A New Approach for Text Documents Classification with Invasive Weed Optimization and Naive Bayes Classifier

Saman Khanalni[1], Farhad Soleimanian Gharehchopogh[2]
1,2 Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmai, Iran.
2 (bonab.farhad@gmail.com)

Received (2018-07-26) Accepted (2018-09-22)

Abstract: *With the fast increase of the documents, using Text Document Classification (TDC) methods has become a crucial matter. This paper presented a hybrid model of Invasive Weed Optimization (IWO) and Naive Bayes (NB) classifier (IWO-NB) for Feature Selection (FS) in order to reduce the big size of features space in TDC. TDC includes different actions such as text processing, feature extraction, forming feature vectors, and final classification. In the presented model, the authors formed a feature vector for each document by means of weighting features use for IWO. Then, documents are trained with NB classifier; then using the test, similar documents are classified together. FS do increase accuracy and decrease the calculation time. IWO-NB was performed on the datasets Reuters-21578, WebKb, and Cade 12. In order to demonstrate the superiority of the proposed model in the FS, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have been used as comparison models. Results show that in FS the proposed model has a higher accuracy than NB and other models. In addition, comparing the proposed model with and without FS suggests that error rate has decreased.*

Keywords: *Text Document Classification, Invasive Weed Optimization, Naive Bayes, Feature Selection.*

How to cite this article:
Khanalni S, Soleimanian Gharehchopogh F. A New Approach for Text Documents Classification with Invasive Weed Optimization and Naive Bayes Classifier. J. ADV COMP ENG TECHNOL, 4(3) Summer 2018 : 167-184

I. INTRODUCTION

TDC is a major part of content analysis of texts and is used in many applications such as text filtering, automatic response systems, and application relevant to automatic organization of documents [1]. Nowadays, a huge mass of information and knowledge is in digital text format. Considering the growth rate of knowledge, document classification to the end of reducing information complexity and easy and quick accessing to information is a very important issue. The purpose of document classification is accessing the data quickly. Nonexistence of a classification system leads to cost increase and spending more time for carrying out for text operations. This is because of the long time needed to be spent on finding the documents in traditional document classification methods.

Many documents are stored in electronic text formats. A required model for extracting knowledge from this big mass of text information is using TDC. As a significant technique in information retrieval and natural language processing, information classification is challenging and an effective solution for organizing text databases [2]. Considering the growth of electronic texts and documents, using an efficient method for data retrieval is mandatory. For retrieving data, understanding the main concept of the text, text classification, finding the proper words for searching, and keyword extraction are the best ways. Keywords are a set of important words in a document that provide us with a description of the document content. They are useful for different purposes. Through finding the keywords, we can have a grasp on the contents of text documents [3]. Overall, keywords are a useful tool for searching a big mass of documents in a short time. Two major methods for keywords extraction are [4]:

- Term Frequency-Inverse Document Frequency (TF-IDF) methods: in these methods, the repetition frequency of a word in a document is regarded in contrast with its repetition frequency in the whole document set.
- Machine Learning (ML) methods: in these methods, by means of a set of training documents and specific keywords for them keyword extraction process is modeled as a classification problem. These methods are highly flexible.

Text document analysis through ML techniques, intelligent information retrieval, natural language processing, and etc. is a subcategory of data mining. These techniques were first applied to structured data which those that are same in structure but are gathered in a file completely independent of one another. However, in the case of text documents that are mainly either unstructured or semi-structured we must first make them structured and then use these methods for extracting information and knowledge from them [5].

TDC means assignment of text documents according to their content to one or more predefined classes. The goal of TDC is to assign text documents to different predefined classes. In classification, there is a training set of documents with specific classes by means of this set, classification is specified, and the class of the new document is determined. For measuring the effectiveness of a TDC model, a test set is defined independent of the training set. The estimated labels are compared with the real labels. The ratio of correctly classified documents to the total documents is calculated based on the accuracy.

In this paper, we propose a hybrid model of IWO [6] and NB classifier [7] for TDC. In the IWO-NB model, we use IWO for FS and NB classifier for classification of similar groups. However there are many volumes of words in a typical text collection, most of the words contain little or no information in the TDC. Thus FS or dimension reduction becomes necessary because it not only reduces the measurement and storage requirements, but also improves prediction performance. IWO is a new and powerful optimization algorithm that imitates the adaptability and randomness of IWO colonies. By definition, IWO is a plant that grows and reproduces in unintended places and according to the environment; it acts as a pest for useful agricultural plants and hinders their growth. Even though it is very simple, IWO is very quick and effective in finding the optimum locations; and acts similar to features of the original and natural IWO in reproduction, growth, and struggle for survival in a colony. NB algorithm is a technique of data mining for classification. NB has characteristics such as simplicity, high computational efficiency, and good classification accuracy, especially for high dimensional data such as texts. In this technique, different classes are considered as a supposition with a probability. Any new training data increases or decreases the probability of prior hypotheses; and eventually the hypotheses with the highest probabilities are considered as a class and are assigned to a label.

Commonly used feature selection methods are the filter methods, such as chi-square (CHI), information gain (IG). Some comparative studies are given by [8, 9, and 10]. These methods simply calculate the scores for each feature and then remove those features with small scores. In this paper, three types of metaheuristic algorithms such as IWO, Genetic Algorithm (GA) [11] and PSO [12] algorithm were used to extract the features, due to the necessity of selecting the feature and achieving high precision. The reason for choosing IWO's algorithm for FS compared to GA and PSO models is the fact that the precision of detection of IWO is high and also it is more precise in choosing the feature.

The remainder of this paper is organized as follows: in Section 2, we review the related works done on TDC. In Section 3, the proposed model is described. In Section 4, experimental results are introduced and also models of GA-NB, PSO-NB, and IWO-NB for FS and classification is presented. In Section 5, assessment of the results of the proposed model is carried out; and the model is compared with other models. And eventually, in Section 6, conclusions are made and suggestions are made for future studies.

II. RELATED WORK

Considering the big volume and wide domain of text documents that are available from online and other sources, unless they are properly classified, retrieval and processing of unclassified text documents will face many problems. The most significant step in classification of text documents is choosing the proper feature space; and accuracy of a model depends highly on the chosen keywords that define the domain of the document.

K-Nearest Neighbor (KNN) model and TF-IDF have been recommended for classification of text documents [13]. Results are, performed on WebKb dataset; highest classification accuracy value for KNN is 0.92. Hybrid model of Support Vector Machine (SVM), NB, and KNN has been recommended for TDC under the title TCFP [14]. Assessment is carried out on Reuters-21578 [30], WebKb [31], and Cade 12 [31] datasets. Accuracy of the factor F-Measure for the three datasets is 86.19, 75.47, and 89.09 respectively. In comparison with SVM, NB, and KNN it has a higher accuracy.

Hybrid model of KNN and Bat Algorithm (BA) has been recommended for TDC [15]. In this model, they used BA for FS and KNN algorithm for text similarity. That so, text documents are first preprocessed; and the keywords in the document are extracted. Then, based on repetition a specific weight is set for each keyword. Assessment is carried out on Reuters-21578, WebKb, and Cade 12 datasets. Comparisons suggest that the proposed model is more accurate than the models K-Means, K-Means-KNN, and NB-K-Means.

Hybrid model KNN-K-Means [16] has been recommended for clustering of text document. In this model, KNN algorithm is used for identifying similar clusters; and K-Means algorithm is used for accuracy in document clustering. Result on Reuters-21578 show that the proposed model is more accurate than K-Means model.

Hybrid model NB-K-Means has been tested on the datasets Reuters-21578, WebKb, and Cade 12 datasets for TDC [17]. Results indicate that the hybrid model NB-K-Means is more accurate than K-Means model. Moreover, the highest accuracy in the proposed model is that of K=3 which is %93.30. Models have been recommended for reducing the size of data using PSO algorithm [18]. PSO algorithm has been used along with hybrid of fuzzy, NB, and SVM models. Results were assessed on Reuters and OHSUMED datasets. Assessments indicate that the accuracy of fuzzy model is higher than other models.

GALSF model [19] has been proposed based on GA and effective FS. In this model, other than FS, the relations between features have been considered; and these relations have been used for finding similar classes. Each feature gets a score according to repetition; and the features with the highest scores are influential in classification and the number of classes. Results of dataset Reuters-21578, shows that GALSF model is more accurate than other models.

A model has been proposed based on semantic web and WordNet for text document clustering [20]. In the model based on semantic web, closeness and synonymy of features have been used for accuracy of clustering. Based on the semantic model of the words, each cluster chooses a feature as cluster head, and if some features are vague, WordNet is used for finding semantic similarity. Results are obtained from performing on Reuters-21578 dataset and the percentage of features and clusters distribution is shown according to F-Measure factor.

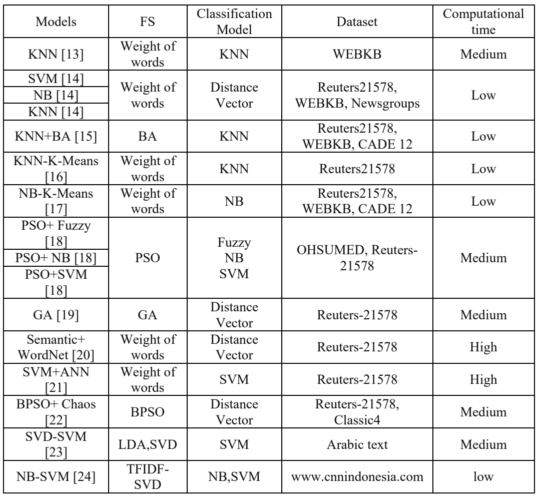
TESC model [21] using SVM and back propagation Artificial Neural Network (ANN) has been recommended for classification of text documents of Reuters-21578 dataset. SVM is a method for data classification based on two pages. Data is grouped on top and bottom of the page. The ANN assesses document identification accuracy based on data training and testing. Results suggest that accuracy of back propagation ANN is lower than that of SVM.

