Machine learning for text document classification-efficient classification approach

Sura I. Mohammed Ali[1], Marwah Nihad[2], Hussien Mohamed Sharaf[3], Haitham Farouk[3]
[1]Department of Mathematics and Computer Application, Collage of Science, Al-Muthanna University, Samawah, Iraq
[2]Faculty of Science, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq
[3]Department of Computer Science, Faculty of Computers and Information, Suez University, Suez, Egypt

Article Info

Article history:

Received Mar 16, 2023
Revised Jul 6, 2023
Accepted Jul 21, 2023

Keywords:

Cosine similarity
Information retrievel
Machine learning
Multinomial naïve bayesian
Text documents classifiers

ABSTRACT

Numerous alternative methods for text classification have been created because of the increase in the amount of online text information available. The cosine similarity classifier is the most extensively utilized simple and efficient approach. It improves text classification performance. It is combined with estimated values provided by conventional classifiers such as Multinomial Naive Bayesian (MNB). Consequently, combining the similarity between a test document and a category with the estimated value for the category enhances the performance of the classifier. This approach provides a text document categorization method that is both efficient and effective. In addition, methods for determining the proper relationship between a set of words in a document and its document categorization is also obtained.

Corresponding Author:

Haitham Farouk
Department of Computer Science, Faculty of Computers and Information, Suez University
Suez, Egypt
Email: h.farouk@suezuni.edu.eg

1. INTRODUCTION

One of the possible solutions of the information resources problem is text document (TD) classification [1]. It's hard to cover all the many algorithms in the field of text categorization. Recently, extensive research in the field of financial sentiment analysis has been conducted. Sentiment analysis (SA) of any text data denotes the feelings and attitudes of the individual on particular topics or products. It applies statistical approaches with artificial intelligence (AI) algorithms to extract substantial knowledge from a huge amount of data. This study extracts the sentiment polarity (negative, positive, and neutral) from financial textual data using machine learning and deep learning algorithms. The constructed machine learning model used ultinomial Naive Bayes (MNB) and logistic regression (LR) classifiers. On the other hand, three deep learning algorithms have been utilized which are recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU) [2], [3]. The challenge of feature selection in text categorization is a significant one. We try to figure out which features are most important to the categorization process during feature selection. This is because some words are considerably more likely than others to be linked with the class distribution. As a result, the study proposes a wide range of strategies for determining the most significant characteristics for classification purposes. We'll also go over the various text classification feature selection approaches that are widely utilized. Preprocessing, feature extraction, feature selection, and categorization are all included in the text categorization process. Text documents are used to extract features in feature extraction process [4].

Each text document term (word) is considered a feature, and the majority of the features are undesirable and unnecessary. Tokenization, stop-word removal, and stemming are also used during preprocessing to remove unnecessary and undesired features [5]. A representation model is used to represent the pre-processed text content in a machine-understandable structure. Then, given the representation model, the feature selection technique selects the most informative features [6]. Feature selection has a significant impact on classifier performance and is primarily utilized for dimensionality reduction [7], [8]. Finally, using the selected feature subset, a classifier is utilized to categorize the text documents. The large dimensionality of feature space makes text categorization so difficult. As a result, the classifiers performance deteriorates, and categorization takes longer [9], [10]. Because of its computing economy and high effectiveness, cosine similarity (CS) is commonly employed in the text categorization sector. There are already classifiers that use CS, such as the centroid-based classifier [11], [12].

Cumuli geometric centroid (CGC), arithmetical average centroid (AAC), and class feature centroid (CFC) are examples of centroid-based classifiers (CBC), where centroid denotes the technique for creating a CBC class prototype vector (i.e., the initialization procedures). The sum of each class's overall number of words is utilized by CGC; AAC utilized the arithmetical average of each class's overall number of words, while CFC uses the inner-class and inter-class term indexes [11]. The weight model is a new CBC model that focuses on categorization hyper plane modification.

