

Received October 29, 2021, accepted November 27, 2021, date of publication December 1, 2021, date of current version December 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3132022

Detecting Fake News in Social Media Using Voting Classifier

EMAN ELSAEED¹, OSAMA OUDA^{2,3}, (Member, IEEE),
MOHAMMED M. ELMOGY¹, (Senior Member, IEEE),
AHMED ATWAN^{1,4}, AND EMAN EL-DAYDAMONY¹

¹Department of Information Technology, Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt

²College of Computer and Information Sciences, Jouf University, Sakakah 72388, Saudi Arabia

³Faculty of Computer and Information Sciences, Mansoura University, Mansoura 35516, Egypt

⁴Faculty of Computing and Information Science, Northern Border University, Arar 73222, Saudi Arabia

Corresponding authors: Mohammed M. Elmoghy (melmoghy@mans.edu.eg) and Osama Ouda (ouoda@ieee.org)

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT The availability of social media, blogs, and websites to everyone creates a lot of problems. False news is a critical issue that can affect individuals or entire countries. Fake news can be created and shared all over the world. The 2016 presidential election in the United States illustrates that problem. As a result, controlling social media is essential. Machine learning algorithms help to detect fake news automatically. This article proposes a framework for detecting fake news based on feature extraction and feature selection algorithms and a set of voting classifiers. The proposed system distinguishes fake news from real news. First, we preprocessed the data taking unnecessary characters and numbers and reducing the words in the dictionary (lemmatization). Second, we extracted some important features by using two types of feature extraction, the term frequency-inverse document frequency technique and the document to vector algorithm, a word embedding technique. Third, the extracted characteristics were reduced with the help of the chi-square algorithm and the analysis of the variance algorithm. We used three data sets that are published online: Fake-or-Real-News, Media-Eval, and ISOT. We used five performance metrics to evaluate the proposed framework: accuracy, the area under the curve, precision, recall, and f1-score. Our system achieved 94.6% of accuracy for the Fake-or-Real dataset. For the Media-Eval dataset, the system achieved 92.3% of accuracy. For the ISOT dataset, the system achieved 100% of accuracy. We contrast the proposed framework with several other classification algorithms. The experimental results show that the proposed framework outperforms the existing works in terms of accuracy by 0.2% for the ISOT dataset.

INDEX TERMS Fake news, news classification, voting classifier, term frequency-inverse document frequency, chi-square.

I. INTRODUCTION

One of the consequences of technology is fake news. It is misinformation or misleading information offered as facts that can affect a person's opinion. This false information has several goals; organizations can use it for financial purposes (e.g., Facebook pages used it to spread fake news leading to specific ads) or for political purposes. Compared with Google, Twitter, and webmail such as Yahoo and Gmail, Facebook is the worst media platform for pervasive fake news. A group of researchers tracked the internet usage of more than 3000 Americans in the run-up to the 2016

presidential election. They discovered that Facebook is the highest reference for untrusted news sources in more than 15% of cases. However, for only 6% of the time, Facebook guided users to authoritative news sites. The authors observed that the untrusted website visits are not observed by 3.3% untrusted news versus 6.2% trust news for Google, or 1% untrusted versus 1.5% trust news for Twitter [1].

Spreading false news is roughly as dangerous as spreading the virus. People are currently encountering fake coronavirus news daily. This fake information triggers fear and panic among people. Therefore, there is a need for ways to fact-check news.

Finland, for example, is leading the fight to tackle fake news through education. They teach primary school students

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

how to combat false news and develop media literacy skills. This has been a priority in Finland's education agenda due to Russia's false stories targeting the country. Thus, Finland ranks first in media literacy compared to the UK, France, and Italy [2].

Fake information can be spread in the form of text, video, pictures, and audio via social media networks such as Facebook and Twitter [3]. The fake news problem has existed for a long time. People used to believe such news even if it was false [4]. Therefore, detecting fake news can be difficult, especially with no supervising body on the internet [5]. The growth of concern regarding the detection of unreliable news is recent. It is difficult for a human to manually detect news, even with the existence of all topics shown on social media. Therefore, there is a need for an efficient way to help us distinguish false information from true ones posted on social media. One of the efficient ways is to classify the news using machine learning (ML) algorithms.