Bharti et al. suggested chaotic BPSO model hybrid for text document clustering [22]. Chaos factor was used for selecting optimum features of BPSO model. First, using BPSO, at feature indexing stage, features are selected; then, using chaos, closeness of features and selection of similar features in one vector are done. Results of performing on Reuters-21578, Classic4, and WebKb show that in identification BPSO model is more accurate than models SGA, CBPSO, and AIWPSO.

AbuZeina, and Al-Anzi [23], proposed the capacity of Linear Discriminant Analysis (LDA) for Arabic text classification. LDA, also known as Fisher's LDA, is one of the popular dimensionality diminish techniques that can show good performance in pattern recognition tasks. On other words, the present study is an attempt to understand whether the LDA is adequate for text classification such as other celebrated successful implementations, face recognition is an example. The prior art shows that the LDA is rarely used for Arabic text classification despite its good capabilities in dimensionality reduction. Therefore, this work is focused on the implementation of the LDA method for Arabic text classification as such applications generally contain sizable vocabularies that lead to large features and vectors. The results of experiments showed that the efficiency of the semantic loss LDA feature vectors is almost the same as the semantic rich latent semantic indexing (LSI) method. In opposition to, LSI employs an SVD method to generate semantic rich features. Semantic rich means that the method preserves and understands the inherent latent relationships between the words in the different documents. Besides the LAD, there is a one favorable dimensionality reduction technique such as singular value decomposition (SVD). Benchmarks comparison showed that the LDA is one of the worthy methods as it gives promising results when compared with SVM, KNN, NN, NB, cosine measure, etc. For instance, the SVD-SVM scored accuracy up to 84.75% while the LDA scored 84.4%. This results point out to the important of employing LDA for text classification.

In [24], news articles which are publicized in www.cnnindonesia.com are crawled with the total number of 5,000 documents. The listed documents consist of 1,000 documents for each class of: Health, Sports, Economy, Politic, and Technology. The documents are randomly partition with the ratio of 80:20 for training and testing goals. The feature selections in this research are done by using TF-IDF and SVD. The classifier used in the experiments of this research is NB and SVM. Comparisons have been done based on the TF-IDF and Singular Value Decomposition (SVD) algorithm for FS, while also compared the Multinomial Naïve Bayes (MNB), Bernoulli Multivariate Naïve Bayes (BNB), and SVM for the classifiers. Based on the test results, the hybrid of TF-IDF and MNB classifier gave the highest result compared to the other algorithms, which precision is 0.9841 and its recall is 0.9840. The hybrid of TFIDF + Multinomial Naïve Bayes (MNB) has provided the highest value of precision and recall, which is around 98.4% followed by the combination of TFIDF and BNB, which is around 98.2%. In terms of time consumed to process the data, MNB and BNB both gave the best result despite having very huge amount of data extracted by TF-IDF. In Table (1), comparison of the proposed models for TDC by researchers is shown.

TABLE1: COMPARISON OF THE PROPOSED MODELS FOR TDC



Models	FS	Classification Model	Dataset	Computational time
KNN [13]	Weight of words	KNN	WEBKB	Medium
SVM [14]	Weight of words	Distance Vector	Reuters21578, WEBKB, Newsgroups	Low
NB [14]				
KNN [14]				
KNN+BA [15]	BA	KNN	Reuters21578, WEBKB, CADE 12	Low
KNN-K-Means [16]	Weight of words	KNN	Reuters21578	Low
NB-K-Means [17]	Weight of words	NB	Reuters21578, WEBKB, CADE 12	Low
PSO+ Fuzzy [18]	PSO	Fuzzy NB SVM	OHSUMED, Reuters-21578	Medium
PSO+ NB [18]				
PSO+SVM [18]				
GA [19]	GA	Distance Vector	Reuters-21578	Medium
Semantic+ WordNet [20]	Weight of words	Distance Vector	Reuters-21578	High
SVM+ANN [21]	Weight of words	SVM	Reuters-21578	High
BPSO+ Chaos [22]	BPSO	Distance Vector	Reuters-21578, Classic4	Medium
SVD-SVM [23]	LDA,SVD	SVM	Arabic text	Medium
NB-SVM [24]	TFIDF-SVD	NB,SVM	www.cnnindonesia.com	low

The KNN and SVM models are performance classifiers, but the KNN does not have unique results, and each time it executes a non-similar response to the previous one. The KNN model uses all educational prototypes when it comes to decision making, which involves some disadvantages, including low speed classification and high memory requirements. The SVM model, despite having unique results, has high computational time. The SVM model, with the entry of unclassified new samples, uses all of the previous educational protocols in the classifier update, which has a high cost. Early convergence is one of the main problems in the PSO algorithm. The particle gradually rotates in the search space near the optimal general point and does not explore the rest of the space, in other words particles converge. Because the particle velocity decreases with increasing frequency, therefore, the algorithm has to converge to the best that has been discovered so far and is not guaranteed to be the best global solution. This is the result of an inappropriate balance between local and global searches. In PSO algorithm, is preferred in first repetitions of global search, and it helps to improve performance and in final repetitions, global searches are reduced, and in order to maximize the information obtained, local searches are preferred.

III. PROPOSED MODEL

When there is independence of features, the accuracy of NB classifier also decreases. In the IWO-NB model, IWO algorithm is used for enhancing the accuracy of NB classifier. Features with the lowest differentiation effect are omitted considering the total of omitted features, and the remaining features are given to NB. The text could have too many features and/or correlated features, which cause both inefficiency and inaccuracy during TDC. As a result, ranking the features by their distinctiveness and only selecting the distinctive ones to perform TDC can help achieve a better TDC performance. Carrying out the FS process based on IWO algorithm results in enhancement of NB classifier in domains with codependent features. In addition, due to omission of less important features, the proposed model increases the calculation speed and yields the optimum answer in a shorter time. In Figure (1), the flowchart of the proposed model is presented.

Table (2) displays the pseudo code of the proposed model.

In reading the datasets stage, the datasets Reuters-21578, WebKb, and Cade 12 are read and then they enter the preprocessing stage. In preprocessing, omission of irrelevant words and verbs takes place. In this step, is removed the functional words that are used to construct nature language documents but not related to any specific topics, such as “a”, “an”, “the”, “in”, “of”, “to”, etc. In the context of TDC, functional words are common words that are not related to the concept of the text. The stop-words to be consist of the pronouns, conjunctions, papers, and prepositions that should be removed for the sake of dimension reduction. In keywords extraction stage, using Equation (1), keywords counting takes place. The two basic parameters in term weighting strategies are raw term frequency TF (number of terms in D) and inverse document frequency IDF (term occurrence across a collection). In this paper, we used TF to obtain weight of the terms and then converted the results to vector space model, Di={wi1, wi2,…,wit}. Here, i, w and t denote the index of document, the weight of the terms in the document and the total number of terms, respectively. Equation (1) is one of the TF methods in which (tk, di) is repetition frequency of each feature tk in the document di [25].

$$w_{ki} = tf(t_k, d_i) = \begin{cases} (t_k, d_i) & t_k \in vector\, of\, d_i \\ 0 & t_k \notin vector\, of\, d_i \end{cases} \tag{1}$$

Different weighting schemes such as the Term Frequency (TF) model [25], and TF-IDF model [25] can be used to assign a weighting value for each term feature and, accordingly, determine the document vector. The weighting is often associated with the frequency of each term. In IWO algorithm stage, the initial population and the vectors are formed. In this stage, vectors are formed based on words’ weight. IWO algorithm starts the search, scrutinizes the distribution of weights, and for weights similar to one another defines one vector. The operation goes on until the placement of weights in the vectors. Then the vectors are assessed and their fit is calculated according to Equation (2).

$$vector_{(i)} = \sum_{i=0}^{n} \frac{S_{(i)}}{n}, S = [S_{\min}, S_{\max}] \tag{2}$$