Beyond the classification of text documents, we present a CS technique in this paper. To classify the collection of words into equivalence classes, we calculate the similarity degree and utilize the symmetric measure for mutual support between words. Because of its computing economy and high effectiveness, CS is commonly employed in the text categorization sector. There are already classifiers that use CS, such as centroid-based classifiers [11]. The main approach consists of 4 steps, and we are using examples in methodology. The remainder of the paper is organized: i) The text classification process is discussed in section 2; ii) The related work is summarized in section 3; iii) Existing classification techniques are discussed in section 4; iv) Section 5 introduces the proposed methodology; v) Section 6 shows the results and discussion; and vi) The conclusion section of the document brings the paper to an end in the final section 7.

2. DATA PROCESSING

The main intention of textual content mining is to allow customers to extract statistics from textual assets and deal with the operations like, retrieval, category, and clustering (supervised, unsupervised, and semi supervised). But how those documented may be nicely annotated, presented, categorized, and clustered. Figure 1 depicts the text classification process. The text classification problem is distinct in that the number of characteristics (unique words or phrases) can easily exceed tens of thousands. When it comes to using numerous complex learning algorithms for text categorization, this poses major hurdles. As a result, approaches for reducing dimensions are required. The two alternatives are to select a subset of the original features or to change the features into new ones by computing new features as functions of the existing ones.

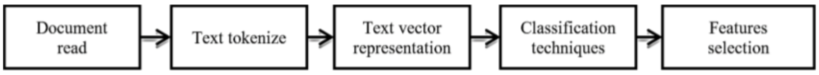

Figure 1. Text clasfication process

The number of attributes (unique words or phrases) in the text classification issue can easily surpass tens of thousands. This presents significant challenges when it comes to applying a variety of complicated learning algorithms for text categorization. As a result, methods for lowering dimension are necessary. The two alternatives are to select a subset of the original features or to change the features into new ones by computing new features as functions of the existing ones.

Although machine learning-based text categorization is a good method in terms of performance, it is inefficient when dealing with big training datasets. As a result, in addition to feature selection, instance selection is frequently required. For text classification, combined feature and instance selection. Their strategy consists of two phases [13]. In the first phase, their algorithm selects features with high precision in predicting the target class in a sequential manner. All documents without at least one of these features are removed from the training set. In the second phase, their algorithm looks for a set of characteristics that tend to predict the complement of the target class inside this subset of the initial dataset, and these features are also chosen. The new feature set is the sum of the features chosen in these two processes, whereas the training set is made up of the documents chosen in the first step. In this paper, the steps followed in the case study are based on the data mining methodology proposed by [14]. The steps include data selection, preprocessing, data transformation, data mining, and analysis. The process of classification of TD approach is as follows:

a) Using the position weight algorithm, generate keywords from text documents. The most crucial information is contained in keywords, which are index terms. The task of automatically extracting limited keywords, key phrases, or set of words from a document that can explain the content's significance is known as automatic keyword extraction. All automatic processing for text resources relies on keyword extraction as a core technology. A survey of keyword extraction strategies has been offered in this study, which can be used to extract effective keywords that uniquely identify a document.
b) Using the CS technique, compare the input (keywords) to other texts (as a query or keyword) to identify the input's class.
c) Creating class probabilities by using keywords.
d) Use text classification techniques to help organize information.

Three predictions emerge from stages 2, 3, and 4. We can make the system's output CLASS1 if the majority forecast was CLASS1. Using the position weight algorithm [12], generate keywords from text sources.

Regarding how to choose important words, in linguistics, the word location is very essential. The entropy of words in different positions varies. The opinion carries additional information when they appear in the document's introduction and conclusion paragraphs, which are normally the first and last paragraphs. Furthermore, leading and summary sentences usually have more important words than the rest of the paragraph. We employ a unique method called position weight (PW) to capture the relevance of a word position.

