The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 1, 2018
Pages 60-72
Regular Paper

An Improvement of Shuffled Frog Leaping Algorithm with a Decision Tree for Feature Selection in Text Document Classification

Mostafa Mahmoudi Farhad Soleimanian Gharehchopogh

Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, IRAN

Abstract

Given the growth of textual documents, the classification of documents is crucial for reducing the complexity of information and easy and quick access to them. Classification is usually carried out through extraction of keywords, sentences, and matching the paragraphs. The major method for finding similarities in the texts is using keywords based on word frequency. The word count is done through various methods such as TF, and then a specific weight is attributed to each word. The main challenge in Text Document Classification (TDC) is to choose the feature. That is the case because Feature Selection (FS) is an effective factor in enhancing the classification accuracy and reduction of calculation time. Hence, in this paper, Shuffled Frog-Leaping Algorithm (SFLA) for FS and ID3 tree for document classification has been used. A problem with SFLA is that it sticks in local optimums; and in the proposed model, a hybrid of the best and the worst situations of the frog is used for enhancement in order to avoid local optimums. The general method in this paper is to enhance SFLA by means of ID3 tree for classification accuracy. The obtained results on Reuters-21578, WebKb, Cade 12, and 20 Newsgroup datasets indicate that the improved proposed model with ID3 tree has a higher accuracy. The results confirm the efficiency of the proposed FS method in improving TDC accuracy.

Keywords: Text Document Classification, Feature Selection, Shuffled Frog Leaping Algorithm, ID3 Tree

1. Introduction

With the increasing volume of information, the process of storing, retrieving, and classifying text documents is of great importance. The documents may include any kind of text (sports, news, trade, science, etc.). The purpose of TDC is helping users with proper storage of information and accessing their intended information in a big bulk of it [1]. Classification of texts means placing a certain text in one or more certain pre-defined classes based on its content [2]. The first step in classification is preprocessing documents. In this step, unnecessary words are deleted; words such as prepositions, many adverbs and adjectives, some verbs etc. whose omission would not negatively affect the general content of the text and would only summarize it [3].

A key step in TDC is extracting the keywords [4]. Because of the number of electronic documents is increasing rapidly, utilizing effective methods for their classification is highly important. Keywords are a set of important words in a document that provides a description of the content of the document and are useful for various purposes. For instance, the keywords are mentioned in the beginning of scientific articles so that the readers would have a clear understanding of the paper. Extracting keywords from the documents is an important operation during processes such as information classification or extraction. Moreover, keywords can be useful for search engines in returning results that are more accurate in a shorter time. Keywords describe the main points of a text; therefore, they can be used as a tool for measuring the similarity of different texts in order to be used in their classification [5]. Generally, keywords are a useful means for searching in a big bulk of documents in a short time. Identifying and manually counting the keywords is a time consuming and difficult process. Therefore, there is a need for an automated process, like weighting methods, to extract keywords from extracted documents and give a certain weight to each word.

FS in a big number of words is a challenging task. In the present paper, SFLA [6] is used for FS. It is an evolutionary algorithm based on the population. This algorithm is fast and it provides us with a global search capability. Using the SFLA, at first, we place the weights in the leap vectors and then calculate the fit of each vector. The vector that has the highest value is chosen and the selected features are entered in ID3 decision tree [7]. ID3 decision tree carries out classification through data training and testing. ID3 uses a fixed number of data to build a decision tree. That tree is used for classification of information and as a result, for decision-making.

The rest structure of this paper is as follows: In Section 2, related studies about TDC are presented. In Section 3, the improved proposed model is described. In Section 4, experimental results are reported and discussed; and the improved proposed model is compared with other models. And eventually, in Section 5, conclusions are reviewed and suggestions are made for future studies.

2. Related Works

Given the increasing volume of text documents, the need to accelerate and improve the accuracy of text document retrieval is necessary. TDC is one of the techniques that is potentially usable for enhancing the extraction and retrieval of document process. In the recent years, several algorithms have been proposed for TDC. We will review some of the proposed models in this section.

A mathematical model for obtaining the keywords frequency in text documents, Reuters-21578, has been proposed [8]. The aim of the mathematical model is counting the number of words in text documents. If the number of words is accurate, classification will be better and the identification accuracy will be higher. The mathematical model defined in Eq. (1) is from TFIDF (Term Frequency & Inverse Document Frequency).

$$w_{ij} = \frac{TFIDF(t_i, d_j)}{\sqrt{\sum_{k=1}^{|T|} (TFIDF(t_k, d_j))^2}} \quad (1)$$

In Eq. (1), the parameter w_{ij} is the weight of the word i in the document j and |T| is the total sum of the words. In addition, the similarity distance of two documents is defined as in Eq. (2).

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|.\|\vec{b}\|} \quad (2)$$

Results show that the proposed mathematical model is more accurate than other models in determining the word frequency in Reuters-21578 dataset.

B-Tree model is recommended for TDC in comparison with Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Naive Bayes (NB) classifier models [9]. In fact, B-Tree is a tree with its roots upside and the leaves downward. In B-Tree model, the weight designated to the leaves of the tree is used for determining the similarity of documents. Results show that the accuracy is higher in the proposed model in comparison with other models. In addition, NB classifier model has a higher accuracy than SVM, KNN. The Fuzzy C-Means (FCM) model is one of the most practical fuzzy models and is used a lot due to its simplicity and small size [10]. In this paper, we have used FCM for finding the centrality and similarity in documents. The results show that the FCM model is more accurate than SVM, KNN and NB classifier models.