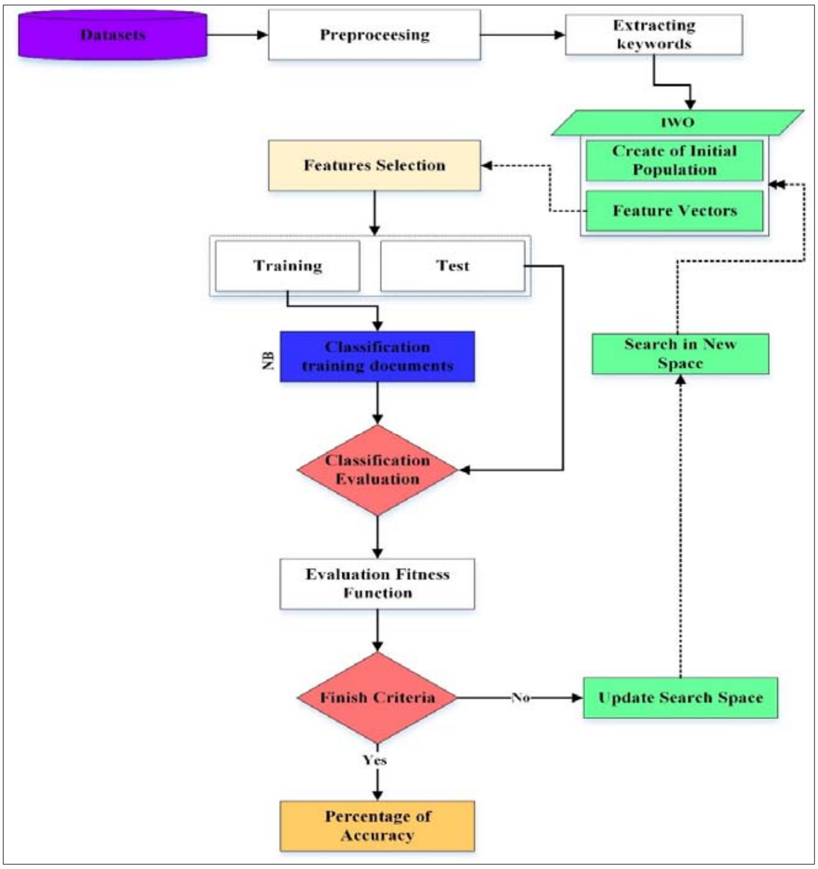


Fig. 1. Flowchart of the Proposed Model

TABLE 2 : THE PSEUDO CODE OF THE PROPOSED MODEL

```
01. start
02. reading the datasets
02.1. Reuters-21578 dataset
02.2. WebKb dataset
02.3. Cade 12 dataset
03. preprocessing the text documents
03.1. omission of prepositions and irrelevant verbs
04. Keyword extraction
05. IWO Algorithm
05.1. Forming the initial population
05.2. Forming the vectors
05.3. Assessment of the vectors based on the average weight of the features of each vector
06. Selecting sub-features
06.1. selecting the best vector with the highest fit value
07. data training
08. data testing
09. NB Classifier
10. TDC based on document training
10.1. testing the trained data
10.2. assessment of classification based on the test
11. assessment of fit function regarding accuracy
12. Is the finishing factor final?
        If answer=yes then 12.1
12.1. final output of the proposed model based on accuracy
        If answer=no then 12.2
12.2. updating search space
12.3. search in new space
13. end
```

In IWO, each weed in the population shows one candidate solution for the problem. Each weed contains some position having dim dimension which is denoted as vector and that have values either 0 or 1 as shown in Figure (2). Each dimension is treaded as one feature. From the Fig. 4, we can say that Weed X has dim (here dim=200) feature that has value either 0 or 1. If the value at position j is 1 that means j the feature is selected otherwise it is not selected. For generate values 0 and 1, we change IWO to Binary IWO [36]. Binary IWO determines its binary seeds in a normally distributed neighborhood in the space of bit-strings (0 or 1). The normal distribution is realized over the number of different bits.

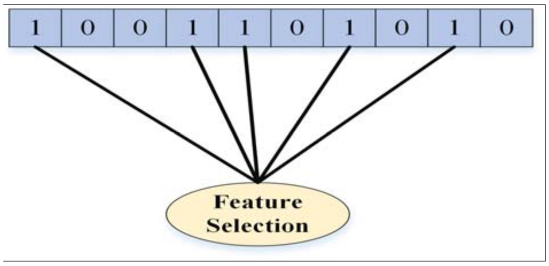


Fig. 2. Solution representation of a weed

The objective function for the proposed model is the mean absolute difference (MAD) [37].

$$MAD(S_i) = \frac{1}{s_i}\sum_{k=1}^{j} | x_{i,k} - \overline{x_l} | \tag{3}$$

$$\overline{x_l} = \frac{1}{s_i}\sum_{k=1}^{j} x_{i,k} \tag{4}$$

Where, is the number of selected features in text document , is the mean value of the vector is the weighting value of feature k in document i and j is the number of features in the original text dataset.

After that, in sub-features selection stage, vectors are chosen and enter testing and training stages. For classifying text documents, first we divide them into two sets, namely training and testing. We form the model with training set and study it with testing set, so that the previous model would have a high accuracy. In fact, testing set is formed to be used for determining the accuracy of the model formed from the training set. In addition, classification of training documents is done according to NB classifier.

Assessment of fit function is carried out to certify accuracy. If the accuracy of the classification is deemed acceptable, the classification is shown as output; otherwise, search space is updated for getting to a better answer. For updating the search space, changes need to be made to solutions vector. For these changes, we use cosine distance according to Equation (5) [26].

$$d(v,w) = \frac{v_i w_i . v_j w_j}{\|v_i w_i\| \times \|v_j w_j\|} = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}} \tag{5}$$

In Equation (5), w and v are word weight and vector respectively. Each vector is assessed with its word weight. If the value of the first vector is bigger than that of the second vector, a random number of vectors' indices are switched according to Equation (6).

$$v_i^k = \left\lfloor w_{\max} - (w_{\max} - w_{\min}) \frac{v_i^k - f_{\min}}{w_{\max} - f_{\min}} \right\rfloor \tag{6}$$

In Equation (6), the parameter wmax is the highest weight value in the vector k, vi is the ith index of the vector k, and fmin is the fit function of the kth vector.

3.1. Naive Bayes Classifier

In NB classifier, classification input include parameter d, i.e. text documents, $C=\{c_1,c_2, \ldots,c_j\}$, i.e. classes, and training data, $(d_1,c_1), \ldots,(d_m,c_m)$. NB classifier is defined for documents and classes according to Equation (7). In Equation (7), parameters w and c are number of words and documents respectively.

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)} \Rightarrow$$
$$P(w \mid c) = \frac{P(c \mid w)P(w)}{P(c)} = p(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|} \tag{7}$$

NB classification starts with the initial step of analyzing the text document by extracting words which are contained in the document to generate a list of words. The list of words is constructed with the assumption that input document contains words w_1, w_2, w_3,…, w_{n-1}, w_n, where the length of the document (in terms of number of words) is n. For explanation of NB classifier in TDC, note Table (3). In Table (3), there are 4 training documents and 1 test document. We determine the neighborhood of the words by means of NB and allocate document 5 to class *c*.

TABLE3:
WORD CLASSIFICATION WITH NB CLASSIFIER

Dataset	Documents	Words	Class
Training	1	Program Project Project	c
	2	Project Pipeline Program Project	c
	3	Project Structure	c
	4	Computer Software Project	j
Test	5	Project Project Project Computer Software	?

In Table (3), the probability of *c* and *j* are P(c)=3/4 and P(j)=1/4. In Table (4), percentage of the words in documents *c* and *j* is assessed. In NB classification method, all features are assumed to be independent and have different weight.

TABLE 4:
EXAMINING CLASSES' PROBABILITY FOR TDC

Class	Evaluating Possibilities
Class c	P(Project |c)=(6+1)/(8+6)=6/14=3/7
	P(Computer |c)=(0+1)/(8+6)=1/14
	P(Software |c)=(0+1)/(8+6)=1/14
Class j	P(Project |j)=(1+1)/(3+6)=2/9
	P(Computer |j)=(1+1)/(3+6)=2/9
	P(Software |j)=(1+1)/(3+6)=2/9

In Table (4), we see if document number 5 is closer to document *c* or document *j*. In Table (5), the probability of class *c* is higher. Therefore, document number 5 belongs to *c*. the probability of *c* is higher because in document number 5 the word *project* is repeated 3 times.

TABLE5: CLASS SELECTION FOR AN UNIDENTIFIED DOCUMENT IN NB CLASSIFIER

class selection
P(c |d5)=3/4*(3/7)3*1/14*1/14=0.0003
P(j |d5)=1/4*(2/9)3*2/9*2/9=0.0001

3.2. Assessment Factors

The results of the proposed model must be analyzed at assessment stage in order to reveal their value and as a result the effectiveness of the model. We can calculate these factors both for the training datasets at the training stage and for training records at the assessment stage. There are different factors for assessment such as precision, recall, F-Measure, and accuracy. For assessment of the IWO-NB model, we use the factor accuracy [27] [28]. Precision (P), Recall (R), and F-Measure are widely used metrics in the text mining literature for the text categorization. Precision measures total number of correct positive predictions to the total numbers of positive predictions and Recall measures total number of correct positive predictions to the total number of positive documents. F-Measure is a harmonic hybrid of P and R.

$$\Pr ecision = \frac{TP}{TP+FP} \tag{8}$$

$$\operatorname{Re} call = \frac{TP}{TP+FN} \tag{9}$$

$$F - Measure = \frac{2 * \Pr ecision * \operatorname{Re} call}{(\Pr ecision + \operatorname{Re} call)} \tag{10}$$

$$AUC = \left(\left(\frac{TP}{TP+FN}\right) + \left(\frac{TN}{TN+FP}\right)\right)/2 \tag{11}$$

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{12}$$

$$ErrorRate = \frac{(FP+FN)}{(TP+TN+FP+FN)} = 1 - Accuracy \tag{13}$$

The parameter TN represents the records with a real positive class that were correctly identified as positive by the algorithm. TP represents the records with a real negative class that were correctly identified as negative by the algorithm. FP represents the records with a real negative class but mistakenly identified as positive by the algorithm. FN represents the records with a real positive class but mistakenly identified as negative by the algorithm.

IV. EXPERIMENTAL RESULTS

In first, the performance of models in this paper has been tested using 13 different datasets. These datasets are taken from UCI machine learning repository [29] and their description is given in Table (6). Some of these data sets show missing data. The missing data is replaced with the average of the values taken on by the features; in addition, the dataset features are normalized. The sizes of the train and test sets are shown in Table (6). Thus, 75% of data is used in training process as a train set, and the remaining 25% of data is used in testing process as a test set. Three criteria were reported to evaluate each approach: classification accuracy, Error Rate, computational time.