Paragraphs make up a common document (the title is considered a special paragraph), sentences make up a paragraph, and words make up a sentence. A term's PW must take into account three key elements: paragraph, sentence, and word. The PW of a phrase t in a certain location is defined as (1). Where $pw\left(t_i, p_j\right)$ in the paragraph *j*, represents the PW of phrase t; $pw(t_i, s_k)$ is in the sentence *k*, reflects the PW of term *t*; $pw(t_i . w_r)$ as a word form *r*, reflects the PW of phrase *t*. In a document, the total weight of the word *t* is the sum of the weights of all spots in which it appears. The pw of a phrase *t* in a document *d* that appears *m* times by (2).

$$pw(t_i) = pw\left(t_i, p_j\right) . pw(t_i, s_k) . pw(t_i . w_r) \tag{1}$$

$$pw(t, d) = \sum_{i=1}^{m} pw(t_i) \tag{2}$$

The importance of keywords is higher than that of other terms; a keyword may be used to characterize the characteristics of a document, which is why they can be used to distinguish between different document types. Assume that documents D_1 and D_2 fall under the "computer science" and "mathematics" categories, respectively. Although "theorem" cannot be deemed a keyword in either D_1 or D_2, the terms "approach" and "theorem" have greater weights in D_1 and D_2.

For property, we use $W_{D_{1k}}$ as the weight of W_K in proportion to D_1. The digits 0 and 1 are used to signify W_K. For example, Table 1 shows the number of times each word appears in each document, as well as the document set D and the word set W that covers D_s. The set of keywords is covering $D_i \in C_i$, where C_i is any class. Suppose that W_{D_1}= {W_5, W_6}, W_{D_2} = {W_2, W_3, W_4, W_5, W_6}, W_{D_3}= {W_1, W_3, W_6}. Using the CS technique, compare the input (keywords) to other texts (as a query or keyword) to identify the input's class. Creating class probabilities by using keywords. Use text classification techniques to help organize information.

Table 1. Data frame

Keyword document	W_1	W_2	W_3	W_4	W_5	W_6
D_1	0	0	0	0	1	1
D_2	0	1	1	1	1	1
D_3	1	0	1	0	0	1

3. RELATED WORK

The approach of categorizing text documents into specified groups is known as text classification, and it has received a lot of interest in contemporary years as a result of the expansion of digital documents. Approaches based on statistical theory or machine learning to improve text categorization ability have become mainstream. with data mining techniques like K-means, EM, Apriori, SVM, C4.5, and PageRank being used. Classification and regression trees (CART), AdaBoost, K-nearest neighbor (KNN), and Nave Bayes are popular algorithm among these since it has a high computational efficiency and an excellent prediction performance.

Enhanced classifiers and conventional classifiers using the accuracy of confusion (or misclassification) matrices based on five based (R8, 20NG, R52, Cade12, and WebKB) have been discussed in [15], [16]. MNB's performance was improved by developing a fine-tuning process. A methodology has been introduced that employs three metaheuristic methodologies to convert an eventual estimation problem into an optimization problem: genetic algorithms, simulation annealing, and differential evolution in [17]. A proposed approach for consolidating the aftereffects of two classifiers, like MNB and a changed most extreme entropy classifier (an adjusted form of the authors' proposed conventional maximum entropy classifier) [18]. CFC, AAC, and CGC are examples of centroid-based classifiers, where centroid refers to the CBC method for generating a class prototype vector (i.e., the initialization procedures). AAC uses the arithmetical average of all words in each class. CGC uses the total of all words in each class, whereas CFC uses the inner-class term and the inter-class term index [19]. Based on, the weight model is a new CBC method that focuses on fine-tuning a classification hyperplane.

4. CLASSIFICATION TECHNIQUES REVIEW

Text categorizations were primarily employed for information retrieval systems in the early days of machine learning (ML) and artificial intelligence (AI). Text classification and document categorization, on the other hand, have become widely used in a variety of domains, including medical, social sciences, healthcare, psychology, law, and engineering, as technological breakthroughs have emerged. The classification of the documents can be done using unsupervised, supervised, and semi-supervised approaches. Many techniques and algorithms have been proposed recently for the classification of electronic documents. Supervised machine learning algorithms that make predictions on given set of samples. They search for patterns within the value labels assigned to data points.