As electronic documents, web pages, messages, etc., all of which mainly contain texts, are increasing dramatically, effective organization, retrieval and mining of the massive textual data are becoming more and more important [11]. TF-IGM and RTF-IGM models are used for accuracy in word frequency and weighting the text documents [11]. Assessment is done using the models of SVM and KNN on the datasets TanCorp, Reuters-21578, and 20 Newsgroups. TF-IGM and RTF-IGM models are defined as in Eqs. (3) and (4). It is proved by extensive experiments on public benchmark datasets that TF-IGM is consistently superior to the famous TF-IDF and the state-of-the-art supervised term weighting schemes. Moreover, if the square root of TF is used as the local weighting factor instead of the raw TF, the improved version, RTFIGM, performs best in most cases. TF-IGM is especially suitable for multiclass text classification applications. Nevertheless, the experimental results show that it is also applicable to binary text classification.

$$w(t_k, d) = tf_{kd} \cdot \left(1 + \lambda . \frac{f_{k1}}{\sum_{r=1}^{m} f_{kr} . r} \right) \quad (3)$$

$$w(t_k, d) = \sqrt{tf_{kd}} \cdot \left(1 + \lambda . \frac{f_{k1}}{\sum_{r=1}^{m} f_{kr} . r} \right) \quad (4)$$

TF-IGM and RTF-IGM models have been proposed for the accuracy of word and weight frequency in textual documents [11]. It offers the following features:

(1) TF-IGM adopts a new statistical model called IGM (inverse gravity moment) to characterize the inter-class distribution and measure the class distinguishing power of a term in the corpus so that terms with stronger class distinguishing power are to be assigned greater weights than others in text representation.

(2) TF-IGM takes into account the fine-grained inter-class distribution of a term across different classes of text so that the calculated weight can reflect the term's importance in text classification more realistically than the other schemes and the classification accuracy is thus improved.

(3) TF-IGM is adaptive to different text corpora or text classification tasks by providing several options or adjustable parameters so as to obtain the optimal performance.

Data mining models that use TF-ISF are recommended for TDC [12]. The purpose of TS-ISF is extracting the keywords. The assessment was done on Reuters-21578 and ACM datasets. Results show that TF-ISF model is more accurate than CSI model in identification. Particle Swarm Optimization (PSO), KNN, and NB classifier models are recommended for TDC [13]. PSO algorithm is used for FS. Assessment is done on the datasets Reuters-21578 and Classic3. Classic3 dataset is comprised of documents gathered from Cornel university

projects. It includes 3891 documents and three classes. The precision of KNN model with 273 features is 96.36.

A hybrid model of Invasive Weed Optimization (IWO) and Naive Bayes (NB) classifier (IWO-NB) is proposed for TDC [14]. In the hybrid model, IWO is used for FS and NB for document similarity. In the hybrid model, at first, the documents are preprocessed ad the keyword are extracted. Then based on the frequency, a specific weight is attributed to them. Assessment is done on Reuters-21578, WebKb, and CADE 12 datasets. Comparisons show that the hybrid model is more accurate than the models Genetic Algorithm (GA) and PSO have been used as comparison models. The hybrid model K-Means-KNN [15] is proposed for text document clustering. In this model, KNN algorithm is used for accuracy in identification of similar clusters. Results on Reuters-21578 dataset indicate that the hybrid model is more accurate in comparison with K-Means model. The hybrid model NB-K-Means was tested on Reuters-21578, WebKb, and CADE 12 datasets for TDC. Results indicate that the hybrid model NB-K-Means is more accurate than K-Means model. Furthermore, the most accuracy in the hybrid model is of k=3 which is 93.30 KNN-K-Means hybrid model has been proposed for clustering text documents [16]. In this model, KNN is used to identify similar clusters. Results on Reuters-21578 dataset show that the hybrid model is more accurate in comparison with K-Means model. It has been described a novel approach to term-weighting scheme (TWS) learning in text classification (TC) [17]. TWSs specify the way in which documents are represented under a vector space model. It has been proposed a Genetic Programming (GP) solution in which standard TWSs, term-document, and term relevance weights are combined to give rise to effective TWSs. For experimentation, it has been considered a suite of benchmark data sets associated to three types of tasks: thematic TC, authorship attribution (AA, a non-thematic TC task) and image classification (IC).

The performance of the proposed method is evaluated under different scenarios. The performance of the proposed model has been evaluated under different settings and the characteristics of the learned TWSs have been analyzed. Maximum and average fitness value obtained by the GP during the search process for 20-Newsgroup, Reuters-8 and Caltech-tiny datasets are about 80%.

3. Proposed Model

For creating the solution space in SFLA we must convert the text documents to vector space; because the documents are in word form and not suitable for vector space. In vector space, numbers need to be integers or decimals. Figure (1) shows the flowchart of the proposed model.

At the beginning, text data is read and preprocessing is done. Preprocessing includes omission of irrelevant or unnecessary words. In the next step, word frequency is determined to identify the type of the document. Usually, every document contains words that play a key role in identifying and classifying and their extraction is a significant factor in the final classification. For word frequency, Eq. (5) is used [18]. Eq. (5) is a word frequency determination method suitable for word extraction due to low time complexion. In Eq. (5), the parameter (t_k , d_i) is the frequency of each t_k feature in the document d_i.

$$w_{ki} = tf(t_k, d_i) = \begin{cases} (t_k, d_i) & t_k \in vector\ of\ d_i \\ 0 & t_k \notin vector\ of\ d_i \end{cases} \quad (5)$$

Once the keywords are extracted, a certain weight in attributed to them based on their frequency. The weight allocated to words is used for forming the vectors. For classification, we use FS, therefore, we must calculate fitness function for each vector. Fit of vectors is based on the weight of the feature. Once the weights of vectors are calculated, we select the vectors with the highest fit value and among the vectors; we select the feature with the highest weights.

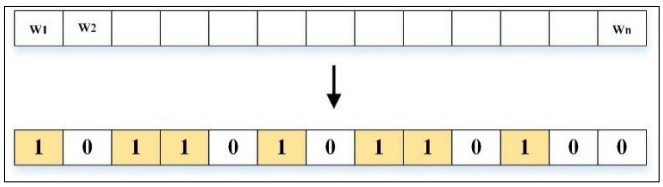

Figure 2. Method of Selecting the Optimum Weights in Solutions' Vectors

For finding the features in the vectors having weights that are the highest, the closest, and the most similar we use Eq. (6); we select them based on the closeness of the features and assume their value as *l*. The calculation of the proximity criterion of vectors values is determined by the cosine distance method in accordance with Eq. (6) [19]. Training vectors are selected according to their similarity with document *x*. Then *k* for the documents with the most similarity is chosen and *d* for the keywords is set according to weight values.

$$sim(x, d_i) = \frac{\sum_{k=1}^{m} x_k \times d_{ik}}{\sqrt{\sum_{k=1}^{m} x_k^2 \sum_{k=1}^{m} d_{ik}^2}} \quad (6)$$

$$p(x, C) = \sum_{d_i} sim(x, d_i)\, y(d_i, C_j) \quad (7)$$

$$y(d_i, c_j) = \begin{cases} 1, d_i & \in C_j \\ 0, d_i & \in C_j \end{cases} \quad (8)$$

A document is represented in the form of a vector D = {xw_1,xw_2,xw_3,…,xw_k}, where xw_i is weight of ith term in a vocabulary containing k number of terms. Membership degree of document *x* to class C_j is denoted by Eq. (7) in which *y(di, ci)* of *x*'s membership in each class is determined by Eq. (8).