There are two delineations for automatically identifying fake news: news and social context [6], [7]. The news content-based approach focuses on extracting unique features from fake news content. Because misinformation tries to spread false claims, knowledge-based approaches use international sources to fact-check the truthfulness in any news content. Style-based approaches uncover imitation information by spotting manipulators in the writing style.

Social context-supported approaches use someone else's party engagements as a secondary assemblage to exploit and find imitation information. Stance-based approaches use users' viewpoints from relevant blog content to conclude the veracity of original news articles. The credibility values of relevant interpersonal media posts verify any news. Imitative news discovery on social media is a new research area. The research directions are outlined in four scenes: data-oriented, feature-oriented, model-oriented, and application-oriented.

Data-oriented is about different aspects of data. The feature-oriented goal is to explore effective features for detecting false information. Model-oriented is used for building more particular models. Application-oriented goes beyond phony tidings discovery, much as diffusion and involvement.

Text classification is a popular task in natural language processing (NLP). It makes the program classify the text document based on predefined classes. These tasks can be done by using ML and deep learning (DL) algorithms. DL is a sub-field of ML that requires massive data to make a sensible decision. This paper enhances the performance of the conventional ML algorithm in detecting fake news because the dataset utilized didn't have a large enough amount of data to feed the DL algorithm [8].

This work proposed a system for differentiating fake news from real news using ordinary ML algorithms and voting classifier techniques. The algorithms used are naïve Bayes (NB), linear support vector machine (LSVM), logistic regression (LR), random forest (RF), passive-aggressive (PA), and stochastic gradient descent classifier (SGD). Online

publication of the experimental results obtained from three datasets.

The contributions of the paper can be summarized into the following points:

- Six different ML models and the voting classifier technique were trained to compare the differences in performance between them.
- To enhance the results, we preprocessed the data by removing all unnecessary characters.
- The term frequency-inverse document frequency (TF-IDF) algorithm and the word embedding document to vector (DOC2VEC) algorithm was used for feature extraction. We used the chi-square algorithm and analysis of variance (ANOVA) algorithm to reduce the features.
- We applied three experiments which are TF-IDF with chi-square, TF-IDF with ANOVA, and DOC2VEC for feature extraction and selection. Then, we compared the results for these three experiments.

The remainder of the paper is organized as follows. Section II presents a review of the related studies. Section III illustrates a proposed framework in detail. Section IV presents the experimental results of the proposed system. Finally, Section V introduces the conclusion and future work.

II. RELATED WORK

The spread of misleading information on social media leads researchers to do their best to solve this problem. We present some of the previous work in this direction. Ahmed *et al.* [3] proposed using of n-gram model to differentiate between fake and real news. They proposed to generate several sets of n-gram from training data to differentiate between false and true news. They used various features of the n-gram baseline established on words. They preprocessed the dataset by stemming and removing the stop words. They use TF-IDF for text feature extraction. They use six ML algorithms: SGD, k-nearest neighbor (KNN), support vector machines (SVM), LSVM, and decision trees (DT) on three online datasets. They achieved an accuracy (ACC) of 87.0% in differentiating fake from real news using n-gram features and LSVM algorithm. However, trying other algorithms can achieve better performance, such as PA.

Keskar *et al.* [4] proposed an N-gram analysis using the DT ML technique to detect fake news. The proposed system was tested using live stream data collected from Twitter. The ACC achieved was 70.0%. For better performance, several ML algorithms such as SVM and PA can be used with N-Gram. Bali *et al.* [5] compared the performances of seven ML algorithms. They used three standard datasets. To evaluate their work, they used a performance measure. Their system achieved a mean ACC of 88.0% and 91.0% of f1-score. Nevertheless, using linguistic features can enhance the ACC.

Wang [9] came with a new dataset, liar, which can be used for automated fake news detection systems, this dataset is bigger, so it does not have the whole article. It only had

12800 short-labeled statements taken from the political fact website that was manually labeled. They used ML algorithms in their dataset, and the evaluation results show that they achieved 27.0% of ACC. However, trying algorithms like SGD and PA could achieve better ACC.