While no labels are attached to data points in unsupervised machine learning methods. To define the data's structure and make it appear simple and organized for analysis, they arrange the data into clusters. Algorithms for reinforcement machine learning decide what to do based on each data point and then evaluate the effectiveness of the choice. Over time, they change their strategy to learn better and achieve the best reward. The following are some examples of classification techniques:

- Deep neural networks (DNN), deep belief networks (DBN), hierarchical attention networks (HAN), recurrent neural network (RNN), convolutional neural network (CNN), and combination approaches are among the neural network-based algorithms described. Naïve bayes classifier (NBC): The bayesian classifier is a probabilistic classifier (also known as a generative classifier). The goal is to categorize text based on the subsequent likelihood of documents relationship to distinct classes depending on the existence of certain words in the documents.
- K-nearest neighbor (KNN): It is a supervised learning technique that can be applied to classification and regression problems. It's straightforward, logical, and adaptable. It can be thought of as an algorithm that generates predictions based on the characteristics of other data points in the training dataset that are close by. In simple terms, the classifier algorithm calculates the similarity between the input sample and the k practice instances that are closest to the input sample and produces the class to which the object is most likely to be allocated. It is presumptively true that similar values can be found in close vicinity. Because it does not learn a discriminative function, KNN is sometimes referred to as a lazy learner.
- Support vector machine (SVM): The usage of linear or non-linear delineations between the distinct classes is used by SVM classifiers to partition the data space. The key part of this classifier is determining the best boundaries between the classes and separating them for classification by creating a line or a hyperplane between classes.
- Decision tree (DT): It is created by using different text properties to create a hierarchical division of the underlying data space. The hierarchical segmentation of the data space is intended to provide class partitions with a more skewed distribution of classes. We calculate the division to which a given text instance is most likely to belong and utilize that for classification purposes.
- The naïve bayes classifier is commonly used for text categorization. However, the k-nearest neighbor technique, which is more conventional but still widely used in science. As classification algorithms, support vector machines (SVMs), particularly kernel SVMs, are widely used. Using tree-based classification techniques like decision trees and random forests, document categorization may be done quickly and reliably [20]–[23].

5. METHOD

The major goal of our technique is to determine the suitable link between documents. The text documents are often classified and retrieved according to the users. In our approach, we suggest classifying documents based on word tokens which extract attributes from text of the above two categories.

Moreover, classification approach techniques include term frequency (TF) and CS. Figure 1 shows general steps of the flow diagram for techniques that used in the proposed classification approach and combined CS with estimated values provided by conventional classifiers, it improves the performance of the classifiers. Combining the similarity between a test document and a category with the estimated value for the category enhances classifier performance. Therefore, all documents in the datasets are independently vectorized by word count and by term frequency-inverse document frequency (TF-IDF) for evaluating the performance of the constructed classifiers.

Cosine-similarity is a mathematical measure that identifies documents that are similar regardless of their size. In two-dimensional space, it is the cosine measure of the distinction formed by two vectors, where the two vectors might contain numeric or text data. We use vectors as text data in this paper. We can combine the strategies mentioned above to create a text classification system. The following is the procedure for our approach,

- Document representation

Create a numeric vector from the documents. The document is represented as a vector in cosine-similarity-based text categorization, then used from a lexicon as a result of all of the training documents. The lexicon's k^{th} term is denoted by F = {t_1, t_2,..., $t_{|F|}$}(t_k, k ∈[1, |F|]), and each document is regarded a vector in |F|-dimension feature space. Term of frequency and inverse document frequency (TFIDF) formula is used to convert a document into a numeric vector as in (3) [24].

$$\boldsymbol{tfidf(t_k, d_i) = tfi(t_k, d_i) \times log\frac{|D|}{|D(t)_k|}} \tag{3}$$

Where tfi(t_k, d_i) is the number of times the word t_k appears in document d_i, |D| is the total number of times training documents, and |D(t_k)| is the total number of t_k- approximate documents in text group D. The phrase weighting is then normalized as in (4) [25].

$$\boldsymbol{W_{ki} = \frac{tfi(t_k,d_i)}{\sum_{z=1}^{|F|}(tfi(t_k,d_i))^2}} \tag{4}$$

Where W_{ki} denotes the document's normalized phrase weight t_k. The class centroid C_j is calculated after the normalized representation of documents by adding vectors of all documents in C_j class and then normalize the result by their size. As a result, a class centroid's formal description by (5).

$$\boldsymbol{C_j = \frac{\sum_{d_i \in cj} d_i}{\left\|\sum_{d_i \in cj} d_i\right\|_2}} \tag{5}$$

Where $\|*\|_2$ represents the 2-norm the cosine function [26]–[28], can be used to measurement the similarity between the centroid C_j and an unlabeled document d which is given by next step.