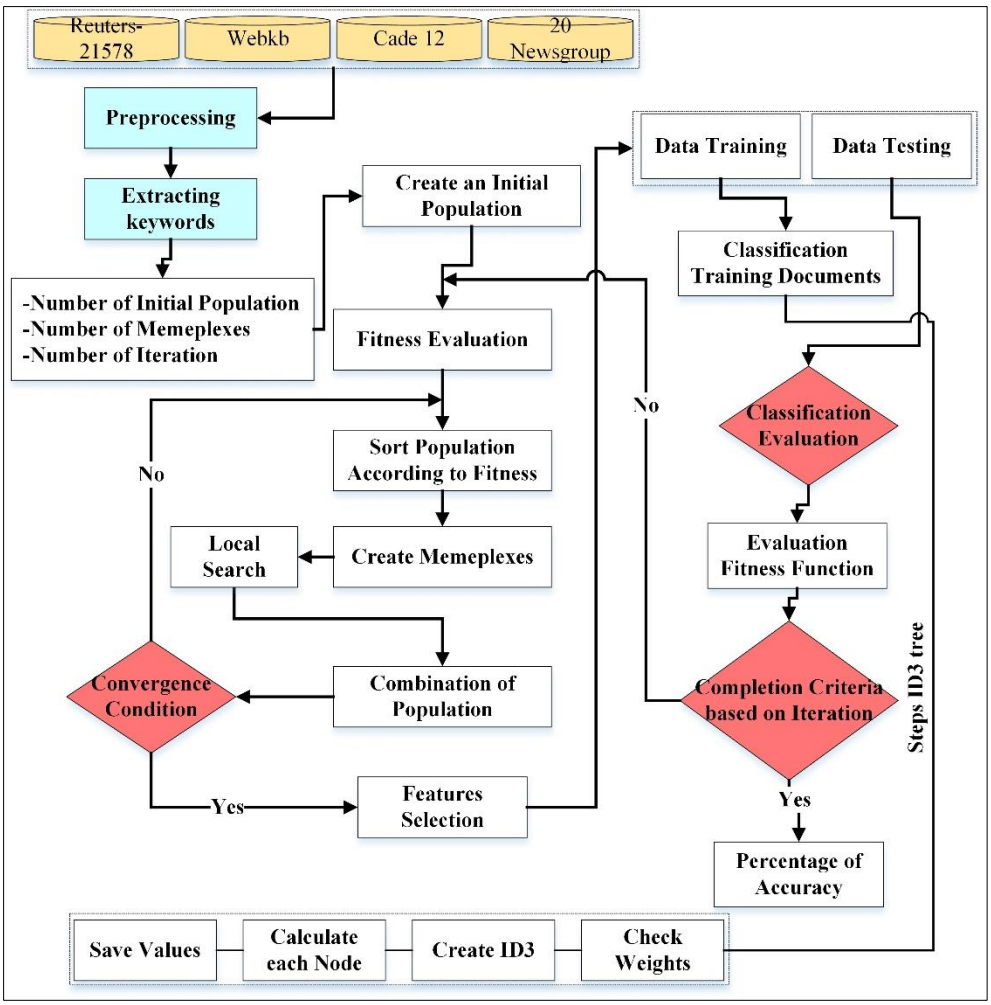

Figure 1. Flowchart of the Proposed Model

3.1. Forming the Vectors

For each vector, the best and the worst frogs are defined as parameters x_b and x_w; and the best frog in the whole population is pointed out by x_g. The best and the worst frogs are features with the most and the least weights. The worst frog moves towards the optimum answer in each repetition. This change is calculated with Eqs. (9 and10).

$$new\_posiotion(X_w) = current\_position(X_w) + rand(X_b, X_w) \quad (9)$$

$$new\_posiotion(X_w) = current\_position(X_w) + rand(X_g, X_w) \quad (10)$$

In Eq. (9), rand parameter is a random number between zero and one. In case Eq. (9) is unable to enhance the rate of the fit function in comparison with the prior status, updating is carried out according to Eq. (10); the difference is that x_g is used instead of x_b. If no enhancement is made in the fit function, a new frog is renders randomly and replaces the worst situation.

3.2. Improved Proposed Model

SFLA is based on population; and mainly its two significant dimensions should be addressed, namely exploration and efficiency. Exploration addresses the problem's space and efficiency finds the optimum solution around the best answer. The important point for having an effective performance in different optimizing problems is enhancing the balance of exploration and efficiency. In some optimizing problems, SFLA is not able to find the absolute optimum and sticks in the local optimum. In SFLA, the worst solution is determined according to updating rules. The way the frog leaps leads to the frog changing location only on a straight line in comparison with the better frog and results in the environment surrounding the frog with a higher chance for getting to answers of higher fit has not been explored. This causes limitation of the explored local environment and may slow down the convergence procedure and increase local being stuck. For getting over this problem we use Eq. (12).

$$new\_posiotion(X_w) = current\_position(X_w) + rand(X_b, X_w) \quad (11)$$

$$new\_posiotion(X_w) = current\_position(X_w) + \left(\frac{X_b - X_w}{|X_g - X_w|}\right) \quad (12)$$

For getting closer to the optimum answer and circumventing the local optimum, we implement the best and worst frogs. That is, the answer gets closer to the intended value in an optimum and global manner, all the spots are assessed together, and there are more cases of finding the solution.

