Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK

A New Dynamic Classifier Selection Method For Text Classification

Ismail Terzi [1], **Alper Kursat Uysal**[2] *
[1]Department of Computer Engineering, Engineering Faculty, Zonguldak Bulent Ecevit University, Zonguldak, Turkiye, ORCID iD: https://orcid.org/0000-0002-9237-0732
[2]Department of Computer Engineering, Engineering Faculty, Alanya Alaaddin Keykubat University, Antalya, Turkiye, ORCID iD: https://orcid.org/0000-0002-4057-934X

Received: .202 • **Accepted/Published Online:** .202 • **Final Version:** ..202

Abstract: The primary objective of employing Multiple Classifier Systems (MCS) in pattern recognition is to enhance classification accuracy. Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES) are two purposeful forms of multiple classifier systems. While DES involves the selection of a classifier set followed by decision combination, DCS opts for the choice of a single competent classifier, eliminating the necessity for classifier combination. As a consequence, DCS methods exhibit superior efficiency in terms of processing time and memory usage compared to DES methods. Moreover, a substantial performance gap exists between the performance of Oracle and both DES and DCS methods. In this study, we introduce a novel method termed DCS-DQ (Dynamic Classifier Selection Technique- Decision Quotient) for text classification based on dynamic classifier selection. Our experimental investigation encompasses four distinct text datasets, with classification accuracy and Macro F-1 score serving as the primary evaluation criteria. The proposed DCS-DQ method is subjected to comparison with seven state-of-the-art DCS methods. Based on our empirical findings, the DCS-DQ method outperforms the other seven DCS methods in terms of classification accuracy across the majority of feature sizes. Notably, in the Reuters dataset, the classification accuracy of DCS-DQ surpasses that of other DCS methods for all feature sizes except when the feature size is 100. Similarly, in the Ohsumed dataset, the DCS-DQ method demonstrates significant performance improvement, with an accuracy value of 77.02% for 3000 features compared to the maximum accuracy value of 72.74% achieved by the DCS method MCB. Additionally, the performance of the proposed DCS-DQ method closely aligns with the oracle performance compared to the other methods. In conclusion, our proposed DCS-DQ method holds promise for significantly improving classification accuracy in text classification literature.

Key words: Text Classification, Dynamic Classifier Selection, Multiple Classifier Systems, DCS-DQ.

1. Introduction

The volume of text documents in the databases of companies and on the Internet is increasing day by day. Consequently, there has been a growing inclination towards the field of text classification. E-mail messages, articles on web pages, research articles, tweets, medical reports, customer correspondence, blogs, customer reviews on shopping sites are composed of text messages. People not only save documents but also discover some useful patterns within them. Since such an amount of text data is overwhelming to analyze for individuals. People need useful tools to deal with such number of text documents. Classification of text documents is the process of determining classes for text documents based on their content. The foremost objective within the process of text classification is assigning the text document to the appropriate class. Numerous machine

*Correspondence: alper.uysal@alanya.edu.tr learning methods have been employed in the field of text classification so far, including; naïve Bayes(NB), Logistic Regression(LR), Support Vector Machines (SVM), Random Forest(RF), K-Nearest Neighbor (KNN), Decision Tree(DT) Classifiers, and Rocchio Algorithm(RA) [1]. Text classification techniques are utilized in many applications to simplify people's lives. Document categorization [2], document routing application [3], author recognition[4], opinion mining and sentiment analysis [5, 6], question-answering systems [7] and detection of spam SMS messages and social spam detection [8] were performed due to the capability of the text classification techniques. Ensembles of classifiers represent a widely discussed area in the domain of machine learning. According to Dietterich [9], ensembles of classifiers are the leading research direction in machine learning and they can improve the accuracy of classification. In literature, authors refer to the ensembles of classifiers as multiple classifier systems[10] frequently. In this work, we use Multiple Classifier Systems (MCS). MCS have better performance than traditional single classifier systems [9, 10]. MCS, as illustrated in the Figure 1, consists of three sequential components: the initial phase entails generation, followed by selection, and culminating in integration. The selection component is categorized into two distinct groups: Dynamic Selection (DS) and Static Selection (SS). In DS, there are two approaches, Dynamic Ensemble Selection (DES) and Dynamic Classifier Selection (DCS). SS methods employ a singular classifier or an ensemble of classifiers during the training phase, subsequently using the identical selected classifiers to predict outcomes for all unknown samples. DS methods select only a single classifier or a combination of classifiers for every unknown samples. As DCS methods employ a single classifier, the requirement for an integration phase is obviated. The absence of a requirement for integration renders DCS methods more efficient than DES methods in processing time and memory usage.

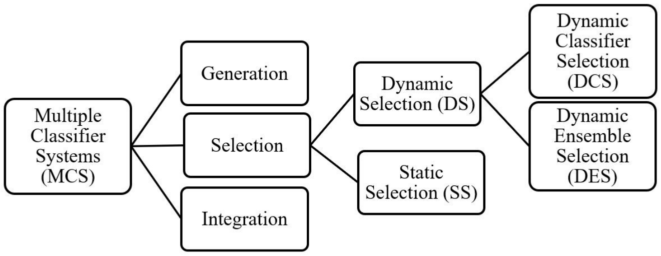
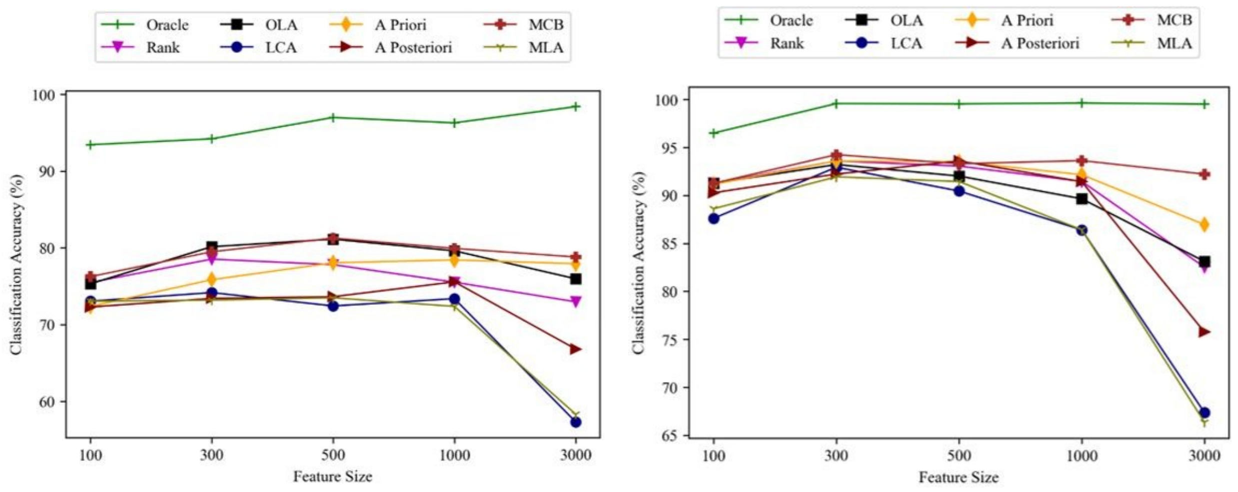

Multiple Classifier Systems (MCS)
Generation
Selection
Integration
Dynamic Selection (DS)
Static Selection (SS)
Dynamic Classifier Selection (DCS)
Dynamic Ensemble Selection (DES)

Figure 1: Multiple classifier systems.

Oracle [11] stands as another significant concept within the domain of DCS. Oracle is an abstract method used to identify the classifier that can accurately classify the text instance among the available classifiers, given the existence of such a classifier. The key aspect of Oracle is that at least one of the classifiers in the pool should be capable of correctly classifying the unknown sample. The performance of Oracle serves as the upper bound for DCS methods [12] and there is a substantial performance gap between Oracle performance and DCS methods. In this study, the proposed DCS-DQ method can contribute to closing these gaps. Performance gap between the existing methods and oracle performance is shown in Figure 2 on two different datasets. When the number of features is 3000, the classification accuracy of the DCS method is 80%. However, Oracle performance is 97% with the same number of features. The discrepancy is significant. Studies aimed at addressing this gap will yield significant contributions to the DCS literature. In all datasets, a substantial performance gap is observable between existing methods and the performance exhibited by the Oracle. In the following sections, we will demonstrate how our proposed method, namely DCS-DQ, effectively narrows this significant gap.



(a) Polarity dataset

(b) Enron1 dataset

Figure 2: Oracle performance for Polarity and Enron1 dataset

The main motivation of this work is to;

1. Propose a new DCS method namely DCS-DQ for text classification
2. Make contribution for reducing the gap between DCS and Oracle performance
3. Show that the proposed DCS-DQ outperforms current state-of-the-art techniques
4. Analyze the effectiveness of proposed DCS-DQ method on text datasets with different characteristics

The organization of this paper is as follows; works related to applications using multiple classification systems are explained in section 2, multiple classifier systems are briefly explained in section 3, existing DCS methods in literature is given in section 4, the proposed method DCS-DQ is presented in section 5, experimental studies are presented in section 6, the experimental findings are given in section 7 and conclusion about our work is presented in section section 8.