- Class prediction

Based on cosine-similarity functions calculate the similarity between a document word and all class words by comparing the similarity of the input with other texts and thereby determining its class. The enhanced classifiers were constructed by combining CS to MNB conventional classifier. Regarding cosine-similarity, (6) is the function of conventional cosine-similarity, regarding MNB; (7) are the algorithms of conventional MNB and (8) is the algorithms of the proposed methodology. The arithmetic in (6) for calculating cosine-similarity is (6), where D_A and D_B are the two vectors that compared, and K is the number of words in each vector (vectors represent documents).

$$\boldsymbol{cos(\theta) = \frac{D_A . D_B}{|A||B|} = \frac{\sum_{i=1}^{K} D_{Ai} D_{Bi}}{\sqrt{\sum_{i=1}^{K} D_{Ai}{}^2}\sqrt{\sum_{i=1}^{K} D_{Bi}{}^2}}} \tag{6}$$

6. RESULTS AND DISCUSSION

In this section, in addition to detailing the research findings, a thorough discussion is also provided. Results can be presented in figures, graphs, tables that make the reader understand easily [29], [30]. Table 2 shows the demonstration form. More discussion will be made in the coming sub-sections.

$$\mathrm{S} = \begin{bmatrix} 1 & 0.612372440 & 0.288675130 \\ 0.61237244 & 1 & 0.353553539 \\ 0.28867513 & 0.353553539 & 1 \end{bmatrix}$$

We can easily evaluate:

$$Cos\ (D_Q, D_A) = \frac{0+0+1+0+0+1+0+0+1}{\sqrt{3}\sqrt{8}} = \frac{3}{\sqrt{24}} = 0.61237244$$

$$Cos\ (D_Q, D_B) = \frac{0+0+1+0+0+1+0+0+1}{\sqrt{3}\sqrt{4}} = \frac{3}{\sqrt{12}} = 0.28867513$$

$$Cos\ (D_A, D_B) = \frac{0+0+1+1+0+1+0+0+0}{\sqrt{8}\sqrt{4}} = \frac{3}{\sqrt{32}} = 0.353553539$$

Table 2. Demonstration form

	D_A	D_B	D_Q	D_A	D_B	D_Q
Introduction processing	0	1	0	1	0	0
in used to language	1	0	0	1	1	1
python language natural	1	1	1	1	0	0
programming	1	1	0	1	0	0
	0	1	0	1	0	1

D_A = ”in natural language processing is Python programming used”
D_B = ”introduction to languages Python”
D_Q = ”programming in Python”

CS is combined with estimated values provided by conventional classifiers such as MNB. In order to achieve CS between a test document and each category, the similarity between a test document and a category is combined with the estimated value for the category. This improves classifier performance. Multinomial naïve bayesian (MNB) uses a vector of words to represent a document *d* as in (7) [31], [32].

$$C_{\text{Predicted}}(d) = argmax_{c_j}\left[log\left(p(c_j)\right) + \sum_{k=1}^{n} f_k log(p(w_k \mid c_j))\right] \tag{7}$$

Where $p(c_j) = \frac{N_K}{N}$ and $p(w_k\ |c_j) = \frac{N_{cjk}+1}{N_{cj}\ + N_{all}}$. Where *d* is a test document, *n* is the number of words in *d*, c_j is the j^{th} category among all possible categories, w_k is the k^{th} word in *d*, and f_k is the frequency count of w_k. N_k is the number of all documents in c_j, *N* is the number of all documents in training documents. N_{cjk} is the number of w_k in c_j, N_{all} is the number of all unique words in training documents, and Nc_j is the number of all words in c_j. (8) represents the proposed methodology.

$$C_{\text{Predicted}}(d) = argmax_{c_j}\left[log\left(p(c_j)\right) + \sum_{k=1}^{n} f_k log(p(w_k \mid c_j))\right] + log((d, c_j)) \tag{8}$$

To test multiple documents and assign them to categories with the highest combined score (estimated value from multinomial naive bayes + cosine similarity score), we follow the next steps: i) Step 1: Preprocessing, ii) Step 2: Feature extraction as shown in (3), iii) Step 3: Training the MNB classifier as shown in (7), iv) Step 4: Calculating cosine similarity, v) Step 5: Combining cosine similarity and MNB as shown in (8), and vi) Step 6: Final prediction.