TABLE 6: DATASETS DESCRIPTION

#	dataset	Instances	training	testing	Features	Classes	Missing
1	Balance	625	469	156	4	3	No
2	Cancer	569	427	142	30	2	No
3	Cancer-Int	699	524	175	9	2	Yes
4	Credit	690	518	172	51	2	Yes
5	Dermatology	366	274	92	34	6	Yes
6	Diabetes	768	576	192	8	2	No
7	E.Coli	327	245	82	7	5	No
8	Glass	214	161	53	9	6	No
9	Heart	303	227	76	35	2	Yes
10	Horse	364	273	91	58	3	Yes
11	Iris	150	112	38	4	3	No
12	Thyroid	215	162	53	5	3	No
13	Wine	178	133	45	13	3	No

The performance of the models is assessed by various analyses using datasets publicly available in the UCI data repository. To assess the classification performance, the classification accuracy is used and compared with the results of GA-NB and PSO-NB. As Table (7) shown, IWO-NB acquires the best accuracy. Obviously, the classification accuracy of all datasets in IWO-NB is better than GA-NB and PSO-NB, and the classification accuracy in PSO-NB is better than GA-NB. FS is one of the key factors in enhancing the classifier abilities in the classification problem. In this paper three variant metaheuristic algorithms based on NB classifier were proposed.

TABLE 7: RESULTS OF MODELS FOR 13 DIFFERENT UCI DATASETS BASED ON (ACCURACY/FS)

#	dataset	GA-NB	PSO-NB	IWO-NB
1	Balance	86.75/3	87.97/3	90.38/3
2	Cancer	84.69/12	85.02/12	86.49/12
3	Cancer-Int	83.92/6	81.63/6	87.66/6
4	Credit	86.06/10	87.91/10	91.52/10
5	Dermatology	80.95/12	78.90/12	82.97/12
6	Diabetes	79.86/5	78.52/5	84.39/5
7	E.Coli	83.79/4	84.98/4	85.19/4
8	Glass	80.97/6	79.36/6	82.05/6
9	Heart	82.95/10	81.89/10	85.24/10
10	Horse	70.75/12	69.74/12	70.98/12
11	Iris	96.99/3	97.82/3	98.95/3
12	Thyroid	89.41/4	91.53/4	93.64/4
13	Wine	90.16/8	92.81/8	95.98/8

We have also compared the error rate of different models using the training and testing samples of each dataset. When we compare the models according to Table (8), we observe that the worst case performance of error rate is belonging to GA-NB. The IWO-NB is successful on almost all datasets except the one dataset "Horse" in terms of error rate value.

TABLE8: ERROR RATE ON TRAINING AND TESTING SETS WITH EACH MODEL

#	dataset	GA-NB	PSO-NB	IWO-NB
1	Balance	13.25	12.03	9.62
2	Cancer	15.31	14.98	13.51
3	Cancer-Int	16.08	18.37	12.34
4	Credit	13.94	12.09	8.48
5	Dermatology	19.05	21.10	17.03
6	Diabetes	20.14	21.48	15.61
7	E.Coli	16.21	15.02	14.81
8	Glass	19.03	20.64	17.95
9	Heart	17.05	18.11	14.76
10	Horse	27.25	30.26	29.02
11	Iris	3.01	2.18	1.05
12	Thyroid	10.59	8.47	6.36
13	Wine	9.84	7.19	4.02

Analyzing the error rate shown in Figure (3), the IWO-NB obtained a lower error value than the GA-NB, and PSO-NB.

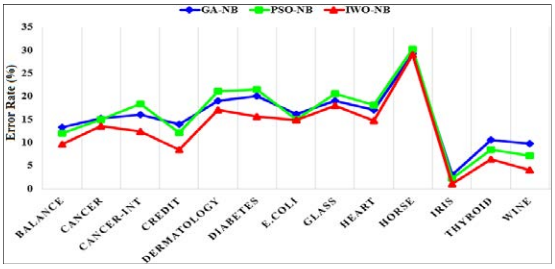


Fig.3. The comparison of the percentage of the error rate for each model

All these models were executed on the same machine with configurations: Intel (R) Core (TM) I7-4510U CPU, 6 GB RAM and Windows 8.1 Operating System. All models use the same parameter settings and are tested on the same datasets, so we used the computational time to compare between the performances of the proposed models. Table (9) presents the computational time (in seconds) required by each model to give near optimal solution.

TABLE9: COMPARISON OF MODELS BASED ON EXECUTE TIME

#	dataset	GA-NB (Seconds)	PSO-NB (Seconds)	IWO-NB (Seconds)
1	Balance	10.30	11.26	6.20
2	Cancer	15.32	12.90	9.01
3	Cancer-Int	14.94	11.05	15.47
4	Credit	11.03	9.32	8.21
5	Dermatology	16.58	14.95	13.79
6	Diabetes	13.32	12.36	10.02
7	E.Coli	10.84	12.26	11.00
8	Glass	12.94	13.08	7.09
9	Heart	13.02	15.79	9.37
10	Horse	15.62	7.11	9.74
11	Iris	3.05	2.49	1.25
12	Thyroid	6.98	5.16	2.67
13	Wine	4.60	3.07	2.14

Experimental results in Table (9) show that, computational time of IWO-NB is shorter than that of GA-NB and PSO-NB. IWO-NB can get a FS in very short time when dealing with the relatively large-scale datasets. Due to GA, PSO and IWO are based on meta-heuristic technique; their results can be different in different runs. Figure (4) shows results obtained based on computational time. Regarding the running times of the models, the best performance is obtained by IWO-NB and the worst performance is obtained by GA-NB. IWO has vigorous exploration ability; it is a gradual searching process that approaches optimal solutions. The execute time of IWO is affected more by the problem dimension (feature numbers), and the size of data. For some datasets with more features, after finding a sub-optimal solution, the GA cannot find a better one. However, IWO can search in the feature space until the optimal solution is found. The GA is affected greatly by the number of features.

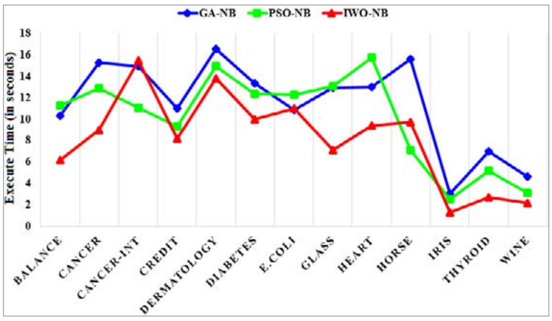


Fig. 4. Chart of Comparison of Models based on Execute Time

V. RESULTS AND ASSESSMENT

In this section the assessment is done and results are presented on Reuters-21578, WebKb, and Cade 12 datasets in VC#.NET 2017 programming language. The primary population and the repetition number in IWO algorithm are 50 and 100 respectively. For showing the efficiency of the proposed model, the dataset was performed in NB classifier first. All experiments are conducted on three different benchmark datasets Reuters-21578, WebKb, and Cade 12. These datasets are pre-classified into several categories. The Reuters-21578 dataset is a standard and widely distributed collection of news published by Reuter's newswire in 1987. It consists of 21,578 documents, which are distributed non-uniformly over 135 thematic categories. The WebKB dataset is prepared by Craven in 1998. It contains 8,282 web pages gathered from the four academic domains. The original dataset has seven categories, but only four of them course, faculty, project and student are used. The cade 12 consists of 40983 documents. The documents in the Cade12 correspond to a subset of web pages extracted from the CADE Web Directory, which points to Brazilian web pages classified by human experts.

5.1. Naive Bayes Classifier

In Table (10), the results of the datasets according to NB classifier are shown. The values of the factor accuracy in Reuters-21578, WebKb, and Cade 12 are 0.7012, 0.7265, and 0.7045 respectively. The dataset WebKb has the highest accuracy.

TABLE10:mRESULTS OF THE DATASETS ACCORDING TO NB CLASSIFIER

Criteria	Reuters-21578	WebKb	Cade 12
Precision	0.6632	0.6548	0.6984
Recall	0.6925	0.7136	0.7214
F-Measure	0.6775	0.6829	0.7097
AUC	0.6894	0.6914	0.7058
Accuracy	0.7012	0.7265	0.7045
Error Rate	0.2988	0.2735	0.2955

5.2. GA Results in FS

In Table (11), the results of the GA model are shown in Reuters-21578 with the selection of different attributes. As you can see in Table (11), the Accuracy criterion value with 160 features is 0.7684.

TABLE11: THE RESULTS OF THE GA MODEL WITH A FS ON REUTERS-21578

Number of Features	Reuters-21578					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7051	0.7185	0.7117	0.7115	0.7634	0.2366
40	0.7161	0.7187	0.7174	0.7268	0.7521	0.2479
60	0.7030	0.7064	0.7047	0.7137	0.7361	0.2639
80	0.6984	0.7021	0.7002	0.7095	0.7297	0.2703
100	0.7031	0.7186	0.7108	0.6941	0.7164	0.2836
120	0.7263	0.7290	0.7276	0.7153	0.7038	0.2962
140	0.7111	0.7174	0.7142	0.7084	0.7519	0.2481
160	0.7052	0.7132	0.7092	0.7134	0.7684	0.2316

In Table (12), the results of the GA model are shown with the choice of different properties on WebKb. In Table (12), the Accuracy criterion value with 60 features is 0.8197.