2. Related Works

In literature, a significant number of articles utilizing DCS methods have been published recently. These methods are being applied to a wide range of real-world problems. Credit scoring [21], face recognition systems, [22], biometric verification [23], signature verification [24] and customer classification [25] are applications of DCS methods. Wen et al. utilized dynamic classifier selection techniques to identify the ball screw degradation[26]. Groccia, Guido and Conforti [27] introduced a framework that integrates several classification algorithms by dynamically selecting the most proficient classifier. In literature, many classification datasets are unbalanced and proposing techniques for unbalanced data is more valuable. Roy et al. [28] have experimentally shown that dynamic selection methods have the potential to achieve superior performance compared to static ensembles in unbalanced classification problems. In experimental studies, they used DS methods, LCA and Rank. Today, with the rise in the use of Android smartphones, malicious applications that threaten the Android platform has also increased. Feng et al.[29] proposed an ensemble-based Android malware detection method called EnDroid to protect Android platforms from malware. Various machine learning algorithms were employed, including

NB, KNN, SVM, DT, Boosted Tree and RF. Credit scoring is another critical issue for financial institutions. Junior et al.[30] investigated the suitability of dynamic selection techniques for credit scoring and introduced the Reduced Minority k-Nearest Neighbors (RMkNN) method. Their proposed approach improves the delineation of local regions in dynamic selection techniques for imbalanced credit scoring datasets. Another ensemble based study on credit scoring was published by Feng et al. [31]. In their study, a dynamically weighted ensemble method is proposed for credit scoring. They used a Markov Chain to dynamically weight the classifiers in the classifier pool for each sample in the test set and combine the classifiers' decisions. Martins et al. [32] published a research on forest species recognition. In this research, they used dynamic classifier selection methods such as MCB, OLA, LCA, A Priory and A Posteriory. The best result in this work is (93.03%). The best result is observed when integrating probabilistic information into a dynamic classifier selection method based on MCB. DS techniques are used in time series forecasting Sergio et al.[35] proposed a dynamic selection of regressors for time series forecasting. The authors developed an algorithm inspired by the dynamic classifier selection method MCB to predict the competence of each of the combiner. The technique, termed Dynamic Selection of Forecast Combiners (DS-FC), is a heuristic approach designed to choose an optimal ensemble from a provided pool of classifiers [36]. The proposed algorithm is a pruning algorithm based on accuracy and diversity. It evaluates both the accuracy of individual classifiers and the pairwise diversity among them. Cruz et al. [37] showed that DCS methods offer a substantial increase in classification accuracy compared to K-NN. Nwulu, Twala and Aigbavboa [38] used dynamic selection techniques, including Rank, LCA, OLA to address the issue of Water Quality Anomaly Detection problem. Groccia et al. use dynamic classifier selection techniques for clinical diagnosis [39]. Text data is frequently used in the medical field, and sorting medical texts into clinical texts, clinical notes, prescriptions and examination requests is an important task. Magalhães et al.[33] analyzed the success of classifier ensemble approaches in classifying medical texts. In their analysis, they obtained results that can automatically and accurately classify clinical texts with higher accuracy than individual approaches. There are also studies that use dynamic selection methods in feature selection. Li et al.[34] proposed a dynamic feature selection method for extracting semantic features from agricultural texts.

3. Multiple Classifier Systems (MCS)

Throughout this paper the following notations are used.

- $D = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_m, y_m)\}$ is a text dataset containing m documents and $(x_1, x_2, x_3, ..., x_m)$ are documents, $(y_1, y_2, y_3, ..., y_m)$ are corresponding class labels of $(x_1, x_2, x_3, ..., x_m)$.

- $\Omega = (\omega_1, \omega_2, \omega_3, ..., \omega_n)$ are the corresponding class labels and $y_m \in \Omega$.

- x_i is a sample text document with unknown class label in test set.

- Ď is the Dynamic Selection Data (DSEL).

- $P = (C_1, C_2, C_3 \ldots C_T)$ is the pool composed number of T base classifiers and C_i is the most competent classifier for x_i selected by DCS method.

- $\delta = (\partial_{C_1}, \partial_{C_2}, \partial_{C_3}, \ldots, \partial_{C_T})$ is the set of accuracy value of the base classifiers on Ď and ∂_{C_t} is the accuracy value of a base classifier C_t on Ď.

- $\theta_{x_i} = (\dot{x}_1, \dot{x}_2, \dot{x}_3..\dot{x}_k)$ is the k nearest neighbor of x_i in Ď and $\dot{x}_k$ is a neighbor of x_i in θ_{x_i}.

- $\emptyset_{t_{x_i}}$; competence level of a base classifier C_t for x_i .
- $\Psi_{x_i} = \{\sigma_{c_1}, \sigma_{c_2}, \sigma_{c_3}, \ldots, \sigma_{c_T}\}$ is the set of accuracy value of the base classifiers on θ_{x_i} and σ_{c_t} is the accuracy value of a base classifier C_t on θ_{x_i} .
- $W_i = \frac{1}{d_i}$; d_i is distance between x_i and $\dot{x}_i$.
- $\tilde{x}_i$; output profile of x_i

MCS systems encompass three primary stages ; the first step is creation of a classifier pool P, the second step is selection of a classifier C_i or a subset of classifiers $C^{'}$, and the last step is the combination of the classification results of various classifiers [13]. DCS methods do not have combination step. The absence of a combination step can be considered as one of the strengths of DCS methods. The pool of classifiers, denoted as P, can be generated in six distinct methods. [20]. These involve different initialization, different feature sets, different parameters, different classifier models, different architectures, different training sets. Additionally, Bagging and Boosting methods can be utilized for creating the pool. [14]. Different subset of training set is used in Bagging[15] method. After creation of pool of classifiers, selection process is performed. SS methods select one competent classifier or a set of classifiers which is also named as ensemble of classifiers at the training stage and anticipate all unknown samples x_i by using the same classifier or a set of classifiers. The most competent classifier for x_i is the classifier that classifies all the samples in θ_{x_i} with the highest accuracy. In DCS strategy base classifier C_i is selected on the fly. Different base classifiers are selected for each x_i . In order to yield more accurate results for the multiple classifier system, each classifier constituting the pool must be accurate and diverse [16]. A classifier can be considered accurate if it yields an error rate lower than that of random guessing for unknown test samples. x_i . If two classifiers exhibit different errors on the same test sample x_i , then these two classifiers can be deemed diverse [17].

The key idea of using DCS technique is that, competence of each base classifier must be determined. The performance of DCS methods is very dependent on the detection of this competence[18]. The rationale behind this explanation is that each base classifier specializes in a distinct region of the feature space. [13]. Determining θ_{x_i} for a given sample x_i is the fundamental concept. Once the region is determined, the classifier C_i that has the highest classification accuracy on θ_{x_i} is chosen. K-Nearest Neighbors technique is mostly used to determine this region[19]. Given a test instance x_i , DCS methods select the most capable classifier C_i . C_i assigns the instance x_i to a class ω_l . The diagram depicting the operation of the DCS method is presented. in Figure 3 below.

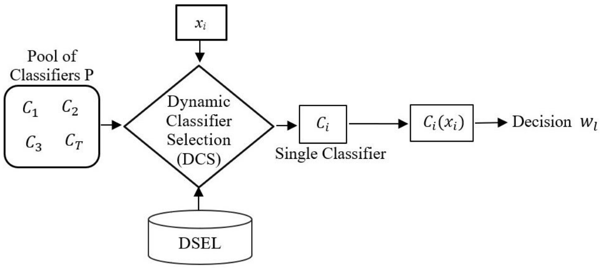


Figure 3: Dynamic classifier selection (DCS).

As shown in Figure 3, vital task of DCS methods is to reveal the most capable classifier C_i for each x_i. DCS methods use a labeled validation or dynamic selection data (DSEL)[20] when deciding which classifier to be chosen. The DSEL data is initially segregated from the dataset and does not overlap with the test and train data.

4. DCS Methods

In this study, proposed DCS method has been compared with the most commonly used DCS methods in the literature. In this section DCS methods will be briefly explained. In our experimental study, these methods were employed with their default parameters, as described in [44].

4.1. Modified Classifier Rank (DCS-Rank)[40]:

In this approach, the ranking of a base classifier C_t is determined by the consecutive correct classifications within the neighborhood θ_{x_i}. ”Consecutive” in this context refers to classifications from the nearest neighbor to the farthest one. A classifier C_t that correctly classifies the greatest number of consecutive nearest samples is assigned the highest ”rank”. Consequently, the classifier with the highest rank is selected, and x_i is classified accordingly.

4.2. Overall Local Accuracy (OLA) [40]:

In this method, the classifier C_t that has the highest classification accuracy on θ_{x_i} is the most competent classifier, so x_i is classified by C_t. Classifier competency $\emptyset_{t_{x_i}}$ is calculated by equation (Equation 1).

$$\emptyset_{t_{x_i}} = \frac{1}{k}\sum_{i=1}^{k} P(\omega_l|\dot{x}_i \in \omega_l, C_t) \tag{1}$$

4.3. Local Classifier Accuracy (LCA) [40]:

In this method, first of all the θ_{x_i} of the test sample x_i is formed. Following this step, the proficiency of the base classifier C_t is determined based on its classification accuracy, focusing solely on the samples from the class ω_l within this neighborhood. Here, ω_l represents the class predicted by the base classifier C_t, for x_i. The competency level $\emptyset_{t_{x_i}}$ is assessed using (Equation 2). If multiple base classifiers achieve identical competency levels, the first one encountered is chosen.

$$\emptyset_{t_{x_i}} = \frac{\sum_{\dot{x}_i \in \omega_l} P(\omega_l|\dot{x}_i, C_t)}{\sum_{i=1}^{k} P(\omega_l|\dot{x}_i, C_t)} \tag{2}$$

4.4. A Priory [41]:

$\emptyset_{t_{x_i}}$ is predicted based on the probability of true classification of the base classifier C_t, taking into account all samples in θ_{x_i}. This method weights the impact of each training sample as per its distance W_i to x_i. Competence level of a classifier $\emptyset_{t_{x_i}}$ is calculated by equation (Equation 3). If multiple base classifiers achieve identical competency levels, the first one encountered is chosen.

$$\emptyset_{t_{x_i}} = \frac{\sum_{i=1}^{k} P(\omega_l|\dot{x}_i \in \omega_l, C_t)W_i}{\sum_{i=1}^{k} W_i} \tag{3}$$

4.5. Multiple Classifier Behavior (MCB) [41]:

The region of competence is estimated considering the feature space and the decision space (using the Behavior Knowledge Space(BKS) method [52]). First, θ_{x_i} is formed for x_i. Then, the similarity in the BKS space between x_i and θ_{x_i} are estimated using the (Equation 4),

$$S(\tilde{x}_i, \tilde{\dot{x}}_i) = \frac{1}{M}\sum_{i=1}^{M} T(x_i, \dot{x}); \qquad T(x_i, \dot{x}) = \begin{cases} 1 & if \;\; C_t(x_i) = C_t(\dot{x}_i) \\ 0 & if \;\; C_t(x_i) \neq C_t(\dot{x}_i) \end{cases} \tag{4}$$

Where $S(\tilde{x}_i, \tilde{\dot{x}}_i)$ denotes the similarity between x_i and $\dot{x}_i$ according to the behavior knowledge space method (BKS). M represents the number of base classifiers in the classifier Pool. Instances with a similarity below a predetermined threshold are excluded from the θ_{x_i}. The competence level of the base classifiers $\emptyset_{t_{x_i}}$ is predicted based on their classification accuracy within the last region of competence θ_{x_i}.