3.3. Feature Selection

The SFLA based FS is described as follows:
Step 1: Initialize the number of frogs (N_n), Number of memeplexes (N_m), Number of frogs in each memeplex (N_f) such that $N_{pop} = N_m * N_f$
Step 2: Generate N_n sample frogs $F(1), F(2), \dots, F(N_n)$ in the feasible space. The ith frog can be represented as a vector with $F(i) = (F_i^1, F_i^2, \dots, F_i^s)$ that is a candidate solution containing S features.
Step 3: The frogs are then sorted in a descending order, based on their fitness value.
Step 4: The all of population is split into m memeplexes. Each contains n frogs, that is ($N_m * n$). Frog f1 moves towards the memeplex (M_1), following which frog f_2 moves to the memeplex (M_2), the m^{th} frog goes to the m^{th} memeplex and hence frog m + 1 moves back to the memeplex (M1) and so on.
Step 5: Within every memeplex, the best and the worst finesses of the frogs identified as X_b and X_w, respectively. Further, the frog with the overall best fitness is represented as X_g. Further, the frog with the overall best fitness is represented as X_g.
Step 6: the position of a frog that has the worst fitness value is adjusted according to the following Eq. (11) and (12).

3.4. Classification

In the next stage, we form the input data for ID3 tree according to the weights of the vectors. We use the weights as samples and *l* value as class samples. First, the entropy of samples and then the information rate of each weight is calculated. The weight with the highest information rate is chosen as the root node and each document is classified according to the root node.
To discover an optimal method to classify a learning set, the information gain (IG) metric function Gain (S, A) is used to find the most balanced splitting. Entropy is a measure in the information theory, which characterizes the impurities of an arbitrary collection of examples. If the target feature takes on w different values, then the entropy S relative to this w-wise classification is defined as in Eq. (13).

$$\text{Entropy(s)} = \sum_{i=1}^{i=n} - p_i \log_2 p_i \quad (13)$$

In Eq. (13), Pi is the proportion/probability of S belonging to class i. Logarithm is base 2 because entropy is a measure of the expected encoding length measured in bits. Information gain measures the expected reduction in entropy by partitioning the examples according to this feature. The IG (S, A) of feature A, relative to the collection of examples S, is defined as in Eq. (14).

$$\text{Gain(S, A)} = \text{Entropy(s)} - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (14)$$

In Eq. (14), $Values(A)$ is the set of all possible values for feature A, and Sv is the subset of S for which the feature A has value v. This can be used to measure rank features and build the decision tree where each node is located, and the feature with the highest IG among the features not yet considered in the path from the root.

3.5. Evaluation Criteria

The results of the improved proposed model need to be analyzed in assessment stage so that their values can be determined and their effectiveness can be measured. These factors can be calculated both at the learning stage for the training datasets and at the assessment stage for the test record sets. Different criteria may be used in assessment; we use accuracy for this end [20].

$$P = \frac{TPc_i}{TPc_i + FPc_i} \quad (15)$$

$$R = \frac{TPc_i}{TPc_i + FNc_i} \quad (16)$$

$$F - Measure = \frac{2 * P * R}{(P + R)} \quad (17)$$

$$AUC = \left(\left(\frac{TP_{ci}}{TP_{ci} + FN_{ci}} \right) + \left(\frac{TN_{ci}}{TN_{ci} + FP_{ci}} \right) \right) / 2 \quad (18)$$

$$Accuracy = \frac{(TP_{ci} + TN_{ci})}{(TP_{ci} + TN_{ci} + FP_{ci} + FN_{ci})} \quad (19)$$

$$ErrorRate = 1 - Accuracy \quad (20)$$

P (Precision) and R (Recall) were computed for multi-class evaluation, i.e., they are computed for each class ci ∈ C of the text collection. Precision returns the percentage of documents correctly classified as a class ci considering all documents classified as ci, and recall returns the percentage of documents correctly classified as ci considering all documents which actually belong to class ci. where TPc_i (True Positive) means the number of test documents correctly assigned to class ci, FPc_i (False Positive) means the number of test documents from class c_j ($c_j \neq c_i$) but assigned to class ci, and FNc_i (False Negative) is the number of test documents from class c_i but assigned to class c_j ($c_j \neq c_i$).

4. Results and Assessment

In this section, the done assessment and results are presented for Reuters-21578 [21], WebKb [22], Cade 12 [23], and 20 Newsgroup [24] datasets in VC#.NET 2017 programming environment. The primary population and the repetition number in SFLA are 50 and 100 respectively. For showing the efficiency of the proposed model, at first the dataset was performed on ID3 tree.
Reuters-21578: This English corpus contains 90 classes of news documents. The top 10 largest classes were selected for the multiclass text classification experiments. The reduced corpus contains 9980 documents, which have been

divided into a training set with 7193 documents and a test set with 2787 documents according to ModApte Split[1]. We removed the duplicate documents from this so-called Reuters-21578 ModApte Top 10 corpus. All the duplicates labeled as different classes are removed. But one document is kept while the rest are removed in each group of duplicates being of the same class. After removing the duplicates in the training set and test set respectively, the corn and wheat classes become empty or only one document left. Hence, the two classes are finally deleted. The remaining eight classes contain totally 8120 documents including 5798 training documents and 2322 test documents. The eight classes in the training set are severely imbalanced with 2843 documents in the largest earn class and only 79 documents in the smallest grain class. By preprocessing in the same way as described above for 20 Newsgroups, the resulting corpus has a vocabulary of 8436 words (terms).

Webkb: In Webkb collection, we can find texts from the websites of university computer science departments, where 8,282 pages were manually classified into the following categories: student, faculty, staff, course, project, department and other. In the Webkb dataset, the "student" class has most training samples, whereas the "project" class has least training samples. The MI has been assigned highest words (3756) to class "project" whereas the least number of words (3214) has been assigned to class "student". The distribution of words in the classes of Webkb dataset was found to be more balanced.

Cade 12: Cade 12 includes a collection of various documents from web pages that are compiled from Brazilian websites. This dataset contains of 40,983 text documents.

20 Newsgroups: This English corpus contains 19,997 documents of newsgroup messages, which are organized into 20 newsgroups or classes. The 20news-bydate version with 1151 duplicate documents and some headers of the messages already removed was used in the experiments. It includes 18,846 documents which have already been sorted by date and divided into a training set with 11,314 documents and a test set with 7532 documents. Most classes in the training set are of approximately equal size, each with nearly 600 documents. During the preprocessing phase, all the stop words in the stop list defined in the SMART project, rare words that occur less than 2 times in the dataset, numbers, punctuations and other non-alphabetic characters are removed. Meanwhile, the letters are converted into lowercase and words are stemmed using Porter's stemmer, for example, removing the verb suffixes such as -s, -ed and -ing. Finally, 35,642 terms are extracted from the training set to construct the original feature space of the corpus.