Yang *et al.* [10] proposed a framework for detecting real news and users' credibility using unsupervised learning and probabilistic graphical. Their system achieved an ACC of 75.9%. However, incorporating features of news content and user profiles can improve the performance of their unsupervised model. Shu *et al.* [11] made the fake news tracker system for fake news problems. First, they collect news and social context automatically to build their dataset. Then, they extract features from the dataset and use ML algorithms to differentiate between false and real news. The experimental results showed that their system achieved an ACC of 74.2%. However, using other available features in the dataset like favorites and re-tweets could enhance the performance.

Wu and Liu [12] proposed a system called trace miner for classifying social media posts. To detect the pathways of a message, they used the LSTM-RNN algorithm. Their proposed system achieved about 93.8% of the f1-score. However, using other measures such as ACC and the area under the curve (AUC) to view their point could be better. Zhang *et al.* [13] introduced a fake-detector inference model for automated fake news detection; they extract a distinct set of latent features from the text. This system builds a deep diffusive network for learning the impersonation of news articles and creators of these articles and subjects. They experimented on a real-world fake news dataset to compare the system they built with various models. The ACC score obtained by their system is about 63.0%. But they could use other datasets to evaluate the proposed method then compare the results of the two datasets.

Gilda [14] created an application for detecting fake news using an open-source dataset. They use TF-IDF for extracting features and probabilistic context-free grammar detection on a corpus of about 11,000 articles. Their experiments were performed on several classification algorithms: SVM, SGD, gradient boosting (GD), bounded decision trees (BDT), and RF. They found that when using the TF-IDF method with the SGD algorithm, the ACC achieved was 77.2%. However, many feature extraction algorithms such as Word2Vec may enhance the results.

Ko *et al.* [15] proposed a cognitive system using backtracking to detect fake news. Their results were an 85.0% detection rate. However, they did not clear how to detect fake news and subjective posts. Atodiresei *et al.* [16] proposed a system for identifying fake users and news on Twitter. Their system will receive a link to a tweet from a user and then compute the tweet credibility, also some statistics such as emotions. However, they didn't clear what are the measure metrics they used to evaluate their work.

Pan *et al.* [17] proposed a B-TransE model system for detecting fake news according to news content using knowledge graphs. The experimental results of their works show

that some of their approaches achieve above 80.0% of f1-score. The results could enhance if they combined two approaches together, such as content-based approaches with style-based approaches. Potthast *et al.* [18] proposed a new method for assessing style similarity across text categories using meta-learning. They show that fake news can be differentiated by its style. The experimental results show that their system achieved about 74.0% of ACC. Feature extraction algorithms like TF-IDF could enhance the results.

Ghafari *et al.* [19] presented TDTrust, a context-aware trust prediction system based on Tensor Decomposition that considers knowledge to determine an entity's condition. They used a real-world dataset to investigate novel algorithms that use social context factors. Nevertheless, they did not evaluate the proposed system using measure performance metrics such as precision, recall, and f1-score. Fernandez-Reyes and Shinde [20] proposed using different deep neural algorithms to detect fake news in the political area. The experimental results showed that the proposed system achieved 48.5% of ACC. Evaluating the proposed framework with several measures' performance metrics such as precision, recall, and f1-score to view their points could help the readers.

Grandis *et al.* [21] proposed using the Multi-Criteria Decision Making paradigm for fake news detection. To evaluate their results, several ML algorithms were used. Their proposed system achieved about 84.0% of ACC. However, aggregation functions, such as feature interaction, can enhance performance. Ksieniewicz *et al.* [22] focused on fake news detection in articles published online. They used ML for detecting fake news. They confirmed that the chosen ML algorithms could distinguish between fake and real information. The system proposed in this study achieved an average ACC of (74.3%). Nevertheless, they did not evaluate the results using different performance measures such as precision, recall, and f1-score.

Crammer *et al.* [23] presented a set of margin-based online learning algorithms for several prediction tasks. This online algorithmic framework solved numerous prediction problems, including classification and sequence prediction. Hakak *et al.* [24] proposed an ensemble classification model to detect fake news. They extracted the features from the data set. Then, they classified the data using three models Decision Tree, Random Forest, and Extra Tree Classifier. They achieved an ACC of 99.8% for the ISOT dataset. However, using algorithms like SVM and PA may enhance the ACC of the lair dataset.