4.6. Modified Local Accuracy (MLA)[42]:

The competence level $\emptyset_{t_{x_i}}$ of C_t is determined according to its classification accuracy, considering solely the samples associated with a specific class ω_l. Here, ω_l denotes the class predicted by the base classifier C_t, for x_i. This approach evaluates the significance of each training sample based on its proximity to the query instance. The competence level of a classifier $\emptyset_{t_{x_i}}$ is computed using (Equation 5).

$$\emptyset_{t_{x_i}} = \sum_{i=1}^{k} P(\omega_l|\dot{x}_i \in \omega_l, C_t)W_i \tag{5}$$

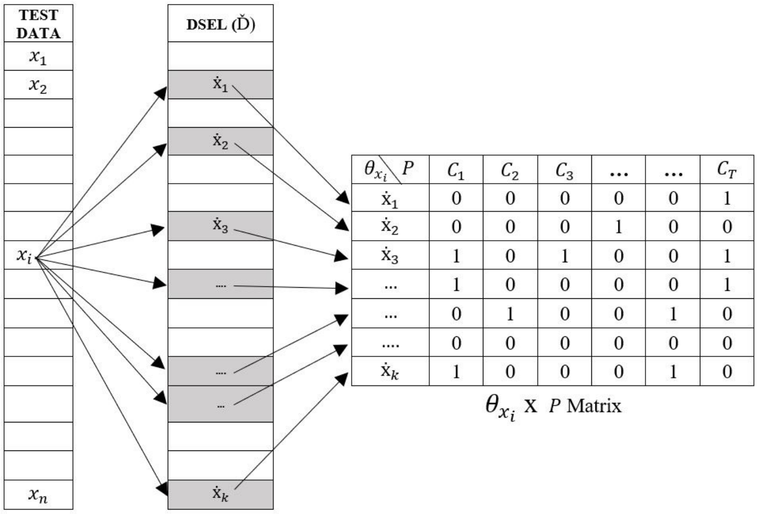
5. Proposed Method

In this study, DCS methods that are prominent in terms of their popularity and number of citations have been selected. The main shortcomings of the existing DCS methods are the way of deciding the competence level of a base classifiers. Existing DCS methods decide competence level of the base classifiers by using k-nearest neighbors of unknown sample x_i in Ď. A base classifier possessing the highest level of competence within the k-nearest neighbors of an unknown sample in Ď is selected and the selected classifier classifies x_i. Most of the time it is challenging to find *k-nearest neighbors* with a high degree of similarity for x_i, since datasets are very sparse[43]. Deciding the competence level of the base classifiers using k-unlike neighbors is often inaccurate. The most important limitation of existing methods is that an example to be classified is classified by one of the classifiers selected from the pool even if it has no similar neighbors.

The proposed method has been developed in order to eliminate the mentioned shortcomings of the existing methods. The basic principle of the proposed method is the k Nearest Neighbor algorithm (k-NN), which is a widely recognized machine learning technique. The basic philosophy underlying the k-NN algorithm is that, any test instance is similar to the nearest instance whose label is known in training set. With the same inference;

- $Claim1$: Any test instance x_i is similar to the nearest samples in DSEL(Ď) data. This similarity can also be used to infer that the test instance is easily classifiable or hardly classifiable.
- $Claim2$: If every classifier in the classifier pool can classify the nearest neighbors of x_i with high accuracy, then x_i is said to be easily classified, conversely, if the nearest neighbors of the test sample x_i are hard to classify, then x_i will also hard to be classified.

Figure 4 below presents the proposed DCS-DQ method, where $(x_1, x_2, x_3, ..., x_n)$ are test documents in Test Data. θ_{x_i} =($\dot{\mathrm{x}}_1, \dot{\mathrm{x}}_2, \dot{\mathrm{x}}_3, ..., \dot{\mathrm{x}}_k$) are the k nearest neighbors for x_i in DSEL(Ď), P is the classifier pool. Most important part of the proposed DCS-DQ method is to construct a (θ_{x_i} x P) Matrix.



θ_{x_i} \ P	C_1	C_2	C_3	...	...	C_T
$\dot{\mathrm{x}}_1$	0	0	0	0	0	1
$\dot{\mathrm{x}}_2$	0	0	0	1	0	0
$\dot{\mathrm{x}}_3$	1	0	1	0	0	1
...	1	0	0	0	0	1
...	0	1	0	0	1	0
....	0	0	0	0	0	0
$\dot{\mathrm{x}}_k$	1	0	0	0	1	0

Figure 4: Proposed dynamic classifier selection method DCS-DQ.

For each x_i in test data, a (θ_{x_i} x P) matrix is constructed. Rows of (θ_{x_i} x P) matrix represents each neighbor $\dot{\mathrm{x}}_i$ for x_i in Ď. Columns of (θ_{x_i} x P) matrix represents number of T base classifiers. $(C_1, C_2, C_3, ..., C_T)$ in the pool P. Cell of (θ_{x_i} x P) matrix represents decision of C_t for $\dot{\mathrm{x}}_i$. Value of the cell is equal to 1 if $\dot{\mathrm{x}}_i \in w_n$ and $C_t\left(\dot{\mathrm{x}}_i\right) \in w_n$, 0 otherwise. Function $\varphi\left(C_t, \dot{\mathrm{x}}_i\right)$ in (Equation 6) determines the values of the cells in (θ_{x_i} x P) matrix.

$$\varphi\left(C_t, \dot{\mathrm{x}}_i\right) = \begin{cases} 1 \text{ if } \dot{\mathrm{x}}_i \in w_n \text{ and } C_t\left(\dot{\mathrm{x}}_i\right) \in w_n \\ 0 \text{ if } \dot{\mathrm{x}}_i \in w_n \text{ and } C_t\left(\dot{\mathrm{x}}_i\right) \notin w_n \end{cases} \tag{6}$$

In (Equation 7), a function $\lambda\left(\theta_{x_i}\right)$ is defined. $\lambda\left(\theta_{x_i}\right)$ gives summation of the values of all the cells in (θ_{x_i} x P) matrix.

$$\lambda\left(\theta_{x_i}\right) = \sum_{i=1}^{k} \sum_{t=1}^{T} \varphi\left(C_t, \dot{\mathrm{x}}_i\right) \tag{7}$$

Let T be the number of base classifiers in P and the number of nearest neighbors for x_i in Ď is k, then maximum value of the function $\varphi\left(C_t, \dot{\mathrm{x}}_i\right)$ is equal to T*k. This situation is possible if all base classifiers correctly classify all samples in θ_{x_i}. Minimum value of the function $\lambda\left(\theta_{x_i}\right)$ is equal to 0. This situation is possible if none of the base classifiers can correctly classify any of the samples in θ_{x_i}.

Decision Quotient (DQ) of a test instance x_i is calculated by (Equation 8).

$$DQ\left(x_i\right) = \frac{\lambda\left(\theta_{x_i}\right)}{T * k} \tag{8}$$

Range of the value of $DQ\left(x_i\right)$ is between 0 and 1. The value closer to 1 means that x_i is easy to classify since its neighbors are easily classified. The value closer to 0 means that x_i is hard to classify since its neighbors are hardly classified. $DQ\left(x_i\right)$ varies between 0 and 1. It is important to define a threshold for making a decision about a test instance x_i. This decision is about if a test instance x_i is hard to classify or easy to classify. (Equation 9). During experimental study, this threshold can be taken as $\min\left(\partial_{C_1}, \partial_{C_2}, \partial_{C_3}, \dots, \partial_{C_T}\right)$. As stated before, $\delta = \left(\partial_{C_1}, \partial_{C_2}, \partial_{C_3}, \dots, \partial_{C_T}\right)$ are accuracy values of the base classifiers on Ď. Accuracy values of $\left(\partial_{C_1}, \partial_{C_2}, \partial_{C_3}, \dots, \partial_{C_T}\right)$ are also between 0 and 1.

$$x_i = \begin{cases} \text{easy to classifiy} & \text{if DQ}\left(x_i\right) \geq \min(\delta) \\ \text{hard to classifiy} & \text{if DQ}\left(x_i\right) < \min(\delta) \end{cases} \tag{9}$$

$\Psi'_{x_i} = \{\sigma_{c_1}, \sigma_{c_2}, \sigma_{c_3}, \dots, \sigma_{c_T}\}$ are the accuracy values of the base classifiers on θ_{x_i}, $\delta = \left(\partial_{C_1}, \partial_{C_2}, \partial_{C_8}, \dots, \partial_{C_T}\right)$ are accuracy values of the base classifiers on Ď. In (Equation 10), we define $\mathcal{L}G_{x_i} = \Psi_{x_i} + \delta$ which are local and global accuracies of the base classifiers for x_i. $\mathcal{L}$ stands for local and $\mathcal{G}$ stands for global. Local accuracy is the accuracy of the base classifiers on θ_{x_i}, and global accuracy is the accuracy of the base classifiers on Ď

$$\mathcal{L}G_{x_i} = \Psi_{x_i} + \delta = \left(\sigma_{c_1} + \partial_{c_1}, \sigma_{c_2} + \partial_{c_2}, \sigma_{c_3} + \partial_{c_3}, \dots, \sigma_{c_T} + \partial_{c_T}\right) \tag{10}$$

There are two ways of selection of most capable classifiers;

1. If x_i is easy to classify then an arbitrary classifier from P can classify it, but in this situation we select the classifier that classifies DSEL(Ď) with highest accuracy. This classifier is called as single best [13].