4.1. ID3 Tree

In Table (1), the results of the ID3 tree on Reuters-21578, WebKb, Cade 12, and 20 Newsgroup are presented. In ID3 tree the highest accuracy value is that of WebKb.

Table 1. Results of Datasets based on ID3 Tree

Criteria	Reuters-21578	WebKb	Cade12	20 Newsgroup
Precision	0.5623	0.6084	0.6348	0.6031
Recall	0.6945	0.6232	0.6942	0.6248
F-Measure	0.6241	0.6157	0.6632	0.6138
AUC	0.7264	0.7149	0.7413	0.7034
Accuracy	0.7341	0.7648	0.7501	0.7194
Error Rate	0.2659	0.2352	0.2499	0.2806

The main problem of the proposed model is that its performance depends on the quality of the keywords and title words. As shown in Table 1, we obtained the worst performance in the 20 Newsgroup dataset. In fact, title words and keywords of each category in the 20 Newsgroup dataset also have high frequency in other categories.

4.2. Proposed Model

In Table (2), the results of the proposed model with various FS are presented on Reuters-21578. As can be seen in Table (2), accuracy has various values based on different features. The highest accuracy value in the proposed model with 200 features on Reuters-21578 is 0.8036.

In Table (3), the results of the proposed model with various FS are presented on WebKb. As can be seen in Table (3), accuracy has various values based on different features. The highest accuracy value in the proposed model with 300 features on WebKb is 0.8452.

In Table (4), the results of the proposed model with various FS are presented on Cade 12. As can be seen in Table (4), accuracy has various values based on different features. The highest accuracy value in the proposed model with 380 features on Cade 12 is 0.8615.

In Table (5), the results of the proposed model with various FS are presented on 20 Newsgroup. As can be seen in Table (5) accuracy has various values based on different features. The highest accuracy value in the proposed model with 340 features on 20 Newsgroup is 0.7708.

4.3. Improved Proposed Model

In Table (6), the results of the improved proposed model with various FS are presented on Reuters-21578. The highest accuracy value in the improved proposed model with 240 features on Reuters-21578 is 0.9514.

In Table (7), the results of the improved proposed model with various FS are presented on WebKb. The highest accuracy value in the improved proposed model with 200 features on WebKb is 0.9507.

In Table (8), the results of the improved proposed model with various FS are presented on Cade 12. The highest accuracy value in the improved proposed model with 300 features on Cade 12 is 0.9561.

In Table (9), the results of the improved proposed model with various FS are presented on 20 Newsgroup. The highest accuracy value in the improved proposed model with 180 features on 20 Newsgroup is 0.8969.

[1] http://disi.unitn.it/moschitti/corpora.htm

a

Table 2. Results of the Proposed Model with Various FS on Reuters-21578

Number of Features	Reuters-21578					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.6523	0.7011	0.6758	0.7388	0.7924	0.2076
80	0.6214	0.6611	0.6406	0.7490	0.7621	0.2379
100	0.6034	0.7112	0.6529	0.7514	0.7526	0.2470
140	0.6948	0.7234	0.7088	0.7032	0.7684	0.2316
180	0.6012	0.7315	0.6600	0.7659	0.7730	0.2270
200	0.6147	0.7614	0.6802	0.7741	0.8036	0.1964
240	0.6340	0.6920	0.6617	0.7698	0.7642	0.2358
300	0.6814	0.7248	0.7024	0.7550	0.7328	0.2672
340	0.6318	0.7398	0.6815	0.7614	0.7703	0.2297
380	0.6547	0.7044	0.6786	0.7362	0.7864	0.2136

Table 3. Results of the Proposed Model with Various FS on WebKb

Number of Features	WebKb					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.6632	0.6841	0.6735	0.7619	0.7935	0.2065
80	0.6847	0.7234	0.7035	0.7805	0.8031	0.1969
100	0.7012	0.7345	0.7175	0.7954	0.8241	0.1759
140	0.7150	0.7511	0.7326	0.8014	0.7624	0.2376
180	0.6621	0.7400	0.6989	0.7511	0.7956	0.2044
200	0.6532	0.6948	0.6734	0.7951	0.8317	0.1683
240	0.6940	0.7241	0.7087	0.7603	0.8297	0.1703
300	0.6725	0.7324	0.7012	0.7612	0.8452	0.1548
340	0.6814	0.7502	0.7141	0.8136	0.8310	0.1690
380	0.6922	0.7333	0.7122	0.7924	0.8259	0.1741

Table 4. Results of the Proposed Model with Various FS on Cade 12

Number of Features	Cade12					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.6458	0.6954	0.6697	0.7620	0.7914	0.2086
80	0.6614	0.7021	0.6811	0.7512	0.8035	0.1965
100	0.6925	0.7013	0.6969	0.7865	0.8241	0.1759
140	0.6718	0.6924	0.6819	0.7921	0.8322	0.1678
180	0.6921	0.7328	0.7119	0.7647	0.8290	0.1710
200	0.6800	0.7194	0.6991	0.7412	0.8432	0.1568
240	0.6540	0.6812	0.6673	0.7365	0.7621	0.2379
300	0.6847	0.7264	0.7049	0.7824	0.8035	0.1965
340	0.6748	0.7924	0.7289	0.7611	0.8132	0.1868
380	0.6654	0.7015	0.6830	0.7760	0.8615	0.1385

Table 5. Results of the Proposed Model with Various FS on 20 Newsgroup

Number of Features	20 Newsgroup					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.6532	0.6631	0.6581	0.6931	0.7319	0.2681
80	0.6489	0.6780	0.6631	0.7014	0.7634	0.2366
100	0.6924	0.7024	0.6974	0.7134	0.7518	0.2482
140	0.6647	0.6733	0.6690	0.7348	0.7664	0.2336
180	0.6821	0.6901	0.6861	0.7420	0.7497	0.2503
200	0.6608	0.6907	0.6754	0.7158	0.7314	0.2686
240	0.6597	0.6751	0.6673	0.7034	0.7580	0.2420
300	0.6924	0.7134	0.7027	0.6921	0.7621	0.2379
340	0.6618	0.6929	0.6770	0.7150	0.7708	0.2292
380	0.6712	0.7064	0.6884	0.7284	0.7694	0.2306