TABLE 12: GA RESULTS WITH THE SELECTION OF DIFFERENT FEATURES ON WEBKB

Number of Features	WebKb					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7361	0.7612	0.7484	0.7861	0.7928	0.2072
40	0.7219	0.7491	0.7352	0.7926	0.8034	0.1966
60	0.7238	0.7502	0.7368	0.7504	0.8197	0.1803
80	0.7317	0.7410	0.7363	0.7315	0.7837	0.2163
100	0.7218	0.7316	0.7267	0.7496	0.8054	0.1946
120	0.7184	0.7208	0.7196	0.7318	0.8184	0.1816
140	0.7216	0.7523	0.7366	0.7684	0.7640	0.2360
160	0.7655	0.7848	0.7750	0.7710	0.7514	0.2486

In Table (13), the results of the GA model are shown in Cade 12 by selecting different attributes. In Table (13), the Accuracy criterion value for the 120 features is 0.8601.

TABLE 13: GA RESULTS WITH A SELECTION OF DIFFERENT FEATURES ON CADE 12

Number	Cade 12					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7216	0.7312	0.7264	0.8030	0.8415	0.1585
40	0.7594	0.7604	0.7599	0.8109	0.8398	0.1602
60	0.7361	0.7519	0.7439	0.7914	0.8591	0.1409
80	0.7130	0.7200	0.7165	0.8160	0.8425	0.1575
100	0.7284	0.7351	0.7317	0.8055	0.8394	0.1606
120	0.7310	0.7468	0.7388	0.7904	0.8601	0.1399
140	0.7419	0.7502	0.7460	0.7814	0.8317	0.1683
160	0.7158	0.7218	0.7188	0.7618	0.8091	0.1909

5.3. PSO Results in FS

In Table (14), the results of the PSO model are shown by selecting different features on Reuters-21578. In Table (14), the Accuracy criterion value with 20 features is 0.8537.

TABLE14:
RESULTS OF THE PSO MODEL BY SELECTING DIFFERENT FEATURES ON REUTERS-21578

Number of Features	Reuters-21578					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7305	0.7511	0.7404	0.8431	0.8537	0.1463
40	0.7498	0.7621	0.7559	0.8219	0.8452	0.1548
60	0.7516	0.7598	0.7557	0.8375	0.8319	0.1681
80	0.7462	0.7501	0.7481	0.8207	0.8479	0.1521
100	0.7319	0.7416	0.7367	0.8100	0.8314	0.1686
120	0.7407	0.7589	0.7497	0.8209	0.8274	0.1726
140	0.7265	0.7315	0.7290	0.8031	0.8130	0.1870
160	0.7182	0.7197	0.7189	0.8090	0.8049	0.1951

In Table (15), the results of the PSO model are shown by selecting different features on WebKb. In Table (15), the Accuracy criterion value with 40 features is 0.8920.

TABLE15: RESULTS OF THE PSO MODEL BY SELECTING DIFFERENT FEATURES ON WEBKB

Number of Features	WebKb					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7921	0.8079	0.7999	0.8614	0.8726	0.1274
40	0.7658	0.8814	0.8195	0.8421	0.8920	0.1080
60	0.7516	0.7690	0.7602	0.8313	0.8621	0.1379
80	0.7468	0.7513	0.7490	0.8394	0.8549	0.1451
100	0.7196	0.7256	0.7230	0.8273	0.8421	0.1579
120	0.7354	0.7408	0.7381	0.8165	0.8219	0.1781
140	0.7200	0.7311	0.7255	0.8098	0.7930	0.2070
160	0.7134	0.7264	0.7198	0.8006	0.7902	0.2098

In Table (16), the results of the PSO model are shown in Cade 12 by selecting different attributes. In Table (16), the Accuracy criterion value for the 60 features is 0.8968.

TABLE 16: RESULTS OF THE PSO MODEL BY SELECTING DIFFERENT FEATURES ON CADE 12

Number of Features	Cade 12					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.8016	0.8273	0.8142	0.8517	0.8621	0.1379
40	0.7916	0.8068	0.7991	0.8430	0.8814	0.1186
60	0.7810	0.7916	0.7863	0.8491	0.8968	0.1032
80	0.7513	0.7741	0.7625	0.8352	0.8632	0.1368
100	0.7494	0.7561	0.7527	0.8264	0.8547	0.1453
120	0.7261	0.7320	0.7290	0.8139	0.8321	0.1679
140	0.7289	0.7464	0.7375	0.8047	0.8247	0.1753
160	0.7135	0.7231	0.7183	0.8013	0.8194	0.1806

5.4. Proposed Model without FS

In Table (17), the results of the datasets according to the proposed model without FS are shown. The values of the factor accuracy in Reuters-21578, WebKb, and Cade 12 are 0.7625, 0.7258, and 0.7414 respectively.

TABLE17:
RESULTS OF THE DATASETS ACCORDING TO PROPOSED MODEL WITHOUT FS

Criteria	Reuters-21578	WebKb	Cade 12
Precision	0.7165	0.7041	0.7213
Recall	0.7536	0.7489	0.7626
F-Measure	0.7346	0.7258	0.7414
AUC	0.7462	0.7524	0.7319
Accuracy	0.7625	0.7803	0.7511
Error Rate	0.2375	0.2197	0.2489

5.5. Proposed Model with FS

In Table (18), the results of the proposed model with various FS are presented on Reuters-21578. In Table (18), with 140 features accuracy factor is 0.9687.

TABLE 18: RESULTS OF THE PROPOSED MODEL WITH VARIOUS FS ON REUTERS-21578

Number of Feature	Reuters-21578					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7536	0.7625	0.7580	0.7388	0.9134	0.0866
40	0.7698	0.7848	0.7772	0.7695	0.9015	0.0985
60	0.7914	0.7956	0.7935	0.7157	0.9189	0.0811
80	0.8015	0.8365	0.8186	0.7934	0.9347	0.0653
100	0.7814	0.8246	0.8024	0.8012	0.9208	0.0792
120	0.8145	0.8268	0.8206	0.8067	0.9344	0.0656
140	0.8235	0.8469	0.8350	0.7936	0.9687	0.0313
160	0.8411	0.8698	0.8552	0.8942	0.9541	0.0459

In Table (19), the results of the proposed model with various FS are presented on WebKb. In Table (19), with 160 features accuracy factor is 0.9647.

TABLE 19: RESULTS OF THE PROPOSED MODEL WITH VARIOUS FS ON WEBKB

Number of Feature	WebKb					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7612	0.7935	0.7770	0.7832	0.9148	0.0852
40	0.7736	0.7956	0.7844	0.7930	0.9016	0.0984
60	0.7964	0.8011	0.7987	0.8044	0.9102	0.0898
80	0.8012	0.8170	0.8090	0.7615	0.9532	0.0468
100	0.8165	0.8295	0.8229	0.7710	0.9484	0.0516
120	0.8034	0.8314	0.8172	0.7887	0.9325	0.0675
140	0.8295	0.8617	0.8453	0.8055	0.9415	0.0585
160	0.8041	0.8915	0.8455	0.8347	0.9647	0.0353

In Table (20), the results of the proposed model with various features selection are presented on Cade 12. In Table (20), with 100 features accuracy factor is 0.9614.

TABLE 20: RESULTS OF THE PROPOSED MODEL WITH VARIOUS FS ON CADE 12

Number of Feature	Cade 12					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
20	0.7912	0.8154	0.8031	0.8236	0.9061	0.0939
40	0.7629	0.8050	0.7834	0.8023	0.9210	0.079
60	0.7813	0.8197	0.8000	0.7561	0.9412	0.0588
80	0.7901	0.8047	0.7973	0.7987	0.9328	0.0672
100	0.8036	0.8264	0.8148	0.8047	0.9614	0.0386
120	0.8149	0.8320	0.8234	0.8255	0.9431	0.0569
140	0.8190	0.8497	0.8341	0.7903	0.9547	0.0453
160	0.8254	0.8531	0.8390	0.8037	0.9580	0.042

In Table (21), results of the proposed model with various FS are shown for the datasets based on error rate factor.

TABLE 21: RESULTS OF THE PROPOSED MODEL WITH VARIOUS FS ACCORDING TO ERROR RATE FACTOR

Number of Feature	Error Rate		
	Reuters-21578	WebKb	Cade 12
20	0.0866	0.0852	0.0939
40	0.0985	0.0984	0.079
60	0.0811	0.0898	0.0588
80	0.0653	0.0468	0.0672
100	0.0792	0.0516	0.0386
120	0.0656	0.0675	0.0569
140	0.0313	0.0585	0.0453
160	0.0459	0.0353	0.042

In Figure (5), diagram of comparison of the error rate factor on Reuters-21578 according to FS is shown. In Figure (5), it is easily seen that with 140 features, the lowest error rate on Reuters-21578 is that of the proposed model.

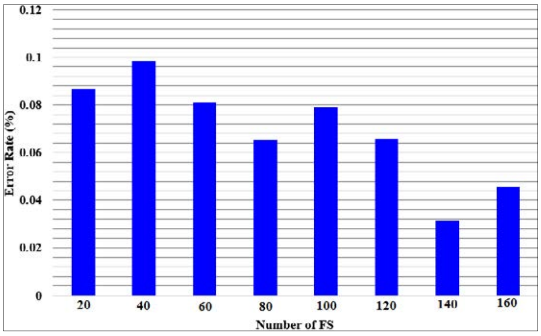


Fig. 5. Comparison Diagram of Error Rate based on FS on Reuters-21578

In Figure (6), diagram of comparison of the error rate factor on WebKb according to FS is shown. In Figure (6), it is easily seen that with 160 features, the lowest error rate on WebKb is that of the proposed model.