2. If x_i is hard to classify then an arbitrary classifier from P can not classify it easily. In this situation, sum the classification accuracies of the base classifiers on Ď and θ_{x_i}, i.e, as stated in (Equation 10) $\max(\mathcal{L}\mathcal{G}_{x_i})$ is calculated and the base classifier C_i, gaining the highest value is selected.

Figure 5 depicts the DCS-DQ method. However, an example scenario for DCS-DQ method is also presented below.

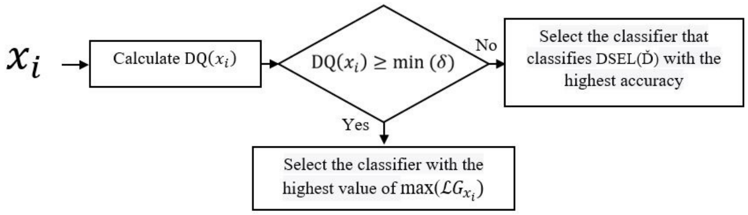
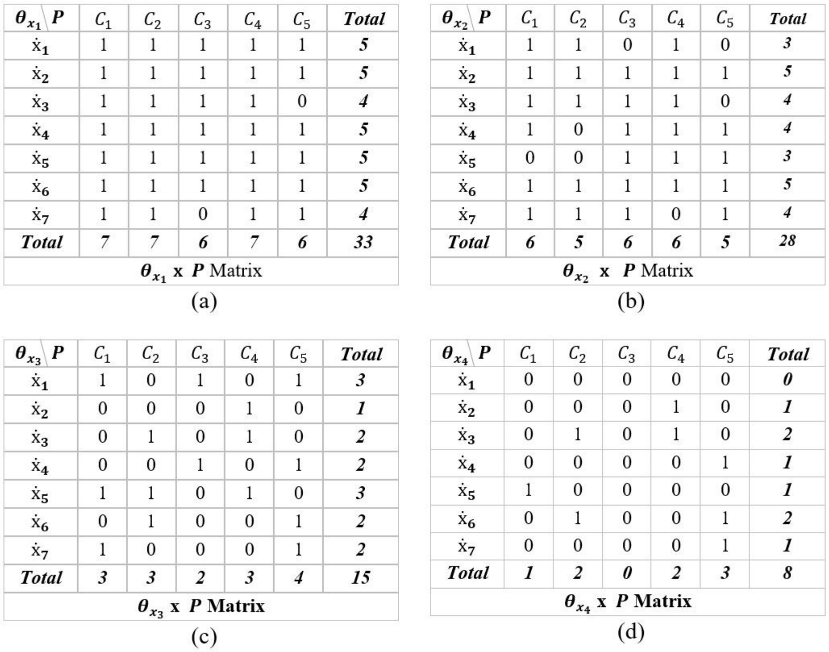

x_i
Calculate DQ(x_i)
DQ(x_i) ≥ min (δ)
No
Select the classifier that classifies DSEL(Ď) with the highest accuracy
Yes
Select the classifier with the highest value of max($\mathcal{L}G_{x_i}$)

Figure 5: DCS-DQ method.

An example for DCS-DQ Method: In this scenario, we have four different test items namely x_1, x_2, x_3, x_4

and five base classifiers namely C_1, C_2, C_3, C_4, C_5 in classifier pool P. We set k=7 and T=5 for all x_1, x_2, x_3, x_4

Step 1: Form (θ_{x_i} x P) matrices for x_1, x_2, x_3, x_4; These matrices are given in Figure 6 (a) (b) (c) (d)



θ_{x_1} \ P	C_1	C_2	C_3	C_4	C_5	*Total*
$\dot{x}_1$	1	1	1	1	1	*5*
$\dot{x}_2$	1	1	1	1	1	*5*
$\dot{x}_3$	1	1	1	1	0	*4*
$\dot{x}_4$	1	1	1	1	1	*5*
$\dot{x}_5$	1	1	1	1	1	*5*
$\dot{x}_6$	1	1	1	1	1	*5*
$\dot{x}_7$	1	1	0	1	1	*4*
Total	*7*	*7*	*6*	*7*	*6*	*33*

θ_{x_1} **x** P Matrix

(a)

θ_{x_2} \ P	C_1	C_2	C_3	C_4	C_5	*Total*
$\dot{x}_1$	1	1	0	1	0	*3*
$\dot{x}_2$	1	1	1	1	1	*5*
$\dot{x}_3$	1	1	1	1	0	*4*
$\dot{x}_4$	1	0	1	1	1	*4*
$\dot{x}_5$	0	0	1	1	1	*3*
$\dot{x}_6$	1	1	1	1	1	*5*
$\dot{x}_7$	1	1	1	0	1	*4*
Total	*6*	*5*	*6*	*6*	*5*	*28*

θ_{x_2} **x** P Matrix

(b)

θ_{x_3} \ P	C_1	C_2	C_3	C_4	C_5	*Total*
$\dot{x}_1$	1	0	1	0	1	*3*
$\dot{x}_2$	0	0	0	1	0	*1*
$\dot{x}_3$	0	1	0	1	0	*2*
$\dot{x}_4$	0	0	1	0	1	*2*
$\dot{x}_5$	1	1	0	1	0	*3*
$\dot{x}_6$	0	1	0	0	1	*2*
$\dot{x}_7$	1	0	0	0	1	*2*
Total	*3*	*3*	*2*	*3*	*4*	*15*

θ_{x_3} **x** P **Matrix**

(c)

θ_{x_4} \ P	C_1	C_2	C_3	C_4	C_5	*Total*
$\dot{x}_1$	0	0	0	0	0	*0*
$\dot{x}_2$	0	0	0	1	0	*1*
$\dot{x}_3$	0	1	0	1	0	*2*
$\dot{x}_4$	0	0	0	0	1	*1*
$\dot{x}_5$	1	0	0	0	0	*1*
$\dot{x}_6$	0	1	0	0	1	*2*
$\dot{x}_7$	0	0	0	0	1	*1*
Total	*1*	*2*	*0*	*2*	*3*	*8*

θ_{x_4} **x** P **Matrix**

(d)

Figure 6: (θ_{x_i} x P) matrices (a) (b) (c) (d).

Step 2: Calculate $\lambda(\boldsymbol{\theta}_{\mathrm{x_i}})$

$$\lambda(\theta_{x_1}) = \sum_{i=1}^{7}\sum_{t=1}^{5}\varphi(C_t, \dot{\mathrm{x}}_i) = 33, \quad \lambda(\theta_{x_2}) = \sum_{i=1}^{7}\sum_{t=1}^{5}\varphi(C_t, \dot{\mathrm{x}}_i) = 28$$

$$\lambda(\theta_{x_3}) = \sum_{i=1}^{7}\sum_{t=1}^{5}\varphi(C_t, \dot{\mathrm{x}}_i) = 15, \quad \lambda(\theta_{x_4}) = \sum_{i=1}^{7}\sum_{t=1}^{5}\varphi(C_t, \dot{\mathrm{x}}_i) = 8$$

Step 3: Calculate DQ (x_i)

$$DQ(x_1) = \frac{33}{35} = 0.94, \quad DQ(x_2) = \frac{28}{35} = 0.80, \quad DQ(x_3) = \frac{15}{35} = 0.43, \quad DQ(x_4) = \frac{8}{35} = 0.22$$

Step 4: Determine if x_i is easy or hard to classify; Suppose the accuracy value of the base classifiers C_1, C_2, C_3, C_4, C_5 is $\delta = (0.75, 0.90, 0.85, 0.95, 0.85)$. In this case, $\min(\delta) = 0.75$. This means that 0.75 is the minimum classification accuracy of the base classifiers on D. As a result x_i with DQ $(x_i) \geq 0.75$ is easy to classify on the contrary, x_i with DQ $(x_i) < 0.75$ is hard to classify. In our example, DQ $(x_1) = 0.94 \geq 75$, DQ$(x_2) = 0.80 \geq 0.75$, DQ $(x_3) = 0.42 < 0.75$ and DQ$(x_4) = 0.22 < 0.75$. Therefore x_1 and x_2 are easy to classify, conversely, x_3 and x_4 are hard to classify. An arbitrary classifier from P can classify x_1 and x_2. We select the classifier that classifies DSEL (D) with highest accuracy. In this situation, $\max(\delta) = 0.95$. So, C_4 is most capable classifier for x_1 and x_2. Consequently DCS-DQ selects the classifier C_4 for x_1 and x_2. Since $DQ(x_3)$ and $DQ(x_4)$ are less than 0.75 so, x_3 and x_4 are hard to classify. Suppose the accuracy value of the base classifiers on θ_{x_3} and θ_{x_4} are $\Psi_{x_3} = \{0.65, 0.85, 0.90, 0.75, 0.95\}$, $\Psi_{x_4} = \{0.70, 0.95, 0.75, 0.85, 0.90\}$

$$\mathcal{LG}_{x_3} = \Psi_{x_3} + \delta = (0.65 + 0.75, 0.85 + 0.90, 0.90 + 0.85, 0.75 + 0.95, 0.95 + 0.85) = (1.40, 1.75, 1.75, 1.70, 1.80)$$

$$\mathcal{LG}_{x_4} = \Psi_{x_4} + \delta = (0.70 + 0.75, 0.95 + 0.90, 0.75 + 0.85, 0.85 + 0.95, 0.90 + 0.85) = (1.45, 1.85, 1.60, 1.80, 1.75)$$