Table 6. Results of the Improved Proposed Model with Various FS on Reuters-21578

Number of Features	Reuters-21578					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.7032	0.7355	0.7190	0.8135	0.9247	0.0753
80	0.7265	0.7508	0.7385	0.8362	0.9321	0.0679
100	0.7569	0.7632	0.7600	0.8027	0.9439	0.0561
140	0.7421	0.7814	0.7612	0.8632	0.9485	0.0515
180	0.7741	0.7932	0.7835	0.8936	0.9154	0.0846
200	0.7621	0.8036	0.7823	0.8810	0.9237	0.0863
240	0.7310	0.7414	0.7362	0.8525	0.9514	0.0486
300	0.7519	0.7622	0.7570	0.8635	0.9431	0.0569
340	0.7698	0.7932	0.7813	0.8730	0.9501	0.0499
380	0.7425	0.7930	0.7669	0.8314	0.9214	0.0786

Table 7. Results of the Improved Proposed Model with Various FS on WebKb

Number of Features	WebKb					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.7236	0.7936	0.7570	0.8514	0.9123	0.0877
80	0.7145	0.8024	0.7559	0.8632	0.9236	0.0764
100	0.7384	0.8214	0.7777	0.8478	0.9341	0.0659
140	0.7564	0.7736	0.7649	0.8314	0.9014	0.0986
180	0.7235	0.7514	0.7372	0.8269	0.9214	0.0786
200	0.7014	0.7325	0.7166	0.8678	0.9507	0.0493
240	0.7625	0.7928	0.7774	0.8725	0.9347	0.0653
300	0.7902	0.8195	0.8046	0.8832	0.9289	0.0711
340	0.7624	0.8036	0.7825	0.8625	0.9352	0.0648
380	0.7532	0.8130	0.7820	0.8347	0.9478	0.0522

Table 8. Results of the Improved Proposed Model with Various FS on Cade 12

Number of Features	Cade12					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.7325	0.7934	0.7617	0.8209	0.9014	0.0986
80	0.7194	0.7514	0.7351	0.8319	0.9347	0.0653
100	0.7036	0.7364	0.7196	0.8417	0.9531	0.0469
140	0.7539	0.7841	0.7687	0.8635	0.9104	0.0896
180	0.7435	0.7620	0.7526	0.8427	0.9278	0.0722
200	0.7314	0.7880	0.7586	0.8314	0.9314	0.0686
240	0.7279	0.8037	0.7639	0.8406	0.9401	0.0599
300	0.7498	0.8195	0.7831	0.8532	0.9561	0.0439
340	0.7311	0.7931	0.7608	0.8230	0.9512	0.0488
380	0.7625	0.7966	0.7792	0.8470	0.9310	0.0690

Table 9. Results of the Improved Proposed Model with Various FS on 20 Newsgroup

Number of Features	20 Newsgroup					
	Precision	Recall	F-Measure	AUC	Accuracy	Error Rate
40	0.6924	0.7164	0.7042	0.7602	0.8034	0.1966
80	0.7023	0.7315	0.7166	0.7514	0.8001	0.1999
100	0.7238	0.7416	0.7326	0.7319	0.8903	0.1097
140	0.7059	0.7223	0.7140	0.7681	0.8017	0.1983
180	0.7149	0.7201	0.7175	0.7846	0.8969	0.1501
200	0.7237	0.7631	0.7429	0.7694	0.8234	0.1766
240	0.7309	0.7518	0.7412	0.7915	0.8404	0.1596
300	0.7401	0.7506	0.7453	0.7838	0.8620	0.1380
340	0.7132	0.7238	0.7185	0.7925	0.8219	0.1781
380	0.7062	0.7297	0.7178	0.7527	0.8947	0.1053

In Table (10), the comparison of the proposed model and the improved proposed model is presented based on the in accuracy criterion with various features in the four text datasets. The results are analyzed based on the selected features. In experimentation, it has been found that the performance of all FS have been improved by embedding the proposed Equation (12).

4.4. Comparison and Assessment

In this section, the results of the improved proposed model are compared with machine learning techniques on datasets Reuters-21578, WebKb, Cade 12 and 20 Newsgroup.

4.4.1. Machine Learning Models

In Table (11), the results of the proposed model are compared with machine learning models on Reuters-21578. The highest accuracy among the machine learning models is that of Bagging+ RF. Many machine learning algorithms have been applied to text classification tasks. In the machine learning paradigm, a general inductive process automatically builds a text classifier by learning, generally known as supervised learning.

In Table (12), the results of the proposed model are compared with NB and KNN models on Reuters-21578. Comparison was done based on 10 features; and the improved proposed model was more accurate than KNN and NB in 6 features.

In Table (13), the results of the improved proposed model are compared with other existing models on 20 Newsgroup. We could achieve significant improvement in the 20 Newsgroup data set. Thus we find that the proposed model for choosing keywords is more useful in a domain with confused keywords between categories such as the 20 Newsgroup data set.

4.4.2. NB-K-Means Model

In Table (14), the results of the proposed model are compared with NB-K-Means model based on the Accuracy criterion on the datasets Reuters-21578, WebKb, and Cade 12. Table (14) shows that the proposed model is more accurate and the reason is use of effective features in classification.

5. Conclusion and Future Works

An improvement model of SFLA with ID3 tree is proposed in this paper. By means of effectiveness of the keywords in text documents and omission of unnecessary words we managed to increase the real accuracy in identifying the similarities of documents and to use this similarity in classification of text documents. The proposed model includes a feature stage in which the keywords of the text are extracted and the initial vectors are formed using SFLA. In the classification stage, by means of ID3 tree and with help from weight of the keywords the similarities of the ID3 tree nodes are estimated. In the improved proposed model a hybrid of the best and the worst frog situations are used to find the optimized vectors. The results of the improved proposed model on the datasets Reuters-21578, WebKb, Cade 12, and 20 Newsgroup suggest that the proposed model has a higher classification accuracy in comparison with ID3 tree. For future works, one can use supervised learning methods that use labeled educational examples, as it is an efficient text documents classification approach.