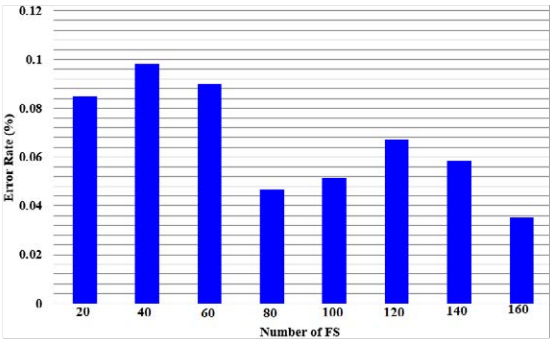


Fig.6. Comparison Diagram of Error Rate based on FS on WebKb

In Figure (7), diagram of comparison of the error rate factor on Cade 12 according to FS is shown. In Figure (7), it is easily seen with 100 features, that the lowest error rate on Cade 12 is that of the proposed model.

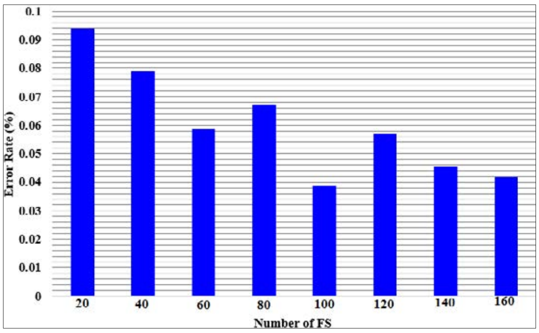


Fig.7. Comparison Diagram of Error Rate based on FS on Cade 12

In Table (22), comparison of the NB classifier, the proposed model without FS, and the proposed model with FS are shown according to error rate factor.

TABLE22: COMPARISON OF MODELS ACCORDING TO ERROR RATE

Models	Error Rate		
	Reuters-21578	WebKb	Cade 12
NB	0.2988	0.2735	0.2955
Proposed Model without FS	0.2375	0.2197	0.2489
Proposed Model with FS	0.0313	0.0353	0.0386

In Figure (8), comparison of the NB classifier, the proposed model without FS, and the proposed model with FS are shown according to error rate factor.

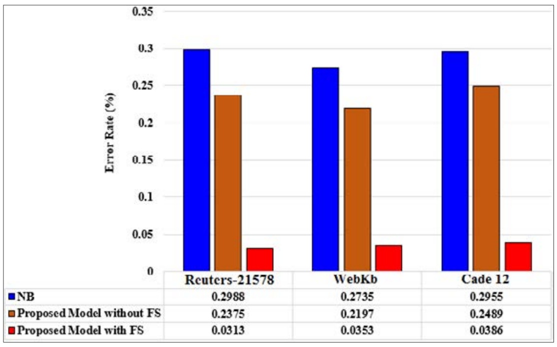


Fig.8. Comparison Diagram of Models According to Error Rate

5.6. Comparison and Assessment

In this section, the results of the proposed model are compared with ML techniques on the datasets Reuters-21578, WebKb, and Cade 12.

5.6.1. Machine Learning Models

In Table (19), comparison of the proposed model and different ML techniques is presented [32]. ML techniques are often applied in TDC applications to reduce human effort and can be divided into two primary types: supervised and unsupervised. The main difference between the two types is that unsupervised ML-based TDC does not require a training process for learning how to classify text into proper categories, whereas supervised ML based TC needs a gold standard for training the classifier. Different algorithms have been used for supervised ML-based TDC, such as NB, KNN, and SVM.

Table (19) suggests that in comparison with other models, the proposed model has a higher accuracy than ML techniques; that Bagging+ RF has the highest F-Measure, and that the models Bagging+ RF and RS+RF have the highest AUC. In the experimental analysis, five statistical keyword extraction methods are taken into account. These methods include most frequent based keyword extraction, term frequency-inverse sentence frequency (TF-ISF) [32] based keyword extraction, co-occurrence statistical information based keyword extraction (CSI) [32], eccentricity-based keyword extraction (EB) [32] and Text Rank algorithm based keyword extraction (TR).

Bagging [32] is one of classes of machine learning which helps to build a strong/improved composite classifier with high predictive efficiency by combining the classifiers trained on different training sets. In this method, each weak learning algorithm is trained on a different training set obtained by a substitution from the training set, where sizes of samples are kept equal to the size of the main training set. For obtain new training sets, the simple random sampling with substitution is utilized. This method yields the diversity required for the ensemble learning. The results of the individual classifiers are combined by majority voting or weighted majority voting. Voting [32] is the simplest form of combining the base learning algorithms. There are several ways to combine the outputs of base classification algorithms. These fusion methods include majority voting, weighted majority voting, NB hybrid rule, behavioral knowledge space method, and probabilistic approximation. In the simple majority voting, the binary outputs of the k base classification algorithms are combined such that the class with the highest number of votes is determined as the output of the ensemble.

TABLE 23
COMPARISON OF THE PROPOSED MODEL WITH ML MODELS ON REUTERS-21578

Accuracy					
Algorithm [32]	CSI	EB	MF	TS-ISF	TR
NB	71.08	82.38	83.70	76.19	82.62
SVM	64.70	74.30	78.53	70.78	77.98
LR	66.70	76.94	76.24	72.71	78.79
RF	70.57	79.94	81.73	75.42	79.78
Bagging+ RF	73.34	87.37	91.49	82.39	88.96
Random Subspace+ RF	72.67	85.77	91.42	82.19	88.63
Majority Voting	72.94	83.18	86.83	76.90	82.64
Proposed Model	95.41				
F-Measure					
Algorithm [32]	CSI	EB	MF	TS-ISF	TR
NB	68.00	83.00	87.00	76.00	81.00
SVM	65.00	79.00	85.00	71.00	76.00
LR	65.00	79.00	84.00	73.00	79.00
RF	67.00	81.00	87.00	75.00	80.00
Bagging+ RF	71.00	87.00	92.00	82.00	89.00
Random Subspace+ RF	70.00	85.00	91.00	81.00	88.00
Majority Voting	70.00	83.00	87.00	75.00	82.00
Proposed Model	85.52				
AUC					
Algorithm [32]	CSI	EB	MF	TS-ISF	TR
NB	79.00	87.00	90.00	83.00	85.00
SVM	75.00	81.00	88.00	78.00	80.00
LR	78.00	82.00	87.00	80.00	81.00
RF	79.00	86.00	89.00	82.00	85.00
Bagging+ RF	95.00	98.00	99.00	98.00	99.00
Random Subspace+ RF	95.00	97.00	99.00	98.00	99.00
Majority Voting	79.00	87.00	90.00	83.00	87.00
Proposed Model	89.42				

The NB showed the better performance than SVM and LR in the experiments and the accuracy of the proposed model is equal to %95.41. The NB classifier has several strong points related to its simplicity and demand for small amount of training data. NB is one of simplest techniques that construct classifiers based on the basic and strong probability theory. Despite its naive design and assumption, NB classifiers have worked quite well in many complex real-world situations.

5.6.2. NB-K-Means Model

In Table (24), the results of the proposed model are compared with NB-K-Means model [16] on the datasets Reuters-21578, WebKb, and Cade 12 according to accuracy factor. Table (24) shows that the proposed model has a higher accuracy and that is due to its selection of effective features in classification.

TABLE24: COMPARISON OF THE PROPOSED MODEL WITH NB-K-MEANS MODEL

Models	Reuters-21578 R8	Reuters-21578 R52	WebKb	Cade 12
NB-K-Means [16]	91.60	88.50	94.80	88.10
Proposed Model	93.40	96.87	96.47	96.14

Table (25) shows the F-Measure scores that were obtained on Reuters-21578 dataset with SVM and NB classifiers [33]. According to Table (25), IGFSS method surpasses the individual performances of three different global feature selection methods in terms of Accuracy.

TABLE 25
F-MEASURE SCORES (%) FOR REUTERS DATASET USING (A) SVM (B) NB [33]

(a)	F-Measure (%)					
Models [33]	250	300	350	400	450	500
IG	85.75	86.00	86.00	85.86	86.00	85.82
IG+IGFSS	85.36	86.47	86.15	86.29	86.11	86.00
GI	85.93	85.97	86.00	86.40	86.07	86.43
GI+IGFSS	85.64	85.79	86.32	86.43	86.76	85.93
Distinguishing Feature Selector (DFS) [35]	85.89	85.89	85.97	85.79	85.89	85.79
DFS+IGFSS	85.00	86.25	86.47	86.25	86.11	85.86
(b)	F-Measure (%)					
Models [33]	250	300	350	400	450	500
IG	83.53	82.38	82.38	82.56	81.91	81.73
IG+IGFSS	84.10	84.28	84.32	84.21	84.53	84.03
GI	84.53	84.21	83.96	84.14	83.67	83.42
GI+IGFSS	85.10	85.46	84.82	84.96	84.35	84.57
DFS	84.93	84.28	84.03	83.88	83.60	83.10
DFS+IGFSS	84.60	85.18	85.28	84.67	84.78	87.75
Proposed Model	86.20	84.96	85.09	84.049	85.13	88.03

Table (26) shows the F-Measure scores that were obtained on WebKb dataset with SVM and NB classifiers [33].