$\max(\mathcal{L}G_{x_3}) = 1.80$ and $\max(\mathcal{L}G_{x_4}) = 1.85$. The max values of $\mathcal{L}G_{x_3}$ and $\mathcal{L}G_{x_4}$ belong to C_5 and C_2, respectively. So, C_5 and C_2 are most capable classifiers for x_3 and x_4, respectively. Consequently, DCS-DQ selects the classifier C_5 from the classifier pool to classify x_3, in the same way DCS-DQ selects the classifier C_2 from the classifier pool to classify x_4.

6. Experimental Study

In this section, the proposed DCS method namely DCS-DQ compared with seven DCS techniques on four different text datasets. By using the same experimental protocol, we compared the DCS-DQ with seven state-of-the-art DCS techniques empirically. Details of the experimental part of this study are expressed in the following subsections. The experiments are carried out using seven DCS techniques given in Section 4. DCS methods are implemented using DESlib library in python [44]. We used default parameter values of all DCS methods in DESlib library. The pseudo-code for the DCS techniques given in the study of Britto et al. [45].

6.1. Classification Scheme

The main research objective of this study is to propose a new DCS method namely DCS-DQ. Experiments were carried out using 4 different benchmark text data sets and 5 state-of-the-art individual diverse base classifiers. The flow of the classification scheme utilized in this study is; document collection, preprocessing, feature extraction, feature weighting, feature selection, classification and analysis of the result. Classification part includes pool generation and classifier selection dynamically. The text classification scheme illustrating the flow of the experiments is pictured in Figure 7. The following section of the paper explains the flow in detail.

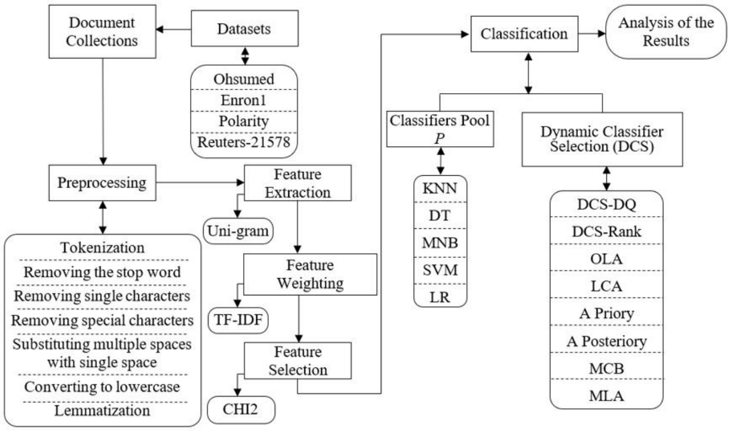


Figure 7: Classification scheme utilized in this study.

6.2. Generating Pool of Classifier

To generate the pool of classifiers, we employed various classifier models. Five state-of-the-art classifiers have been selected. The selected classifiers are commonly employed in the text classification domain due to their high

Table 1: Properties of datasets.

Ohsumed		Reuters-21578- ModApte	
Class Label	# of Documents	Class Label	# of Documents
Neoplasms	2513	earn	3964
Digestive System Diseases	837	acq	2369
Respiratory Tract Diseases	634	money-fx	717
Urologic and Male Genital Diseases	842	grain	582
Nervous System Diseases	1328	crude	578
Cardiovascular diseases	2876	trade	486
Nutritional and Metabolic Diseases	815	interest	478
Immunologic Diseases	1060	ship	286
Disorders of Environmental Origin	1283	wheat	283
Pathological Conditions, Signs and Symptoms	1924	corn	237
Enron1		Polarity	
Legitimate	3672	Positive	1000
Spam	1500	Negative	1000

classification accuracy. Furthermore, these classifiers are heterogeneous, meaning they have diverse structures, thereby establishing diversity. Selected classifiers are; KNN, DT, SVM, LR, and MNB.

6.3. Datasets

Four distinct benchmark datasets, have been utilized in this study. The Reuters-21578 [48] dataset is widely used in text classification studies in the literature. Reuters-21578, known as ModApte split contains the top-10 classes with the highest number of documents. Ohsumed [46] is a multiclass-unbalanced dataset. Ohsumed dataset is generated from a subset of the Medline database. Top ten of its classes are used in the experiments. Enron1 [47] is an e-mail dataset consisting of two unbalanced classes, namely “Legitimate” and “Spam”. Polarity [50] is a two-class balanced dataset including 1000 positive and 1000 negative processed reviews about movies. The total number of documents and class labels related to the datasets used in the experimental studies are presented in Table 1.

6.4. Text Pre-processing Steps

Text pre-processing is essential in text classification due to the presence of noise and redundant words in documents within a corpus. Therefore, pre-processing steps such as tokenization, stop word removal, removal of single characters, removal of special characters, substitution of multiple spaces with a single space, lowercase conversion, and lemmatization have been applied. Classifiers cannot directly operate on raw text data. Initially, text documents must be converted into numerical values. Various methods are employed for this conversion, among which the most popular model is Bag of Words. In this approach, the text dataset is initially transformed into a matrix, where the rows represent documents from the corpus and the columns represent features or words. The numerical values within the cells of the matrix are weighted utilizing Term Frequency-Inverse Document Frequency (TF-IDF) [49] values.

6.5. Feature Extraction and Selection

The total number of features obtained after applying the preprocessing steps described in the previous section is presented in Table 2. The total number of features is relatively high. However, utilizing a large number of features during the classification process can lead to reduced accuracy and classifier performance. Therefore, the Chi-Square (CHI2) [50] feature selection method, which is commonly used, is employed to select the most appropriate features. Feature sizes of 100, 300, 500, 1000, and 3000 are used for all datasets.

Table 2: The numbers of documents and features in datasets.

Dataset	# of Documents	# of Features
Ohsumed	14112	13169
Enron1	5172	9190
Polarity	2000	12638
Reuters- ModApte	9980	9942

6.6. Evaluation

Although 3 out of 4 datasets used are unbalanced, there is not a significant imbalance between the classes. Taking this into consideration, classification accuracy and Macro-averaged F-measure[51] are used as success measures. Macro-F1 measure is particularly suitable for imbalanced data. Macro-F1 is calculated for each class and then averaged across all classes. This ensures that equal weight is given to each class, irrespective of the number of documents in the classes.

Although 3 out of 4 datasets used are unbalanced, there is not a great imbalance between the classes. Considering this imbalance we used classification accuracy and Macro averaged F Measure[51] as success measure. Macro-F1 measure is a suitable success measure for imbalanced data. Macro-F1 is calculated for each class and averaged across all classes. In this manner, each class is assigned equal importance, irrespective of the number of documents within each class. The calculation of Macro-F1 can be formulated as follows:(Equation 11);

$$Macro - F1 = \frac{\sum_{k=1}^{n} F_k}{n}, F_k = \frac{2.p_k.r_k}{p_k.r_k} \tag{11}$$

In (Equation 11); p_k is precision value for class k, $r_k$is the recall value for class k, n; is the number of classes in datasets.

Classification accuracy is the second success measure for this study. It is defined as the proportion of accurately classified samples to the total number of samples to be classified. The equation for classification accuracy is given by (Equation 12);

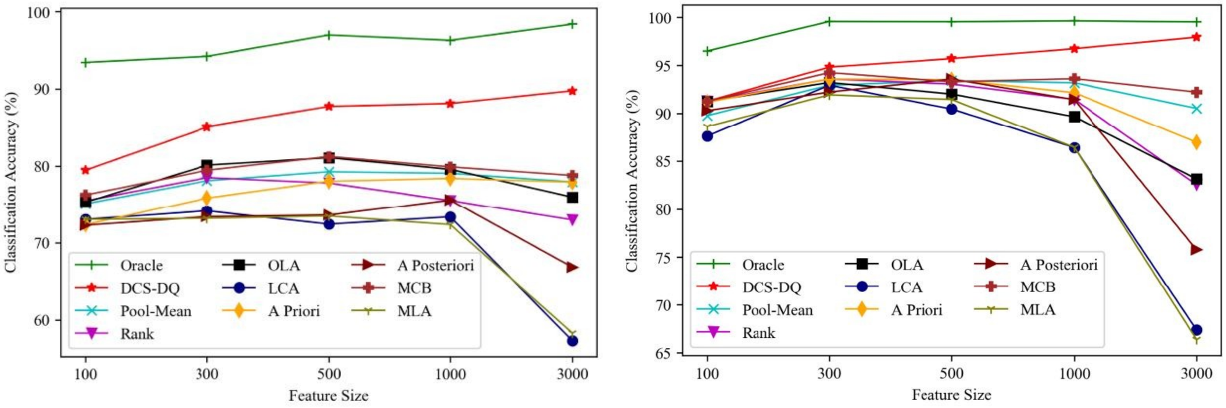
$$\text{Classification Accuracy } = \frac{\text{correctly classified samples}}{\text{total number of samples}} * 100\% \tag{12}$$

7. Experimental Results

After the implementation of the feature selection method, the resulting matrix was randomly split into 50% for training, 25% for testing, and 25% for dynamic selection (DSEL). The base classifiers were trained employing identical training data and tasted using identical testing data. Due to the random partitioning of the test and training data, experiments were conducted 10 times, and the resulting classification accuracies were averaged. In each of the 10 scenarios, training data, testing data, and dynamic selection (DSEL) data were generated randomly. In order to evaluate the performance of DCS-DQ method, experiments were conducted 100, 300, 500, 1000, 3000 feature sizes. Experimental results are presented using line graph. In the graphs, the results of the proposed method DCS-DQ presented in red color. The Oracle scores are shown in green color. In the graphs representing experimental results, DCS methods namely Rank, OLA, LCA, A Priori, A Posteriori, MCB, MLA are shown in different colors. Oracle performance is presented in Figure 8-9. As explained in the introduction, there is a big performance gap between oracle performance and the DCS methods. The proposed DCS-DQ method helps to fill the performance gap between the oracle performance and existing DCS methods in all datasets.