Table 10. Comparison of the Proposed Model and the Improved Proposed Model based on Accuracy

Number of Features	Accuracy				Accuracy			
	Proposed Model				Improved Proposed Model			
	Reuters-21578	WebKb	Cade12	20 Newsgroup	Reuters-21578	WebKb	Cade12	20 Newsgroup
40	0.7924	0.7935	0.7914	0.7319	0.9247	0.9123	0.9014	0.8034
80	0.7621	0.8031	0.8035	0.7634	0.9321	0.9236	0.9347	0.8001
100	0.7526	0.8241	0.8241	0.7518	0.9439	0.9341	0.9531	0.8903
140	0.7684	0.7624	0.8322	0.7664	0.9485	0.9014	0.9104	0.8017
180	0.7730	0.7956	0.8290	0.7497	0.9154	0.9214	0.9278	0.8969
200	0.8036	0.8317	0.8432	0.7314	0.9237	0.9507	0.9314	0.8234
240	0.7642	0.8297	0.7621	0.7580	0.9514	0.9347	0.9401	0.8404
300	0.7328	0.8452	0.8035	0.7621	0.9431	0.9289	0.9561	0.8620
340	0.7703	0.8310	0.8132	0.7708	0.9501	0.9352	0.9512	0.8219
380	0.7864	0.8259	0.8615	0.7694	0.9214	0.9478	0.9310	0.8947

Table 11. Comparison of the Improved Proposed Model and Machine Learning Models on Reuters-21578

Models [23]	Criteria		
	F-Measure	AUC	Accuracy
NB [23]	0.8100	0.8500	0.8162
SVM [23]	0.7600	0.8000	0.7798
LR [23]	0.7900	0.8100	0.7879
RF [23]	0.8000	0.8500	0.7978
Bagging+ RF [23]	0.8900	0.9900	0.8896
RS+RF [23]	0.8800	0.9900	0.8863
MV [23]	0.8200	0.8700	0.8264
Improved Proposed Model	0.7362	0.8525	0.9514

Table 12. Comparison of the Improved Proposed Model and KNN and NB Models on Reuters-21578

Models [13]	Criteria	Number of Features									
		91	**182**	**273**	**364**	**455**	**546**	**637**	**728**	**819**	**911**
KNN [13]	Precision	0.9590	0.9631	0.9636	0.9600	0.9598	0.9575	0.9585	0.9569	0.9556	0.9584
	Recall	0.9311	0.9550	0.9555	0.9542	0.9542	0.9558	0.9539	0.9539	0.9526	0.9521
	F-Measure	0.9436	0.9478	0.9510	0.9508	0.9515	0.9530	0.9530	0.9515	0.9509	0.9569
	Accuracy	0.8809	0.8923	0.8988	0.8972	0.8988	0.9005	0.9008	0.8967	0.8955	0.8946
NB [13]	Precision	0.9598	0.9231	0.9109	0.9071	0.9122	0.9065	0.9054	0.9024	0.9068	0.9019
	Recall	0.8484	0.8755	0.8943	0.9028	0.9038	0.9081	0.9089	0.9102	0.9094	0.9091
	F-Measure	0.8899	0.8984	0.9008	0.9012	0.9046	0.9029	0.9025	0.9033	0.9060	0.9029
	Accuracy	0.7690	0.7793	0.7828	0.7898	0.7971	0.7977	0.8024	0.8001	0.8038	0.8036
Improved Propose	Precision	0.9623	0.9615	0.9624	0.9534	0.9418	0.9741	0.9584	0.9420	0.9514	0.9651
	Recall	0.9421	0.9264	0.9334	0.9507	0.9337	0.9425	0.9530	0.9304	0.9288	0.9515
	F-Measure	0.9520	94.36	94.76	95.20	93.77	95.80	95.56	93.61	93.99	95.82
	Accuracy	0.9034	0.9317	0.9187	0.9047	0.8624	0.8918	0.9187	0.8608	0.8716	0.9064

Table 13. Comparison of the Improved Proposed Model and other Existing Models on 20 Newsgroup

Models	**Accuracy**
NB [24]	85.00
NB [24]	74.00
SVM [25]	86.40
SVM [25]	87.00
NB [26]	79.00
Centroid Classifier [26]	73.00
KNN [26]	70.00
SVM [27]	84.40
Centroid Classifier [28]	84.40
Centroid Classifier [29]	83.89
NB [29]	83.55
KNN [29]	84.61
SVM [30]	85.06
SVM [31]	86.44
NB +SVM [32]	78.05
NB +SVM [32]	77.00
NB +SVM [32]	75.00
SVM [33]	84.82
SVM[34]	86.79
KNN [35]	82.30
SVM [35]	81.60
NB [36]	87.45
NB+SVM [36]	88.02
KNN [37]	86.80
SVM [37]	90.06
Regression Classifier [38]	89.47
Interval Valued Classifier+ Neural Network Classifier [39]	96.48
TFCP [1]	86.19
NB [1]	72.68
NB [1]	91.72
Improved Proposed Model	93.07

Table 14. Comparison of the Improved Proposed Model and NB-K-Means model based on Accuracy

Models	Reuters-21578 R8	Reuters-21578 R52	WebKb	Cade12
NB-K-Means [15]	91.60	88.50	94.80	88.10
Improved Proposed Model	**96.30**	**95.14**	**95.07**	**95.61**

References

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

[2] X. Ma, R. Jin, J.Y. Paik, T.S. Chung, Large Scale Text Classification with Efficient Word Embedding, International Conference on Mobile and Wireless Technology, ICMWT 2017: Mobile and Wireless Technologies, Springer, Singapore, Vol. 425, pp. 465-469, 2018.

[3] A. Allahverdipour, F.S. Gharehchopogh, An Improved K-Nearest Neighbor with Crow Search Algorithm for Feature Selection in Text Documents Classification, Journal of Advances in Computer Research, 9(2): 37-48, 2018.

[4] H. Majidpour, F.S. Gharehchopogh, An Improved Flower Pollination Algorithm with AdaBoost Algorithm for Feature Selection in Text Documents Classification, Journal of Advances in Computer Research, 9(1): 29-40, 2018.

[5] U. Pandey, S. Chakraverty, R. Mihani, R. Arya, S. Rathee, R. K. Sharma, Semantic Based Category-Keywords List Enrichment for Document Classification, Advances in Intelligent and Soft Computing, Vol. 166, pp. 297-309, 2012.