Typically, the cosine similarity value ranges from 0 to 1, where a high value indicates that data are well-matched to their own categories. Three categories: "Computers," "Programming," and "Technology." We have a training set with labeled documents in each category. To test three new documents and assign them to the category with the highest combined score. Calculate the cosine similarity scores and create a TF-IDF matrix using the training data. Assume that the cosine similarity scores between the test documents and training documents for each category are as in Table 3.

Table 3. Values of the cosine similarity

No		TF-IDF matrix			(MNB) scores	
	Computers	programming	Technology	Computer	programming	Technology
Document $_1$	0.2	0.1	0.3	0.4	0.3	0.3
Document $_2$	0.1	0.4	0.1	0.2	0.6	0.2
Document $_3$	0.3	0.2	0.5	0.1	0.2	0.7

Table 4 shows combining the scores for each document in CS with estimated values MNB. Based on the combined scores, assign the documents to the category with the highest score where each score based on their importance. Combine the scores by multiplying the MNB score by its weight and add it to the cosine similarity score multiplied by its weight. Assigning weights to each score based on their relative importance, it can assign a higher weight to the MNB score. In this example, Document1 is assigned to the "Computers" category because it has the highest combined score. Similarly, Document2 and Document3 are assigned to the "Programming" and "Technology" categories, respectively.

Table 4. Assigned documents to categories

Computer		Programming		Technology	
Document$_1$	0.4	Document$_2$	0.6	Document$_3$	0.7

7. CONCLUSION

Automatic text classification is a vital field of information retrieval. There are numerous issues and difficulties associated with text classification. In this study, we focus on two fundamental procedures for text document classification: partitioning the set of words and document categorization. Texts are divided into equivalence classes based on the cosine similarity classifier. One of the most important features of cosine similarity classifier is the speed and high efficiency in obtaining the best results, in terms of improving searches and making them faster and effective. As a result, in the domain of information retrieval, the position weight approach may be able to play an important role. Furthermore, using the concept of position weight, we present a method for selecting key terms from a list of words encompassing document classification, allowing large-scale information to be retrieved quickly and more effectively.

REFERENCES

[1] V. Lytvyn, V. Vysotska, O. Veres, I. Rishnyak, and H. Rishnyak, “Classification Methods of Text Documents Using Ontology Based Approach,” in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2017, pp. 229–240. doi: 10.1007/978-3-319-45991-2_15.

[2] K. H. Jihad, M. R. Baker, M. Farhat, and M. Frikha, *Machine Learning-Based Social Media Text Analysis: Impact of the Rising Fuel Prices on Electric Vehicles*. Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-27409-1_57.

[3] H. O. Ahmad and S. U. Umar, “Sentiment Analysis of Financial Textual data Using Machine Learning and Deep Learning Models,” *Informatica*, vol. 47, no. 5, pp. 153–158, May 2023, doi: 10.31449/inf.v47i5.4673.

[4] A. Onan, S. Korukoğlu, and H. Bulut, “Ensemble of keyword extraction methods and classifiers in text classification,” *Expert Systems with Applications*, vol. 57, pp. 232–247, Sep. 2016, doi: 10.1016/j.eswa.2016.03.045.

[5] A. K. Uysal and S. Gunal, “The impact of preprocessing on text classification,” *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, Jan. 2014, doi: 10.1016/j.ipm.2013.08.006.

[6] J. T. Pintas, L. A. F. Fernandes, and A. C. B. Garcia, “Feature selection methods for text classification: a systematic literature review,” *Artificial Intelligence Review*, vol. 54, no. 8, pp. 6149–6200, Dec. 2021, doi: 10.1007/s10462-021-09970-6.