7.1. The results of the experiments conducted on Polarity and Enron 1 dataset

Polarity and Enron1 are two-class datasets. While the Polarity dataset is balanced, the Enron1 dataset is imbalanced. The classification performances of DCS-DQ and existing methods are presented in Figure 8.



(a) Polarity dataset

(b) Enron1 dataset

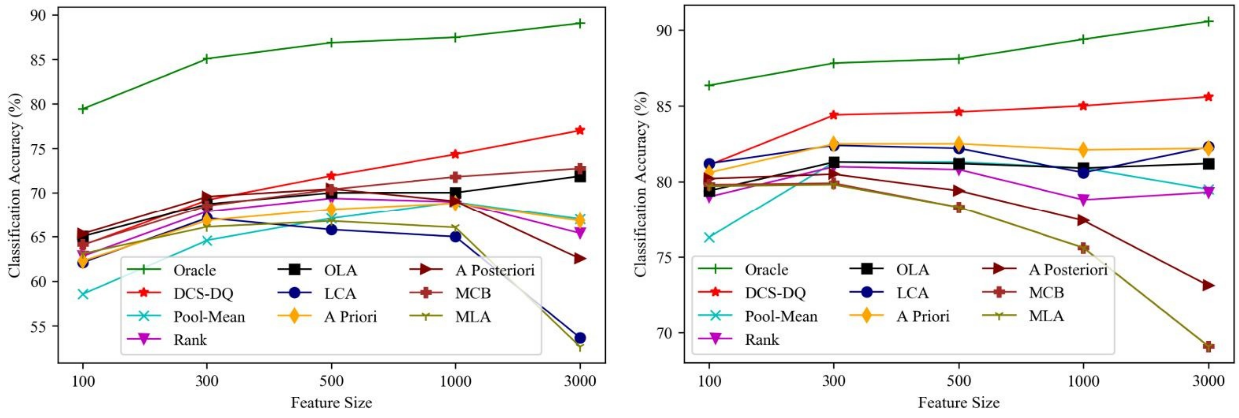
Figure 8: Classification accuracies on Polarity and Enron1 dataset

When analyzing the experimental results in the Polarity dataset, DCS-DQ method outperforms other DCS methods for all feature sizes. Additionally, with an increasing number of features, the performance of the DCS-DQ method improves, whereas the performance of other DCS methods increases up to 500 features and then starts to decline. Upon reviewing the experimental outcomes of the Enron1 dataset, the proposed method demonstrates superior performance compared to the Pool-Mean for all feature sizes. The DCS-DQ method also outperforms other DCS methods for all feature sizes. Moreover, as the feature size increases, the performance of the DCS-DQ method improves, while the performance of other DCS methods decreases. From Figure 8, it can be inferred that the proposed method performs well on both the Enron1 and Polarity datasets.

7.2. The results of the experiments conducted on Ohsumed and Reuters dataset

Classification accuracies for Ohsumed and Reuters dataset are shown in Figure 9.

In accordance with the experimental results of the Ohsumed dataset, for feature size 100, OLA and A Posteriory methods perform well than DCS-DQ. Besides, only A Priory method has better performance than DCS-DQ when the number of feature is 300. For feature sizes which are 500, 1000 and 3000, DCS-DQ method's



(a) Ohsumed dataset

(b) Reuters dataset

Figure 9: Classification accuracies on Ohsumed and Reuters dataset

classification accuracy is much better than others. Upon scrutinizing the experimental results of the Reuters dataset, it is observed that for all feature sizes except for 100, the classifiers selected by the DCS-DQ method yield higher classification accuracy than those selected by other methods. According to the results presented in Figure 9 it is evident that the DCS-DQ method is more efficient than other methods on both the Ohsumed and Reuters datasets. Ohsumed is a collection of medical texts. As depicted in Figure 9-a, the classification accuracy of all DCS methods is low. Based on this outcome, it can be inferred that the Ohsumed dataset poses challenges in classification. Moreover, Ohsumed exhibits the lowest classification accuracy among all datasets for the proposed method.

7.3. The analysis of the Macro F1 scores of the DCS methods and base classifiers

The Macro-F1 scores of the DCS methods and base classifiers in the pool are presented in Table 3 and Table 4. The highest score is highlighted in bold for each corresponding feature size. We have also denoted the highest score with both bold and underlined formatting for clarity. As shown in Table 3 and Table 4, there is no single base classifier or DCS method that achieves the highest Macro-F1 score for all dimensions across different datasets. However, it is evident that the proposed method, DCS-DQ, consistently outperforms all other methods in terms of Macro-F1 score. The range of Macro-F1 scores for the Polarity dataset varies between 0.519 and 0.894. Notably, the highest Macro-F1 score achieved for feature sizes 500, 1000, and 3000 in the Polarity dataset is attributed to the DCS-DQ method. Consequently, it can be inferred that the DCS-DQ method demonstrates remarkable performance compared to all other methods in the Polarity dataset. Moreover, the highest Macro-F1 score for feature sizes 1000 and 3000 in the Enron 1 dataset, 500 and 1000 in the Ohsumed dataset, and 300, 500, 1000, and 3000 in the Reuters dataset is achieved by the DCS-DQ method. This further reinforces the effectiveness of the DCS-DQ method across multiple datasets and feature sizes.

7.4. The analysis of the overall performance of DCS-DQ method on all datasets

To elucidate the results depicted in Figure 8 and Figure 9, the names of the methods corresponding to the peak points for each feature size are provided in Table 5. In the last column of the table, the performance ratio of the proposed DCS-DQ method is presented. This ratio indicates the number of peak points reached by the DCS-DQ method for each classifier on each dataset. For instance, the DCS-DQ method attains the peak point

Table 3: Macro-F1 Scores for Polarity and Enron 1 datasets

Datasets	Polarity					Enron1				
Feature Size	100	300	500	1000	3000	100	300	500	1000	3000
KNN	0.712	0.701	0.707	0.622	0.519	0.873	0.871	0.858	0.809	0.667
SVM	0.768	0.832	0.848	0.862	0.890	0.877	0.925	**0.949**	0.961	0.973
DT	0.670	0.669	0.662	0.651	0.650	0.888	0.923	0.924	0.926	0.921
MNB	**0.807**	**0.858**	0.873	0.876	0.889	0.809	0.899	0.926	0.947	0.959
LR	0.803	0.841	0.854	0.857	0.877	0.869	0.914	0.939	0.951	0.963
Rank	0.731	0.782	0.796	0.761	0.750	0.896	0.924	0.922	0.913	0.806
OLA	0.734	0.785	0.814	0.802	0.772	0.895	0.918	0.904	0.898	0.803
LCA	0.710	0.729	0.739	0.708	0.628	0.872	0.913	0.894	0.858	0.663
A Priori	0.725	0.754	0.768	0.779	0.766	**0.899**	0.925	0.920	0.923	0.829
A Posteriori	0.713	0.732	0.748	0.756	0.733	0.885	0.917	0.922	0.919	0.735
MCB	0.747	0.792	0.808	0.803	0.796	0.895	**0.935**	0.926	0.932	0.904
MLA	0.710	0.729	0.739	0.708	0.628	0.872	0.913	0.895	0.858	0.663
DCS-DQ	0.775	0.849	**0.881**	**0.880**	**0.894**	0.894	0.930	0.947	**0.971**	**0.977**

Table 4: Macro-F1 Scores for Ohsumed and Reuters datasets

Datasets	Ohsumed					Reuters				
Feature Size	100	300	500	1000	3000	100	300	500	1000	3000
KNN	0.613	0.664	0.672	0.639	0.482	0.637	0.600	0.593	0.410	0.610
SVM	0.578	0.653	0.689	0.720	**0.761**	0.599	0.635	0.605	0.557	0.652
DT	0.577	0.624	0.627	0.624	0.600	0.554	0.649	0.609	0.575	0.664
MNB	0.394	0.492	0.551	0.592	0.600	0.474	0.654	0.601	0.581	0.669
LR	0.596	0.667	0.696	0.718	0.744	0.384	0.660	0.590	0.553	0.664
Rank	0.615	0.681	0.694	0.679	0.632	0.722	0.722	0.714	0.691	0.680
OLA	0.642	0.686	0.698	0.695	0.690	**0.730**	0.732	0.725	0.714	0.708
LCA	0.627	0.654	0.667	0.648	0.524	0.712	0.688	0.671	0.620	0.548
A Priori	0.611	0.663	0.678	0.678	0.650	0.728	0.740	0.737	0.722	0.727
A Posteriori	**0.647**	**0.693**	0.706	0.690	0.622	0.728	0.709	0.699	0.651	0.592
MCB	0.637	0.686	0.703	0.708	0.701	0.729	0.737	0.737	0.724	0.728
MLA	0.627	0.654	0.667	0.648	0.525	0.711	0.688	0.671	0.620	0.548
DCS-DQ	0.610	0.685	**0.709**	**0.731**	0.756	0.725	**0.746**	**0.752**	**0.753**	**0.755**

Table 5: DCS Methods producing the highest classification accuracies on all datasets for each feature size.