TABLE 26
F-MEASURE SCORES (%) FOR WEBKB DATASET USING (A) SVM (B) NB [33]

(a)	F-Measure (%)					
Models [33]	250	300	350	400	450	500
IG	82.01	81.69	82.01	80.90	81.61	81.06
IG+IGFSS	83.59	83.91	81.93	81.85	81.69	79.79
GI	81.22	81.53	81.30	83.43	82.56	82.96
GI+IGFSS	84.31	83.04	82.01	82.56	82.72	81.69
DFS	83.75	83.67	82.40	81.37	80.56	79.95
DFS+IGFSS	84.54	82.72	82.25	81.69	81.06	80.58
(b)	F-Measure (%)					
Models [33]	250	300	350	400	450	500
IG	81.06	81.22	80.98	80.34	79.95	79.23
IG+IGFSS	83.12	83.51	83.04	82.64	81.45	80.98
GI	57.76	61.25	64.89	69.01	70.91	72.58
GI+IGFSS	78.13	77.65	77.73	77.33	76.94	76.62
DFS	82.64	81.61	82.25	81.85	80.74	80.66
DFS+IGFSS	84.70	83.36	82.96	82.56	83.04	82.56
Proposed Model	83.59	84.35	83.59	85.16	83.07	82.31

Information Gain (IG) scores show the contribution ratio of the presence or absence of a term to correct classification of text documents. IG assigns a maximum value to a term if it is a good indicator for assigning the document to any class. As it is indicated Equation (14), IG is a global FS metric as producing only one score for any term t and this score is calculated according to Equation (14) [33].

$$IG(t) = -\sum_{i=1}^{M} \begin{matrix} P(C_i)\log P(C_i) + \\ P(t)\sum_{i=1}^{M} P(C_i \mid t)\log P(C_i \mid t) + P(\bar{t})\sum_{i=1}^{M} P(C_i \mid \bar{t})\log P(C_i \mid \bar{t}) \end{matrix} \tag{14}$$

In Equation (14), P(Ci) express the probability of class Ci, M is the number of classes, P(t) and P(t̄) are the probabilities of presence and absence of term t, P(Ci|t) and P(Ci|t̄) are the conditional probabilities of class Ci given presence and absence of term t, respectively. Gini index (GI) is a global FS method for TDC which can be used as an improved type of an FS algorithm used in decision tree construction. It has a simple formulation which is defined by the following equation (15) [33].

$$GI(t) = \sum_{i=1}^{M} P(C_i \mid t)^2 P(C_i \mid t)^2 \tag{15}$$

In Equation (15), P(t|Ci) is the probability of term t given presence of class Ci, P(Ci|t)is the probability of class Ci given presence of term t, respectively. DFS is one of the most efficient FS algorithms for TDC and is also a global FS metric. The idea behind DFS is to select distinctive features while eliminating uninformative ones considering some predetermined criteria. DFS is defined according to Equation (16) [33].

$$DFS(t) = \sum_{i=1}^{M} \frac{P(C_i \mid t)}{P(\bar{t} \mid C_i) + P(t \mid \overline{C}_i) + 1} \tag{16}$$

In Equation (16), M is the number of classes, P(Ci|t) is the conditional probability of class Ci given presence of term t, P(t|Ci) is the conditional probability of absence of term t given class Ci, and P(t|C'i) is the conditional probability of term t given all the classes except Ci.

IG [34] is one of the popular approaches employed as a term importance criterion in the text document data. The idea is based on information theory. Before dimension reduction, each term within the text is ranked depending on their importance for the classification in decreasing order using the IG method. The experimental results with the KNN and C4.5 decision tree classifier are summarized in Table (27).

TABLE27: THE COMPARISON OF KNN AND C4.5 WITH PROPOSED MODEL BASED ON IG ON REUTERS-21,578

KNN [34]			
Number of Features	Precision	Recall	F-Measure
75	95.14	94.71	94.93
151	94.26	97.38	95.80
226	94.03	97.62	95.79
302	94.87	97.86	96.34
377	94.04	97.73	95.85
453	93.33	97.54	95.39
528	91.74	97.86	94.31
603	91.07	97.78	94.31
679	90.63	97.92	94.13
754	90.14	97.65	93.74
C4.5 [34]			
Number of Features	Precision	Recall	F-Measure
75	94.50	94.63	94.57
151	94.82	94.84	94.83
226	94.86	94.63	94.74
302	95.48	95.38	95.43
377	94.83	96.02	95.42
453	95.61	95.40	95.51
528	95.21	94.47	94.84
603	95.32	95.27	95.30
679	95.24	94.52	94.88
754	95.18	95.43	95.30
Proposed Model			
Number of Features	Precision	Recall	F-Measure
75	96.21	96.52	96.12
151	95.89	96.18	95.82
226	95.73	96.26	96.03
302	94.60	95.43	94.98
377	93.52	94.86	93.72
453	93.91	95.26	94.35
528	93.15	94.36	93.64
603	92.66	93.07	92.91
679	91.67	92.17	92.03
754	91.21	92.15	91.79

TABLE 28: THE COMPARISON OF KNN AND C4.5 WITH PROPOSED MODEL BASED ON IG-GA ON REUTERS-21,578

KNN [34]			
Number of Features	Precision	Recall	F-Measure
42	95.37	94.68	95.03
83	96.64	95.99	96.31
121	97.50	96.93	97.21
169	98.17	97.52	97.84
197	97.73	97.60	97.66
241	97.42	97.73	97.57
286	97.16	97.84	97.50
317	97.04	98.05	97.54
352	97.04	98.10	97.57
384	96.93	97.78	97.35
C4.5 [34]			
Number of Features	Precision	Recall	F-Measure
45	96.20	93.40	94.78
78	95.98	94.47	95.22
116	95.39	94.60	95.00
175	95.95	95.65	95.80
201	96.41	95.40	95.96
244	96.51	95.40	95.96
281	96.11	95.65	95.88
328	96.40	95.11	95.75
355	95.84	96.08	95.96
380	95.72	95.51	95.61
Proposed Model			
Number of Features	Precision	Recall	F-Measure
45	96.79	97.23	96.64
78	96.35	97.41	96.53
116	96.18	96.92	96.07
175	95.93	96.83	96.14
201	95.81	95.35	95.40
244	96.03	96.91	95.61
281	95.84	93.12	95.55
328	96.07	96.23	96.11
355	95.86	96.48	95.61
380	96.01	96.50	96.21

Table (28) shows the detailed comparison results. From these results, we can see that our proposed model significantly outperform KNN and are even much better than C4.5. The GA is an optimization method mimicking the evolution. This algorithm, which is an effective optimization method in wide search spaces, is preferred because it is the appropriate method for the solution of the problem. Although, terms of high importance in documents are acquired through IG method, but main problem is the high dimensionality of the feature space. Since given a feature set U via IG method is high dimensionality, it is impractical to evaluate all the possible subsets of U. Due to this deficiency GA-based FS method is adopted in [34]. Accordingly, GA is used to provide near-optimal solutions for FS. The objective of the GA-based FS is to find the optimal subset of a given feature set U that maximizes classification performance in [34].

In [35] has been proposed and explored a novel discriminative and semantic FS method for text categorization. The proposed method first selects features with strong discriminative power and then considers the semantic similarity between features and documents. The FS is tested using SVM classifier upon two datasets (Reuters-21578 and 20-Newsgroups [34]). In this type of model, a document is represented as a feature vector whose components are the term weights, dk=(w1k,w2k,…,wik,…,wnk), where wik is the weight of term ti in document dk. In this method, features are selected in documents based on a scale of discriminative power, and also on a measure of the similarity between features and the similarity between features and documents independent of the external information sources. To transform all documents into feature vectors using the selected features, and these vectors form the input data for the SVM. The SVM is used to evaluate the usefulness of the FS method. The comparisons involve five FS methods, which include the χ2 statistic, IG, and mutual information (MI). The other two are incorporated in the proposed method, i.e., the discriminative feature selection method (DFS), and the discriminative and semantic FS method (DFS+ Similarity).

TABLE 29: PERFORMANCE COMPARISON WITH DIFFERENT NUMBER OF FEATURES ON (A) REUTERS-21578 AND (B) 20 NEWSGROUPS [35]

(a)	Reuters-21578, F-Measure (%), FS									
Models [35]	500	1000	1500	2000	2500	3000	3500	400	4500	5000
χ^2 statistic	93.74	94.97	95.36	95.50	95.60	95.60	95.65	95.80	95.70	95.89
DFS	94.62	95.11	95.60	95.60	95.60	95.60	95.70	95.75	95.85	95.85
DFS+ Similarity	95.80	95.94	96.43	96.63	96.48	96.53	96.29	96.53	96.68	96.82
MI	51.71	55.52	57.97	63.59	68.96	73.26	77.86	77.47	85.34	88.86
IG	51.22	51.91	52.30	52.54	77.61	77.71	80.26	82.70	84.95	91.98
Proposed Model	96.15	92.35	91.49	96.47	91.16	83.56	90.27	92.79	89.11	92.84
(b)	20 Newsgroups, F-Measure (%), FS									
Models [35]	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
χ^2 statistic	63.64	69.64	72.84	73.89	74.79	75.04	75.73	75.99	76.07	76.22
DFS	66.31	71.19	73.40	74.68	75.09	75.57	75.88	76.44	76.54	76.60
DFS+ Similarity	70.27	73.70	75.02	76.01	76.74	77.05	77.16	77.83	77.86	78.08
MI	5.72	5.82	6.02	9.40	8.89	11.21	11.60	13.56	17.04	27.58
IG	6.03	6.41	7.28	9.67	11.12	11.77	13.31	17.95	19.47	27.71
Proposed Model	72.61	69.12	76.38	70.10	61.05	67.49	60.48	71.96	72.34	67.13