[7] D. Agnihotri, K. Verma, and P. Tripathi, “Variable Global Feature Selection Scheme for automatic classification of text documents,” *Expert Systems with Applications*, vol. 81, pp. 268–281, Sep. 2017, doi: 10.1016/j.eswa.2017.03.057.

[8] A. Sridharan, R. A. A.S., and S. Gopalan, “A Novel Methodology for the Classification of Debris Scars using Discrete Wavelet Transform and Support Vector Machine,” *Procedia Computer Science*, vol. 171, pp. 609–616, 2020, doi: 10.1016/j.procs.2020.04.066.

[9] A. Qazi and R. H. Goudar, “An Ontology-based Term Weighting Technique for Web Document Categorization,” *Procedia Computer Science*, vol. 133, pp. 75–81, 2018, doi: 10.1016/j.procs.2018.07.010.

[10] Y. Lu and Y. Chen, “A Text Feature Selection Method Based on the Small World Algorithm,” *Procedia Computer Science*, vol. 107, pp. 276–284, 2017, doi: 10.1016/j.procs.2017.03.102.

[11] C. Liu, W. Wang, G. Tu, Y. Xiang, S. Wang, and F. Lv, “A new Centroid-Based Classification model for text categorization,” *Knowledge-Based Systems*, vol. 136, pp. 15–26, Nov. 2017, doi: 10.1016/j.knosys.2017.08.020.

[12] X. Hu and B. Wu, “Automatic Keyword Extraction Using Linguistic Features,” in *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW’06)*, Dec. 2006, pp. 19–23. doi: 10.1109/ICDMW.2006.36.

[13] D. Fragoudis, D. Meretakis, and S. Likothanassis, “Integrating feature and instance selection for text classification,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Jul. 2002, pp. 501–506. doi: 10.1145/775047.775120.

[14] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining (Second Edition)*. Pearson, 2019.

[15] G. Zeng, “On the confusion matrix in credit scoring and its analytical properties,” *Communications in Statistics - Theory and Methods*, vol. 49, no. 9, pp. 2080–2093, May 2020, doi: 10.1080/03610926.2019.1568485.

[16] B. Baharudin, L. H. Lee, and K. Khan, “A Review of Machine Learning Algorithms for Text-Documents Classification,” *Journal of Advances in Information Technology*, vol. 1, no. 1, Feb. 2010, doi: 10.4304/jait.1.1.4-20.

[17] D. M. Diab and K. M. El Hindi, “Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification,” *Applied Soft Computing*, vol. 54, pp. 183–199, May 2017, doi: 10.1016/j.asoc.2016.12.043.

[18] A. Jain and R. D. Mishra, “An Effective Approach for Text Classification,” *International Journal of Research in Engineering and Technology*, vol. 05, no. 06, pp. 24–30, Jun. 2016, doi: 10.15623/ijret.2016.0506005.

[19] H. Guan, J. Zhou, and M. Guo, “A class-feature-centroid classifier for text categorization,” in *Proceedings of the 18th international conference on World wide web*, Apr. 2009, pp. 201–210. doi: 10.1145/1526709.1526737.

[20] L. Jiang, S. Wang, C. Li, and L. Zhang, “Structure extended multinomial naive Bayes,” *Information Sciences*, vol. 329, pp. 346–356, Feb. 2016, doi: 10.1016/j.ins.2015.09.037.

[21] T. A. Wotaifi and B. N. Dhannoon, “Improving Prediction of Arabic Fake News Using Fuzzy Logic and Modified Random Forest Model,” *Karbala International Journal of Modern Science*, vol. 8, no. 3, pp. 477–485, Aug. 2022, doi: 10.33640/2405-609X.3241.

[22] J. Su, J. Sayyad-Shirabad, and S. Matwin, “Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes,” 2011.

[23] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, “Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification,” in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, Apr. 2019, pp. 593–596. doi: 10.1109/ICACTM.2019.8776800.

[24] T. T. Nguyen, K. Chang, and S. C. Hui, “Supervised term weighting centroid-based classifiers for text categorization,” *Knowledge and Information Systems*, vol. 35, no. 1, pp. 61–85, Apr. 2013, doi: 10.1007/s10115-012-0559-9.