Datasets	Feature Size					Ratio of DCS-DQ
	100	300	500	1000	3000	
Polarity	DCS-DQ	DCS-DQ	DCS-DQ	DCS-DQ	DCS-DQ	1.0
Enron1	OLA	DCS-DQ	DCS-DQ	DCS-DQ	DCS-DQ	0.8
Ohsumed	A Posteriori	A Posteriori	DCS-DQ	DCS-DQ	DCS-DQ	0.6
Reuters	LCA	DCS-DQ	DCS-DQ	DCS-DQ	DCS-DQ	0.8

for all feature sizes on the Polarity dataset, resulting in a performance ratio of 1. Similarly, the DCS-DQ method outperforms other methods for 4 out of 5 feature sizes, leading to a performance ratio of 0.8 for DCS-DQ on the Enron1 dataset.

Table 5 demonstrates that the proposed method is more effective than existing state-of-the-art methods.

7.5. The statistical analysis of the overall performance of DCS-DQ method

The Paired Samples t-Test [53] was employed to analyze the experimental findings. This statistical test compares the means of a variable observed in two different situations. In our study, we compared the classification accuracy of the DCS-DQ method with that of existing methods. Our analysis revealed a significant difference between the DCS-DQ method and the other 7 methods. The results of this comparison are presented in Table 6.

Hypotheses for Paired Samples t-Test;

Null Hypothesis (H0): With 95% confidence, there is no statistically significant difference between the mean classification accuracy before and after the experiment. $(M1 = M2)$

Alternative Hypothesis (H1): With 95% confidence, there is a statistically significant difference between the mean classification accuracy before and after the experiment. $(M1 \neq M2)$

The decision about statistical significance is given using the p values on Table 6. Since the p-values are less than $0.05 (0.000 < 0.05)$ for all seven pairs in Table 6, the null hypothesis (H0) is rejected. This indicates that there is a statistically significant difference between the mean classification accuracy before and after the experiment. Therefore, the alternative hypothesis (H1) is accepted. Consequently, we can infer for all pairs that the DCS-DQ method is statistically significant with 95% confidence. In other words, the DCS-DQ method is effective in improving classification accuracy.

7.6. Time analysis of DCS methods

We compared the time anaysis of the proposed method with other methods, and the results of the comparison are presented in Table 7. The table presents the time analysis for the Polarity and Ohsumed datasets. All values are expressed in milliseconds and denote the average classification time for only one test sample in the respective dataset. The best-performing DCS method for any number of feature size is highlighted in bold. For the Polarity dataset, the MCB method is the fastest performing DCS method for 100 attributes. Interestingly, the MCB method also stands out as the fastest performing DCS method for both the Polarity and Ohsumed datasets overall. Although the running time of our proposed method is not better than that of other DCS methods, it is not significantly worse. Given its strong performance in terms of classification accuracy, the slight disadvantage in running time can be considered negligible.

Table 6: Paired Samples t-Test results

		Paired Differences					t	df	p
		Mean	Std. Deviation	Std. Err. Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	DCS-DQ & Rank	5.91	4.89	1.09	3.63	8.20	5.41	19	0.000
Pair 2	DCS-DQ & OLA	4.64	4.06	0.91	2.74	6.54	5.10	19	0.000
Pair 3	DCS-DQ & LCA	9.32	9.55	2.14	4.85	13.78	4.36	19	0.000
Pair 4	DCS-DQ & A_Priory	5.05	3.88	0.87	3.24	6.87	5.82	19	0.000
Pair 5	DCS-DQ & A Posteriory	7.57	7.10	1.59	4.24	10.89	4.77	19	0.000
Pair 6	DCS-DQ & MCB	4.66	4.19	0.94	2.79	6.62	4.97	19	0.000
Pair 7	DCS-DQ & MLA	10.57	9.31	2.08	6.22	14.93	5.08	19	0.000

Table 7: Time analysis of DCS methods

Dataset	Polarity					Ohsumed				
Feature Size	100	300	500	1000	3000	100	300	500	1000	3000
DCS-DQ	3.4	8.5	18.0	37.0	101.8	14.3	51.1	101.8	213.4	761.8
DCS-Rank	2.3	7.2	15.0	30.0	76.2	10.4	48.3	102.7	208.4	556.0
OLA	2.4	8.3	17.5	35.8	92.0	10.9	43.5	96.1	203.6	547.4
LCA	2.2	7.1	14.7	29.3	**71.7**	**9.6**	39.2	87.7	190.0	550.7
A Priory	2.8	8.5	17.7	36.3	97.9	11.3	43.5	98.0	213.0	583.4
A Posteriory	2.5	8.2	17.0	34.5	95.3	11.6	44.1	100.1	218.5	594.6
MCB	**2.1**	**6.9**	**13.8**	**29.0**	74.0	9.8	**37.9**	**87.3**	**186.3**	**483.3**
MLA	2.2	7.2	15.1	30.7	78.3	10.9	44.5	96.3	211.9	640.5

8. Conclusions and Future Works

The purpose of the current study is to propose a new DCS method, namely DCS-DQ. DCS methods have been shown to improve classification accuracy in many classification problems. However, according to our research, this is the first time that DCS methods have been used in a text classification problem. Four different benchmark text datasets are used in experimental study. In all datasets used in experimental studies, classification accuracies were improved when compared to existing methods. The highest improvement in classification accuracy was observed in the Polarity dataset. The proposed method outperforms other DCS methods in terms of Macro F1 score. The DCS-DQ method demonstrates superior performance compared to other DCS methods in the Polarity dataset for feature sizes of 500, 1000 and 3000. Similarly, in the other three datasets, the proposed method outperforms other DCS methods based on the Macro F1 score. We also demonstrated that the DCS-DQ method is statistically significant with a 95% confidence. Classification accuracy is one of the most important success measure used in classification problems and the proposed method has been shown to have high classification accuracy. In the light of all the findings, we can infer that the proposed DCS-DQ method can make a significant contribution to the text classification literature. The difference between the classification accuracy of DCS methods and Oracle classification accuracy is still quite large for all datasets. For example, when the number of attributes in Ohsumed dataset is 3000, the Oracle classification accuracy is 89.04%. For the same number of attributes, the classification accuracy of the DCS-DQ method is 77.02%. In future studies, developing different DCS methods to close this gap can contribute to the classification literature. In this study, we used different classifier model. In the experiments, we used the CHI2 as a feature selection method. In our future studies, we will investigate the effects of alternative feature selection methods on classification accuracy. However, the proposed DCS-DQ method can be adapted and applied to other pattern recognition problems, such as credit scoring, face recognition systems and music genre classification.

References

[1] Kowsari K, Jafari MK, Heidarysafa M, Mendu S, Barnes L, Brown D. Text classification algorithms: A survey. Information 2019; 10(4), 150. https://doi.org/10.3390/info10040150.

[2] Abooraig R, Al-Zu'bi S, Kanan T, Hawashin B, Al Ayoub M, Hmeidi I. Automatic categorization of Arabic articles based on their political orientation. Digital Investigation 2018; 25, 24-41. https://doi.org/10.1016/j.diin.2018.04.003.

[3] Iyer RD, Lewis DD, Schapire RE, Singer Y, Singhal A. Boosting for document routing. In Proceedings of the ninth international conference on Information and knowledge management 2000; (pp. 70-77). https://doi.org/10.1145/354756.354794

[4] Kale SD, Prasad RS. Influence of language-specific features for author identification on Indian literature in Marathi. In Soft Computing and Signal Processing: Proceedings of 2nd ICSCSP 2019 2 (pp. 639-652). Springer Singapore. 2020. Springer. https://doi.org/10.1007/978-981-15-2475-2_59

[5] Uysal D, A.K. Uysal, AUTOMATIC CLASSIFICATION OF EFL LEARNERS'SELF-REPORTED TEXT DOCUMENTS ALONG AN AFFECTIVE CONTINUUM. Advanced Education 2022; p. 4-14. https://doi.org/10.20535/2410-8286.248091.

[6] Zhu L, Zhu Z, Zhang C, Xu Y, Kong X. Multimodal sentiment analysis based on fusion methods: A survey. Information Fusion 2023; 95, 306-325. https://doi.org/10.1016/j.inffus.2023.02.028

[7] Ghosh S, Razniewski S, Weikum G. Answering Count Questions with Structured Answers from Text. Journal of Web Semantics 2023; 76, 100769. https://doi.org/10.48550/arXiv.2209.07250

[8] Rao S, Verma AK, Bhatia T. Hybrid ensemble framework with self-attention mechanism for social spam detection on imbalanced data. Expert Systems with Applications 2023; 217, 119594. https://doi.org/10.1016/j.eswa.2023.119594

[9] Dietterich TG. Machine-learning research. AI magazine 1997; 18(4): p. 97-97. https://doi.org/10.1609/aimag.v18i4.1324

[10] Polikar R. Ensemble based systems in decision making. IEEE Circuits and systems magazine 2006; 6(3): p. 21-45. https://doi.org/10.1109/MCAS.2006.1688199

[11] Kuncheva LI. A theoretical study on six classifier fusion strategies. IEEE Transactions on pattern analysis and machine intelligence 2002; 24(2): p. 281-286. https://doi.org/10.1109/34.982906.

