiers like BERT or XLNet for text classification. In addition, ARDC could be integrated with other techniques, such as relevant-based feature ranking and meta-heuristic techniques, to leverage the strengths of each technique and produce a more robust set of features for the model. This can potentially lead to improved model performance, as the model is better able to focus on the most relevant features.

REFERENCES

[1] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2015, pp. 1200–1205.

[2] A. Adeleke, N. A. Samsudin, Z. A. Othman, and S. K. A. Khalid, "A two-step feature selection method for quranic text classification," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 730–736, 2019.

[3] M. Labani, P. Moradi, F. Ahmadizar, and M. Jalili, "A novel multivariate filter method for feature selection in text classification problems," *Eng. Appl. Artif. Intell.*, vol. 70, pp. 25–37, Apr. 2018.

[4] A. Faraz, "An elaboration of text categorization and automatic text classification through mathematical and graphical modelling," *Comput. Sci. Eng., Int. J.*, vol. 5, nos. 2–3, pp. 1–11, Jun. 2015.

[5] P. Pundir, V. Gomanse, and N. Krishnamacharya, "Classification and prediction techniques using machine learning for anomaly detection," *J. Eng. Res. Appl.*, vol. 1, no. 4, pp. 1716–1722, 2011.

[6] S. Z. Mishu and S. M. Rafiuddin, "Performance analysis of supervised machine learning algorithms for text classification," in *Proc. 19th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2016, pp. 409–413.

[7] W. Sharif, N. A. Samsudin, M. M. Deris, and M. Aamir, "Improved relative discriminative criterion feature ranking technique for text classification," *Int. J. Artif. Intell.*, vol. 15, no. 2, pp. 61–78, 2017.

[8] A. Rehman, K. Javed, H. A. Babri, and M. Saeed, "Relative discrimination criterion—A novel feature ranking method for text data," *Exp. Syst. Appl.*, vol. 42, no. 7, pp. 3670–3681, May 2015.

[9] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[10] S. Venkataraman and R. Selvaraj, "Optimal and novel hybrid feature selection framework for effective data classification," in *Lecture Notes in Electrical Engineering*. Singapore: Springer, 2018, pp. 499–514.

[11] A. K. Uysal, "On two-stage feature selection methods for text classification," *IEEE Access*, vol. 6, pp. 43233–43251, 2018.

[12] S. Lefkovits and L. Lefkovits, "Gabor feature selection based on information gain," *Proc. Eng.*, vol. 181, pp. 892–898, 2017.

[13] A. Chinnaswamy and R. Srinivasan, "Hybrid feature selection using correlation coefficient and particle swarm optimization on microarray gene expression data," in *Advances in Intelligent Systems and Computing*, vol. 424. Cham, Switzerland: Springer, 2016, pp. 139–149.

[14] Z. Wang, Y. Zhang, Z. Chen, H. Yang, Y. Sun, J. Kang, Y. Yang, and X. Liang, "Application of ReliefF algorithm to selecting feature sets for classification of high resolution remote sensing image," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 755–758.

[15] M. Arabnejad, B. A. Dawkins, W. S. Bush, B. C. White, A. R. Harkness, and B. A. McKinney, "Transition-transversion encoding and genetic relationship metric in ReliefF feature selection improves pathway enrichment in GWAS," *BioData Mining*, vol. 11, no. 1, pp. 1–17, Dec. 2018.

[16] V. D. P. Jasti, "Relevant-based feature ranking (RBFR) method for text classification based on machine learning algorithm," *J. Nanomaterials*, vol. 2022, Aug. 2022, Art. no. 9238968.

[17] D. Ö. Şahin and E. Kılıç, "Two new feature selection metrics for text classification," *Automatika*, vol. 60, no. 2, pp. 162–171, Apr. 2019.

[18] I. Perikos and I. Hatzilygeroudis, "Recognizing emotions in text using ensemble of classifiers," *Eng. Appl. Artif. Intell.*, vol. 51, pp. 191–201, May 2016.

[19] D. Badawi and H. Altinçay, "A novel framework for termset selection and weighting in binary text classification," *Eng. Appl. Artif. Intell.*, vol. 35, pp. 38–53, Oct. 2014.

[20] Z. Erenel and H. Altinçay, "Nonlinear transformation of term frequencies for term weighting in text categorization," *Eng. Appl. Artif. Intell.*, vol. 25, pp. 1505–1514, Oct. 2012.

[21] M. Hemmati and S. Mousavirad, *A New Hybrid Method for Text Feature Selection Through Combination of Relative Discrimination Criterion and Ant Colony Optimization*. Cham, Switzerland: Springer, Accessed: May 2, 2023.

[22] L. Jin and L. Zhang, "De-redundancy relative discrimination criterion-based feature selection for text data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2023, pp. 1–8.

[23] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved chi-square for Arabic text classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 32, no. 2, pp. 225–231, Feb. 2020.

[24] F. Wang, Y. Zhang, H. Xiao, L. Kuang, and Y. Lai, "Enhancing stock price prediction with a hybrid approach based extreme learning machine," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 1568–1575.

[25] H. Banka, H. Mahmood, K. Fatih, S. Mehmet, and V. Üniversitesi, *A Hybrid Feature Selection Approach Based on LSI for Classification of Urdu Text*, vol. 907. Cham, Switzerland: Springer, 2021, pp. 3–18.