[25] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu, “Supervised and Traditional Term Weighting Methods for Automatic Text Categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721–735, Apr. 2009, doi: 10.1109/TPAMI.2008.110.

[26] A. A. Jalal and B. H. Ali, “Text documents clustering using data mining techniques,” *International Journal of Electrical and*

Computer Engineering (IJECE), vol. 11, no. 1, p. 664, Feb. 2021, doi: 10.11591/ijece.v11i1.pp664-670.

[27] S. Hariharan and R. Srinivasan, "A Comparison of Similarity Measures for Text Documents," *Journal of Information & Knowledge Management*, vol. 07, no. 01, pp. 1–8, Mar. 2008, doi: 10.1142/S0219649208001889.

[28] J. Wang and Y. Dong, "Measurement of Text Similarity: A Survey," *Information*, vol. 11, no. 9, p. 421, Aug. 2020, doi: 10.3390/info11090421.

[29] H. Margossian, G. Deconinck, and J. Sachau, "Distribution network protection considering grid code requirements for distributed generation," *IET Generation, Transmission & Distribution*, vol. 9, no. 12, pp. 1377–1381, Sep. 2015, doi: 10.1049/iet-gtd.2014.0987.

[30] O. Núñez-Mata, R. Palma-Behnke, F. Valencia, A. Urrutia-Molina, P. Mendoza-Araya, and G. Jiménez-Estévez, "Coupling an adaptive protection system with an energy management system for microgrids," *The Electricity Journal*, vol. 32, no. 10, p. 106675, Dec. 2019, doi: 10.1016/j.tej.2019.106675.

[31] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial Naive Bayes for Text Categorization Revisited," in *AI 2004: Advances in Artificial Intelligence*, 2005, pp. 488–499.

[32] S. Xu, Y. Li, and Z. Wang, "Bayesian Multinomial Naïve Bayes Classifier to Text Classification," in *Advanced Multimedia and Ubiquitous Engineering*, 2017, pp. 347–352.

BIOGRAPHIES OF AUTHORS

Sura I. Mohammed Ali iD SC holds a master of computer science degree from Cairo University, Egypt in 2015. She also received his B.Sc. (computer science) from University Qadisiyah, Iraq in 2006. She is currently a lecturer at Computer Science Department, Al-Muthanaa University, Iraq. Her research includes information Retrieval, SIS, machine learning, and image processing. She has published 10 papers in international journals and conferences, from 2014 to 2021. She can be contacted at email: suraibraheem@mu.edu.iq.

Marwah Nihad iD SC holds a master of computer science degree from Department of Computer Science, Faculty of Computers and Information, Cairo University, Egypt in 2015. She also received his B.Sc. in computer science from Department of Computer Science, Faculty of Science, University of Kirkuk, Iraq in 2007. She is currently a lecturer in the Faculty of Science, and the College of Computer Science and Information Technology, University of Kirkuk, Iraq. She has published research papers in prestigious international journals, and conference proceedings. Marwah's interests are big data, data management, machine learning, and internet of things (IoT). She can be contacted at email: marwah.nihad@uokirkuk.edu.iq.

Dr. Hussien Mohamed Sharaf iD SC received his masters in 2006 and Ph.D. in 2011 in computer science from Faculty of Computers and Information, Cairo University, Egypt. He has been working as a full time Ph.D. lecturer in Suez university since 2016. He has published research papers in prestigious international journals, and conference proceedings. Research interests include big data, IoT, machine learning, soft computing techniques, security applications, and bioinformatics. He can be contacted at email: h.sharaf@suezuni.edu.eg.

Dr. Haitham Farouk iD SC holds his M.Sc. (2005) and Ph.D. (2015) in computer science from Faculty of Computers and Information, Helwan University and Cairo University in Egypt, respectively. He is currently a computer science at Faculty of Computers and Information, Suez University, Egypt. His research includes machine learning, image processing, GIS, satellite imageries analysis, remote sensing, IoT, and big data. He can be contacted at email: h.farouk@suezuni.edu.eg.