[6] M. Eusuff, K. Lansey, and F. Pasha, Shuffled frogleaping algorithm: a memetic meta-heuristic for

discrete optimization, EngOptim, 38(2):129-154, 2006.
[7] J.R. Quinlan, Induction of Decision Trees. Machine learning, 1(1), 81-106, 1986
[8] Li-Wei Lee and Shyi-Ming Chen, New Methods for Text Categorization Based on a New Feature Selection Method and a New Similarity Measure Between Documents, Springer-Verlag Berlin Heidelberg, IEA/AIE 2006, LNAI 4031, pp. 1280-1289, 2006.
[9] B.S. Harish, D.S. Guru, and S. Manjunath, Classification of Text Documents Using B-Tree, CCSIT 2012, Part II, LNICST 85, pp. 627-636, 2012
[10] B.S. Harish, Bhanu Prasad, and B. Udayasri, Classification of Text Documents Using Adaptive Fuzzy C-Means Clustering, Springer International Publishing Switzerland, Recent Advances in Intelligent Informatics, Advances in Intelligent Systems and Computing, Vol. 235, pp. 205-214,
[11] 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.
[12] A. Onan, S. Korukoglu, H. Bulut, Ensemble of keyword extraction methods and classifiers in text classification, Expert Systems With Applications, Vol. 57 pp. 232-247, 2016.
[13] F. Yigit; O.K. Baykan, A new feature selection method for text categorization based on information gain and particle swarm optimization, 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, pp. 523-529, 2014
[14] S. Khalandi, F.S.Gharehchopogh, A New Approach for Text Documents Classification with Invasive Weed Optimization and Naive Bayes Classifier, Vol:4, No:3, Page 31-40, 2018
[15] A. Allahverdipour, F.S. Gharehchopogh, "A New Hybrid Model of K-Means and Naïve Bayes Algorithms for Feature Selection in Text Documents Categorization," Journal of Advances in Computer Research, Vol: 8, No: 4, 2017.
[16] 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.
[17] H.J. Escalante, M.A. Garcia-Limon, A.M. Reyes, M. Graff, J.M. Carranza, Term-weighting learning via genetic programming for text classification, Knowledge-Based Systems, Vol. 83, pp. 176-189, 2015.
[18] 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.
[19] G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.
[20] R.S. Michalski, I. Bratko, M. Kubat, Machine Learning and Data Mining: Methods and Applications, New York: Wiley, 1998.
[21] http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection
[22] http://ana.cachopo.org/datasets-for-single-label-text-categorization
[23] 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, pp. 1024-1032, 2011.
[24] A.K. Mccallum, K. Nigam, Employing EM in pool-based active learning for text classification, in Proceedings of the 15th International Conference on Machine Learning, USA, pp. 350-358, 1998.
[25] L.D. Baker, A.K. Mccallum, Distributional clustering of words for text classification, in Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval, Australia, pp. 96-103, 1998.
[26] K.M. Schneider, A new feature selection score for multinomial naive Bayes text classification based on KL-divergence, in Proceedings of the 42nd Annual Meeting ofthe Association for Computational Linguistics, pp. 186-189, 2004.
[27] XX. Sun, Y. Liu, H.T. Loh, Imbalanced text classification: a term weighting approach, Expert Systems with Applications, Vol. 36, Issue 1, pp. 690-701, 2009.
[28] V. Lertnattee, T. Theeramunkong, Class normalization in centroid-based text categorization, Information Sciences, Vol. 176, Issue 12, pp. 1712-1738, 2006.
[29] S. Tan, An effective refinement strategy for k-NN text classifier, Expert Systems with Applications, Vol. 30, Issue 2, pp. 290-298, 2007.
[30] M.A. Kumar, M. Gopal, A comparison study on multiple binary-class SVM methods for unilabel text categorization, Pattern Recognition Letters, Vol. 31, Issue 11, pp. 1437-1444, 2010.
[31] S. Aseervatham, A. Antoniadis, E. Gaussier, M. Burlet, Y. Denneulin, A sparse version of the ridge logistic regression for large-scale text categorization, Pattern Recognition Letters, Vol. 32, Issue 2, pp. 101-106, 2011.
[32] LL. Dhillon, S. Mallela, and R. Kumar, Enhanced word clustering for hierarchical text classification, in: Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining, Canada, pp. 191-200, 2002.
[33] H.A. Mubaid, S.A. Umair, A new text categorization technique using distributional clustering and learning logic, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, Issue 9, pp. 1156-1165, 2006.
[34] S.S. Kim, K.S. Han, H.C. Rim, S.H. Myaeng, Some effective techniques for naive Bayes text classification, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, Issue 11, pp. 1457-1466, 2006.
[35] D.Cai, X. He, W.V. Zhang, and J. Han, Regularized locality preserving indexing via spectral regression, in Proceedings of the 16th Conference on Information and Knowledge Management, USA, pp. 741-750, 2007.
[36] D. Isa, L.H. Lee, V.P. Kallimani, R. Rajkumar, Text document preprocessing with the bayes formula for classification using the support vector machine, IEEE Transactions on Knowledge and Data Engineering, Vol. 20, Issue 9, pp. 23-31, 2008.
[37] X.B. Xue, Z.H. Zhou, Distributional features for text categorization, IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 3, pp. 428-442, 2009.
[38] XX. Danti and S.N.B. Bhushan. Classification of text documents using integer representation and regression: an integrated approach. Special Issue IIOAB Scopus Index. J. Vol. 7, No. 2, pp. 45–50. 2016.
[39] S.N.B. Bhushan, A. Danti, Classification of text

documents based on score level fusion approach, Pattern Recognition Letters, Vol. 94, pp. 118-126, 2017.

Farhad Soleimanian Gharehchopogh received his Ph.D. in computer engineering from Hacettepe University, Ankara, Turkey. He is an Asistant Professor at the department of Computer Engineering in Islamic Azad University, Urmia Branch, Iran. His research focus is on Machine Learning, Natural Language Processing, Information Retrieval.
Email: bonab.farhad@gmail.com

Mostafa Mahmoudi received his MSc. degree in computer engineering from Islamic Azad University, Urmia Branch, IRAN in 2017. His research interests are mainly in the field of Data Mining, Text Document Classification, and Information Retrieval.
Email: m.mahmoudi@farhangmail.ir

Paper Handling Data:

Submitted: 13.02.2018
Received in revised form: 19.10.2018
Accepted: 07.11.2018
Corresponding author:
Affiliation of the corresponding author: Dr. Farhad Soleimanian Gharehchopogh, Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, IRAN