Madani *et al.* [25] focused on fake news that tweeted during the CORONA virus. They proposed a classification approach based on natural language processing, ML, and deep learning. They used an RF algorithm and achieved 79.0% of ACC. However, as compared to other systems, the ACC they achieved is low. Nasir *et al.* [26] proposed a novel hybrid DL system that gathers convolutional and recurrent neural networks. They used two data sets ISOT and FA-KES. They achieved 99.0% of ACC for the ISOT dataset. In comparison to other systems, their system has a good ACC.

Jiang *et al.* [27] used hold-out cross-validation to compare the performance of five machine learning models and three DL models on two fake and real news datasets of various sizes. They used term frequency, term frequency-inverse document frequency, and embedding approaches to get text representation for machine learning and DL models. They employed accuracy, precision, recall, and F1-score as evaluation metrics, as well as a corrected version of McNemar's test to see if models' performance differed significantly. Then they presented their innovative stacking model, which obtained testing accuracy of 99.94% on the ISOT dataset and 96.05% on the KDnugget dataset.

Saleh *et al.* [28] presented unique methodologies based on ML and DL for a fake news identification system to combat the false news epidemic. The goal was to develop the best model for achieving high accuracy results. In order to detect bogus news, they created an efficient Convolutional Neural Network model (OPCNN-FAKE). They used four fake news benchmark datasets to evaluate the performance of the OPCNN-FAKE with the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and the six normal ML techniques: DT, LR, KNN, RF, SVM, and NB. The parameters of ML and DL were optimized using grid search, and hyperopt optimization approach, respectively. In addition, standard ML models employed N-gram and TF-IDF to extract features from the benchmark datasets, while DL models used Glove word embedding to encode features as a feature matrix. The results were validated using accuracy, precision, recall, and the f1 score.

The previous studies have several limitations, which are discussed further below. Some of the current works did not use algorithms like SGD and PA to achieve better ACC. Other works discard using some important features. This feature could enhance the performance. Other works did not use any measures to view their point. There are not enough datasets or standard data sets used in other studies to evaluate the proposed method. Several studies on detecting fake news faced an issue of low accuracy in distinguishing real news from fake news.

The goal of this work is to learn how to handle and preprocess text data. To be able to compare the results of various feature extraction and reduction algorithms. Furthermore, the goal of this work was to improve the performance of traditional ML algorithms for the purpose of detecting fake news.

In summary, this framework proposes an ensemble-voting classifier to beat the limitations of the previous works. The novelty of this framework can be summarized in a set of points. First, the study used three published datasets, making it easy to compare the results with other works. Second, the framework is based on a set of six different classifier algorithms. Third, we preprocessed the data by removing unnecessary characters, which helps to enhance the performance. Fourth, to avoid overfitting, we split the data using the k-fold cross-validation method. Fifth, feature extraction algorithms reduce the computational cost, redundant and

irrelevant words that the machine cannot understand. Sixth, the feature selection algorithms select the best features for better performance. Seventh, the ML algorithms were combined to form the ensemble method, and we used a voting function to get the final decision based on the base classifiers. Finally, we used five metrics to evaluate the proposed framework: ACC, AUC, precision, recall, and f1-score. The proposed approach can detect any news in text format with high ACC.

Table 1 represents a comparison of the performance for some current studies.

III. PROPOSED FRAMEWORK

The suggested framework is shown in this section of the article. Three publicly available datasets are used to train the system. To begin, clean up the data by deleting any superfluous characters or digits. After that, 10-fold cross-validation was applied to partition the dataset. TF-IDF and DOC2VEC are utilized to extract features from the dataset [22], [29]. The chi-square discretization algorithm and the ANOVA algorithm are used to choose the best features. Finally, the classification algorithms are fed the features. An ensemble voting classifier is utilized for the best results. The proposed framework is illustrated in Figure 1.

A. DATA PREPROCESSING

Preprocessing the data is required before extracting the features. This data may contain special characters, numbers, and unnecessary space. First, remove all special characters, also known as non-word characters and numbers. Then we remove all single characters. For example, when a punctuation mark is removed from Alice's and replaced with space, Alice and a single character "s" have no meaning in the text. Also, replace every single character from the beginning of the text with a single space; however, this will result in numerous spaces being replaced with a single space. The final step in the preprocessing process is lemmatization, which reduces words to their dictionary root form. Computers, for example, will be reduced to a computer. The purpose of the lemmatization stage is to avoid repeating features [30].