[12] Cruz RM, Sabourin R, Cavalcanti GD, Ren TI. META-DES: A dynamic ensemble selection framework using meta-learning. Pattern recognition 2015; 48(5), 1925-1935. https://doi.org/10.1016/j.patcog.2014.12.003

[13] Cruz RM, Sabourin R, Cavalcanti GD. Dynamic classifier selection: Recent advances and perspectives. Information Fusion 2018; 41: p. 195-216. https://doi.org/10.1016/j.inffus.2017.09.010

[14] Xiao H, Xiao Z, Wang Y. Ensemble classification based on supervised clustering for credit scoring. Applied Soft Computing 2016; 43: p. 73-86. https://doi.org/10.1016/j.asoc.2016.02.022

[15] Breiman L. Bagging predictors. Machine learning 1996; 24(2): p. 123-140. https://doi.org/10.1007/BF00058655

[16] Woźniak M, Graña M, Corchado E. A survey of multiple classifier systems as hybrid systems. Information Fusion 2014; 16: p. 3-17.https://doi.org/10.1016/j.inffus.2013.04.006

[17] Dietterich TG. Ensemble Methods in Machine Learning. 2000. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45014-9_1.

[18] Cruz RM, Sabourin R, Cavalcanti GD. A DEEP analysis of the META-DES framework for dynamic selection of ensemble of classifiers. arXiv preprint arXiv 2015; 1509.00825. https://doi.org/10.48550/arXiv.1509.00825

[19] Oliveira DV, Cavalcanti GD, Sabourin R. Online pruning of base classifiers for dynamic ensemble selection. Pattern Recognition 2017; 72: p. 44-58. https://doi.org/10.1016/j.patcog.2017.06.030

[20] Cavalin PR, Sabourin R, Suen CY. Dynamic selection approaches for multiple classifier systems. Neural computing and applications 2013; 22(3): p. 673-688. https://doi.org/10.1007/s00521-011-0737-9

[21] Melo L, Macêdo JF, Nardini FM, Renso C. RMkNN and KNORA-IU: Combining Imbalanced Dynamic Selection Techniques for Credit Scoring. In 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 823-830). IEEE. https://doi.org/10.1109/ICTAI52525.2021.00131

[22] Bashbaghi S, Granger E, Sabourin R, Bilodeau GA. Dynamic ensembles of exemplar-SVMs for still-to-video face recognition. Pattern recognition 2017; 69, 61-81. https://doi.org/10.1016/j.patcog.2017.04.014

[23] Porwik P, Doroz R, Wrobel K. An ensemble learning approach to lip-based biometric verification, with a dynamic selection of classifiers. Expert Systems with Applications 2019; 115: p. 673-683. https://doi.org/10.1016/j.eswa.2018.08.037

[24] Batista L, Granger E, Sabourin R. Dynamic ensemble selection for off-line signature verification. in International Workshop on Multiple Classifier Systems 2011; Springer. https://doi.org/10.1007/978-3-642-21557-5_18

[25] Xiao J, Xie L, He C, Jiang X. Dynamic classifier ensemble model for customer classification with imbalanced class distribution. Expert Systems with Applications 2012; 39(3), 3668-3675. https://doi.org/10.1016/j.eswa.2011.09.059

[26] Wen J, Gao H, Liu Q, Hong X, Sun Y. A new method for identifying the ball screw degradation level based on the multiple classifier system. Measurement 2018; 130, 118-127. https://doi.org/10.1016/j.measurement.2018.08.005

[27] Groccia M.C, Guido R, Conforti D. Multi-Classifier Approaches for Supporting Clinical Decision Making. Symmetry 2020; 12(5): p. 699, https://doi.org/10.3390/sym12050699

[28] Roy A, Cruz RM, Sabourin R, Cavalcanti G D. A study on combining dynamic selection and data preprocessing for imbalance learning. Neurocomputing 2018; 286, 179-192. https://doi.org/10.1016/j.neucom.2018.01.060

[29] Feng P, Ma J, Sun C, Xu X, Ma Y. A novel dynamic android malware detection system with ensemble learning. IEEE Access 2018; 6, 30996-31011. doi: https://doi.org/10.1109/ACCESS.2018.2844349

[30] Junior LM, Nardini FM, Renso C, Trani R, Macedo JA. A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. Expert Systems with Applications 2020; 152, 113351. https://doi.org/10.1016/j.eswa.2020.113351

[31] Feng X, Xiao Z, Zhong B, Dong Y, Qiu J. Dynamic weighted ensemble classification for credit scoring using Markov Chain. Applied Intelligence 2019; 49, 555-568. https://doi.org/10.1007/s10489-018-1253-8

[32] Martins JG, Oliveira LS, Britto AS, Sabourin R. Forest species recognition based on dynamic classifier selection and dissimilarity feature vector representation. Machine Vision and Applications 2015; 26, 279-293. https://doi.org/10.1007/s00138-015-0659-0

[33] Magalhães DMV. Ensemble of Classifiers for Multilabel Clinical Text Categorization in Portuguese. In Intelligent Systems Design and Applications: 22nd International Conference on Intelligent Systems Design and Applications (ISDA 2022) Held December 12-14, 2022-Volume 2 (Vol. 715, p. 42). Springer Nature. https://doi.org/10.1007/978-3-031-35507-3_5

[34] Li Y, Zhang S, Lai C. ”Agricultural Text Classification Method Based on Dynamic Fusion of Multiple Features,” in IEEE Access, vol. 11, pp. 27034-27042, 2023. https://doi.org/10.1109/ACCESS.2023.3253386

[35] Sergio AT, de Lima TP, Ludermir TB. Dynamic selection of forecast combiners. Neurocomputing 2016; 218: p. 37-50. https://doi.org/10.1016/j.neucom.2016.08.072

[36] Bhatnagar V, Bhardwaj M, Sharma S, Haroon S. Accuracy–diversity based pruning of classifier ensembles. Progress in Artificial Intelligence 2014; 2(2-3), 97-111. https://doi.org/10.1007/s13748-014-0042-9

[37] Cruz RM, Zakane HH, Sabourin R, Cavalcanti GD. Dynamic ensemble selection vs k-nn: why and when dynamic selection obtains higher classification performance?. In 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA) (pp. 1-6). IEEE. https://doi.org/10.1109/IPTA.2017.8310100

[38] Dogo EM, Nwulu NI, Twala B, Aigbavboa C. Accessing imbalance learning using dynamic selection approach in water quality anomaly detection. Symmetry 2021; 13(5), 818. https://doi.org/10.3390/sym13050818

[39] Groccia MC, Guido R, Conforti D. Multi-Classifier Approaches for Supporting Clinical Diagnosis. in International Conference on Optimization and Decision Science 2017; Springer. https://doi.org/10.1007/978-3-319-67308-0_13

[40] Woods K, Kegelmeyer WP, Bowyer K. Combination of multiple classifiers using local accuracy estimates. IEEE transactions on pattern analysis and machine intelligence 1997; 19(4): p. 405-410. https://doi.org/10.1109/34.588027

[41] Giacinto G, Roli F. Methods for dynamic classifier selection. in Proceedings 10th International Conference on Image Analysis and Processing. 1999. IEEE. https://doi.org/10.1109/ICIAP.1999.797670

[42] Smits PC. Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. IEEE Transactions on Geoscience and Remote Sensing 2002; 40(4): p. 801-813. https://doi.org/10.1109/TGRS.2002.1006354

[43] Liu G, Guo J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 2019; 337: p. 325-338. https://doi.org/10.1016/j.neucom.2019.01.078

[44] Cruz RM, Hafemann LG, Sabourin R, Cavalcanti GD. DESlib: A Dynamic ensemble selection library in Python. The Journal of Machine Learning Research 2020; 21(1), 283-287. https://doi.org/10.48550/arXiv.1802.04967

[45] Britto Jr AS, Sabourin R, Oliveira LE. Dynamic selection of classifiers—a comprehensive review. Pattern Recognition 2014; 47(11): p. 3665-3680. https://doi.org/10.1016/j.patcog.2014.05.003

[46] Hersh W, Buckley C, Leone TJ, Hickam D. OHSUMED: an interactive retrieval evaluation and new large test collection for research. in SIGIR’94 1994; Springer. https://doi.org/10.1007/978-1-4471-2099-5_20

[47] Klimt B, Yang Y. The enron corpus: A new dataset for email classification research. in European Conference on Machine Learning 2004; Springer. https://doi.org/10.1007/978-3-540-30115-8_22

[48] Asuncion A, Newman D. UCI machine learning repository 2007; Irvine, CA, USA.

[49] Uysal AK, Gunal S. The impact of preprocessing on text classification. Information Processing & Management 2014; 50(1): p. 104-112. https://doi.org/10.1016/j.ipm.2013.08.006

[50] Uysal AK, Gunal S. A novel probabilistic feature selection method for text classification. Knowledge-Based Systems 2012; 36: p. 226-235. https://doi.org/10.1016/j.knosys.2012.06.005

[51] Lever J. Classification evaluation: It is important to understand both what a classification metric expresses and what it hides. Nature methods 2016; 13(8): p. 603-604. https://doi.org/10.1038/nmeth.3945

[52] Huang YS, Suen CY. The behavior-knowledge space method for combination of multiple classifiers. in IEEE computer society conference on computer vision and pattern recognition 1993; Institute of Electrical Engineers Inc (IEEE). https://doi.org/10.1109/CVPR.1993.1626170

[53] Ross A, Willson VL. Paired samples T-test, in Basic and advanced statistical tests 2017; Springer. p. 17-19. https://doi.org/10.1007/978-94-6351-086-8_4