The Chi Square (χ2) statistic: This method, a feature is selected according to their correlation with a category. The χ2 statistic measures the lack of independence between t and c and can be compared to the χ2 distribution with one degree of freedom to judge extremeness. The statistic is defined in Equation (17) [35].

$$\chi^2(t_i,c_j)=\frac{N.(a_{ij}d_{ij}-b_{ij}c_{ij})^2}{(a_{ij}+b_{ij}).(a_{ij}+c_{ij}).(b_{ij}+d_{ij}).(c_{ij}+d_{ij})} \tag{17}$$

In Equation (17), N is the total number of documents, aij is the number of documents that contain feature ti in category cj, bij is the number of documents that do not contain feature ti in category cj, cij is the number of documents that contain feature ti but do not belong to category cj, and dij is the number of documents that do not contain feature ti and do not belong to category cj. MI is a metric of the correlation between signals, and can be used to identify the features relevant to a particular category, as in Equation (18) [35].

$$MI(t_i) = \sum_{j=1}^{C} P(c_j).\log\frac{P(t_i \mid c_j)}{P(t_i)} \qquad (18)$$

DFS: The main objectives of the DFS method consist of (i) selecting features with a higher average term frequency in cj, because these features have a high probability in representing category cj; (ii) selecting features with a higher occurrence rate in most of the documents in cj, because these features have a high probability in representing category cj; and (iii) ignoring features occurring in most of the documents in cj and cj, because these features have a weak discriminative ability between categories.

$$DFS(t_i, c_j) = \frac{tf(t_i, c_j) / df(t_i, c_j)}{tf(t_i, \bar{c}_j) / df(t_i, \bar{c}_j)}.$$

$$\frac{a_{ij}}{(a_{ij}+b_{ij})}.\frac{a_{ij}}{(a_{ij}+c_{ij})}.\left|\frac{a_{ij}}{(a_{ij}+b_{ij})} - \frac{c_{ij}}{(c_{ij}+d_{ij})}\right| \qquad (19)$$

In Equation (19), and represent the term frequency of feature ti in category cj and in category , respectively, and represent the number of documents containing feature ti in category cj and in category respectively.

TF-IDF: TF-IDF [35] is the most popular term weighting scheme in information retrieval.

$$TF - IDF(w_{m,k}) = \frac{tf(t_m, d_k).\log\left({}^{N}\!/_{n_m} + 0.01\right)}{\sqrt{\sum_{m=1}^{n}\left(tf(t_m, d_k).\log\left({}^{N}\!/_{n_m} + 0.01\right)\right)^2}} \qquad (20)$$

In Equation (20), n is the number of chosen features, tf (tm,dk) is the term frequency of feature tm in document dk, and nm is the number of documents that contain feature tm.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we used a hybrid of IWO algorithm and NB classifier for TDC. We used IWO for selecting important features and NB for document classification based on training and testing. Results indicate that the proposed model is more accurate in comparison with NB classifier. In addition, error rate factor indicates that the errors of the proposed model with FS are less. Comparison of the proposed model with other models indicated that the proposed model is more accurate because of using FS and is able to explore the features space better. The error rate of the proposed model with FS on the datasets Reuters-21578, WebKb, and Cade 12 is 0.0313, 0.0353, and 0.0386 respectively. For future studies, and enhancement of the proposed model one can use a hybrid of the operators of metaheuristic algorithms for selecting the optimum solution.

REFERENCES

1. W. Hadi, Q.A. Al-Radaideh, S. Alhawari, Integrating associative rule-based classification with Naïve Bayes for text classification, Applied Soft Computing, Vol. 69, pp. 344-356, 2018.
2. Mahata, R.R. Shah, J. Kuriakose, R. Zimmermann, J.R. Talburt, Theme-Weighted Ranking of Keywords from Text Documents Using Phrase Embeddings, IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, pp. 184-189, 2018.
3. A. Kulkarni, V. Tokekar, P. Kulkarni, Discovering Context of Labeled Text Documents Using Context Similarity Coefficient, Procedia Computer Science, Vol. 49, pp. 118-127, 2015
4. K. Chen, Z. Zhang, J. Long, H. Zhang, Turning from TF-IDF to TF-IGM for term weighting in text classification, Expert Systems with Applications, Vol. 66, pp. 245-260, 2016.
5. S. Ramanna, J.F. Peters, C. Sengoz, Application of Tolerance Rough Sets in Structured and Unstructured Text Categorization: A Survey, Thriving Rough Sets, Springer, Vol. 708, pp. 119-138, 2017.
6. A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1(4): 355-366, 2006.
7. A. McCallum, K. Nigam, A Comparison of Event Models for Naive Bayes Text Classification, In AAAI-98 workshop on learning for text categorization, Vol. 752, pp. 41-48, 1998.

8. X. Deng, Y. Li, J. Weng, J. Zhang, Feature selection for text classification: A review, Multimedia Tools and Applications, pp. 1-20, 2018.

9. M. Rogati, Y. Yang, High-performing variable selection for text classification, in: CIKM '02 Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 659-661, 2002.

10. Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: The Fourteenth International Conference on Machine Learning (ICML97), pp. 412-420, 1997.

11. J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan, Michigan, USA, 1975.

12. J. Kennedy, R. C. Eberhart, Particle Swarm Optimization, In Proceedings of the IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

13. A. Trstenjak, S. Mikac, D. Donko, KNN with TF-IDF based Framework for Text Categorization, Procedia Engineering, Vol. 69, pp. 1356-1364, 2014.

14. Y. Ko, J. Seo, Text classification from unlabeled documents with bootstrapping and feature projection techniques, Information Processing & Management, Vol. 45, Issue 1, pp. 70-83, 2009

15. D. Ghasempour, F.S.Gharehchopogh, A New Approach for Feature Selection in Text Documents Classification by Using Hybrid Model of Bat and K-Nearest Neighborhood Algorithms, Islamic Azad University, Urmia Branch, Thesis, Summer 2016.

16. A. Allahvirdipour, F.S. Gharehchopogh, New Approach in Features Selection in Text Documents Classification using the Hybrid Model Algorithms of Naive Bayes and K-Means, Islamic Azad University, Urmia Branch, Thesis, Spring 2016.

17. R. Habibpour, K. Khalilpour, A New Hybrid K-means and K-Nearest-Neighbor Algorithms for Text Document Clustering, International Journal of Academic Research, Vol. 6 Issue 3, pp. 79-84, 2014

18. M. Karabulut, Fuzzy unordered rule induction algorithm in text categorization on top of geometric particle swarm optimization term selection, Knowledge-Based Systems, Vol. 54, pp. 288-297, 2013.

19. [19] A.K. Uysal, S. Gunal, Text classification using genetic algorithm oriented latent semantic features, Expert Systems with Applications, Vol. 41, Issue 13, pp. 5938-5947, 2014

20. T. Wei, Y. Lu, H. Chang, Q. Zhou, X. Bao, A semantic approach for text clustering using WordNet and lexical chains, Expert Systems with Applications, Vol. 42, Issue 4, pp. 2264-2275, 2015

21. W. Zhang, X. Tang, T. Yoshida, TESC: An approach to TExt classification using Semi-Supervised Clustering, Knowledge-Based Systems, Vol. 75, pp.152-160, 2015

22. K.K. Bharti, P.K. Singh, Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering, Applied Soft Computing, Vol. 43, pp. 20-34, 2016.

23. D. AbuZeina, F.S. Al-Anzi, Employing fisher discriminant analysis for Arabic text classification, Computers & Electrical Engineering, in press, corrected proof, Available online 10 November 2017.

24. R. Wongso, F.A. Luwinda, B.C. Trisnajaya, O. Rusli, Rudy, News Article Text Classification in Indonesian Language, Procedia Computer Science, Vol. 116, pp. 137-143, 2017.

25. H.P. Luhn, A Statistical Approach to the Mechanized Encoding and Searching of Literary Information, IBM Journal of Research and Development, Vol. 1, No. 4, pp. 309-317, 1957.

26. G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.

27. R.S. Michalski, I. Bratko, M. Kubat, Machine Learning and Data Mining: Methods and Applications, New York: Wiley, 1998.

28. D. Francois, Binary classification performances measure cheat sheet, 2009.

29. C. Blake, C.J. Merz, UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/?mlearn/MLRepository.html], University of California. Department of Information and computer science, Irvine, CA, 1998, pp. 55

30. http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

31. http://ana.cachopo.org/datasets-for-single-label-text-categorization

32. A. Onana, S. Korukoglub, H. Bulut, Ensemble of keyword extraction methods and classifiers in text classification, Expert Systems with Applications, Vol. 57, pp. 232-247, 2016.

33. A.K. Uysal, An improved global feature selection scheme for text classification, Expert Systems with Applications, Vol. 43, pp. 82-92, 2016.

34. H. Uguz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowledge-Based Systems, Vol. 24, Issue 7, pp. 1024-1032, 2011.

35. W. Zong, F. Wu, L.K. Chu, D. Sculli, A Discriminative and Semantic Feature Selection Method for Text Categorization, International Journal of Production Economics, Vol. 165, pp. 215-222, 2015.

36. C. Veenhuis, Binary Invasive Weed Optimization, Second World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 449-454, 2010.

37. L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, The Journal of Supercomputing, Vol. 73, Issue 11, pp. 4773-4795, 2017.