[26] J. Piri, P. Mohapatra, M. R. Pradhan, B. Acharya, and T. K. Patra, "A binary multi-objective chimp optimizer with dual archive for feature selection in the healthcare domain," *IEEE Access*, vol. 10, pp. 1756–1774, 2022.

[27] J. Piri, P. Mohapatra, B. Acharya, F. S. Gharehchopogh, V. C. Gerogiannis, A. Kanavos, and S. Manika, "Feature selection using artificial gorilla troop optimization for biomedical data: A case analysis with COVID-19 data," *Mathematics*, vol. 10, no. 15, p. 2742, Aug. 2022.

[28] J. Piri, P. Mohapatra, H. K. R. Singh, B. Acharya, and T. K. Patra, "An enhanced binary multiobjective hybrid filter-wrapper chimp optimization based feature selection method for COVID-19 patient health prediction," *IEEE Access*, vol. 10, pp. 100376–100396, 2022.

[29] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, Mar. 1980.

[30] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowl.-Based Syst.*, vol. 36, pp. 226–235, Dec. 2012.

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[32] J. Nayak, B. Naik, and H. S. Behera, "A comprehensive survey on support vector machine in data mining tasks: Applications & challenges," *Int. J. Database Theory Appl.*, vol. 8, no. 1, pp. 169–186, Feb. 2015.

[33] A. Kumar and A. Jaiswal, "Swarm intelligence based optimal feature selection for enhanced predictive sentiment accuracy on Twitter," *Multimedia Tools Appl.*, vol. 78, no. 20, pp. 29529–29553, Oct. 2019.

[34] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12201–12220, Aug. 2020.

[35] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

SARAH ABDULKAREM ALSHALIF received the bachelor's degree in computer science and information technology and the master's degree in information technology from Universiti Tun Hussein Onn Malaysia (UTHM), in 2014 and 2018, respectively, where she is currently pursuing the Ph.D. degree in information technology with the Faculty of Computer Science and Information Technology. Her research interests include soft computing, text classification, and feature selection.

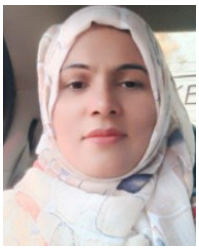
NORHALINA SENAN is currently a Senior Lecturer with Universiti Tun Hussein Onn Malaysia. She was with the Multimedia Department, Faculty of Computer Science and Information Technology, for 21 years. Currently, she is the Head of the Postgraduate Department, Faculty of Computer Science and Information Technology. Her research interests include soft computing, data mining, sound and image processing, augmented and virtual reality, user experience, and persuasive interface design.

FAISAL SAEED received the B.Sc. degree in computers (information technology) from Cairo University, Egypt, the M.Sc. degree in information technology management, and the Ph.D. degree in computer science from Universiti Teknologi Malaysia (UTM), Malaysia. He is currently a Senior Lecturer with the Computing and Data Science Department, School of Computing and Digital Technology, Birmingham City University (BCU), U.K., where he is also leading the Smart Health Laboratory, Data Analytics and AI Research Group. Previously, he was an Assistant/Associate Professor with Taibah University, Saudi Arabia, from 2017 to 2021, and a Senior Lecturer with the Department of Information Systems, Faculty of Computing, UTM, from 2014 to 2017. He has published several papers in indexed journals and international conferences. He served as the general chair for several international conferences and a guest editor for several indexed journals. His research interests include data mining, artificial intelligence, machine learning, information retrieval, and health informatics.

WAD GHABAN received the B.Sc. degree (Hons.) in computer science from King Abdul-Aziz University, Jeddah, and the M.Sc. degree (Hons.) in advanced computer science and the Ph.D. degree from the University of Birmingham, in 2015 and 2020, respectively. She is currently an Assistant Professor with the Applied College, University of Tabuk, Saudi Arabia. During her study, she worked on several projects related to human–computer interaction, survival analysis, online learning, natural language processing, and sentiment analysis. She also published a number of papers that are published and presented in several international conferences and indexed journals. Her research interests include human–computer interaction, machine learning, sentiment analysis, and data analysis.

NORAINI IBRAHIM received the Bachelor of Science (B.Sc.) degree in information technology from Universiti Kebangsaan Malaysia (UKM), in 2000, the Master of Science (M.Sc.) degree in real-time software engineering from Universiti Teknologi Malaysia (UTM), in 2003, and the Ph.D. degree in information technology focusing on software engineering related research—consistency checking for UML diagrams using logical approach from Universiti Tun Hussein Onn Malaysia (UTHM), in 2013. She is currently an active researcher in the area of software requirement specification, software testing, and software project management.

MUHAMMAD AAMIR received the master's degree in computer science from the City University of Science and Information Technology, Pakistan, and the Ph.D. degree in information technology from University Tun Hussien Onn Malaysia. He has been a Research Data Scientist and a Machine Learning Development Engineer with the University of Derby, U.K., since October 2020. He had worked for two years with Xululabs LLC as a Data Scientist. His research interests include data science, deep learning, and computer programming.



WAREESA SHARIF received the master's degree in computer science from The Islamia University Bahawalpur, Pakistan, and the Ph.D. degree from the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia, in 2020. She was a Research Assistant with UTHM, from 2015 to 2019. She is currently an Assistant Professor with the Department of Artificial Intelligence, The Islamia University of Bahawalpur, where she is also a member of Admission Committee, Department of Artificial Intelligence, Faculty of Computing. Her research interests include probabilistic modeling, machine learning, data mining, feature selection, text classification, optimization, sentiment analysis, and recommender system and data science (OSA).