B. DATASET SPLIT

Splitting the dataset into training and testing sets is important for evaluating the classification models. In this step, the datasets are split into training and testing parts. The training part is used to train the model, and it is called a training set. The other part is used to test the classification model, and it is called a test set. For better performance, the test set is usually smaller than the training set. The method we used to split the data is called k-fold cross-validation [31]. Cross-validation is robust against overfitting. This method resamples the dataset depending on a parameter called *k* — *fold*. The (*k* parameter) refers to the number of groups that the dataset will split into. In this experiment, we used 10-fold cross-validation [32], [33].

TABLE 1. A comparison of some current studies with respect to performance.

Study	Year	Algorithm's name	Brief Methodology	Target/ class	Dataset	Performance (%)	Highlights and limitation
Ahmed et al. [3]	2017	N-gram Model with SGD, SVM, LSVM, KNN, LR, and DT.	Using of n-gram model to differentiate between fake and real news. They proposed to generate several sets of n-gram from training data to clarify false and true news.	Binary/ Fake news Truthful news	Collected from Reuters.com and kaggle.com.	ACC= 92.0	They proposed a false news detection model based on n-gram analysis and a variety of feature extraction approaches. However, they did not extract the characteristics using a word embedding approach.
Gilda [14]	2018	SVM, SGD, GD, BDT, and RF.	He made an application for detecting fake news using an open-source dataset. He applied TF-IDF for extracting features. He used probabilistic context-free grammar detection (PCFG).	Fake / true	From OpenSources.co	ACC= 77.0	To test the dataset, they used two feature extraction algorithms and various classification models. Nevertheless, they assessed the models using absolute probability thresholds, which might not be the most reliable method for poorly calibrated probability scoring models.
Potthast et al. [18]	2018	Unmasking Style Categories	They proposed a new assessing style similarity between text categories through meta-learning.	Fake/real	BuzzFeed Fake News	ACC= 74.0	They're comparing the writing style of fake news writers in connection to hyper-partisan content. But They won't be able to detect false news only based on style analysis.
Yang et al. [10]	2019	Unsupervised learning, Gibbs Sampling	They used unsupervised learning and probabilistic graphical for detecting the real news and the users' credibility.	True /fake	LIAR, BuzzFeed News	ACC= 76.0	They proposed an efficient collapsed Gibbs sampling strategy for inferring news facts and users' reliability without any labeled data. However, in the false news detection model, they did not use news content or user profiles.
Shu et al. [11]	2019	Semi-supervised learned features	They automatically collected the news. Then, they extracted the features from the dataset and applied ML algorithms to classify false and real news.	True /fake	PolitiFact, BuzzFeed	ACC= 74.0	They created a system that can collect data for news stories and their social context automatically. However, they did not use all the dataset's features.
Zhang et al. [13]	2019	Deep diffusive network	They extracted a clear group of latent features from the text. This system builds a deep diffusive network for learning the impersonation of news articles and creators of these articles and subjects.	Real/fake	PolitiFact	ACC= 63.0	They developed a new computerized methodology for determining the veracity of false news. Based on the extraction of explicit and latent properties from textual data. Unfortunately, when compared to other studies, the results are disappointing.
Ksieniewicz et al. [22]	2019	Machine-learning algorithms	They used ML algorithms (ML) for detecting fake news.	Fake/real	Articles published online.	ACC= 74.0	They applied classifier ensemble methods to the challenge of detecting fake news. They used the Random Subspace approach. Unfortunately, the results are poor when compared to other studies.
Grandis et al. [21]	2019	Multi-Criteria Decision-Making using ML algorithms	They used the multi-criteria decision-making (MCDM) paradigm for fake news detection.	True /fake	CREDBANK Dataset	ACC= 84.0	For fake news detection, they adopted a Multi-Criteria Decision Making (MCDM) paradigm. However, they did not consider the aggregation functions, and the outcomes were unsatisfactory when compared to other studies.
Keskar et al. [4]	2020	N-gram analysis using decision trees (DT)	They used N-gram analysis with the (DT) ML technique to detect fake news on live stream data collected from Twitter.	Fake/real	Live stream data collected from Twitter.	ACC= 70.0	The Flume technology is used to extract data from websites. Once the data has been retrieved and stored in the necessary format, it can be further analyzed to determine whether it is fake or true news. Unfortunately, when compared to other studies, the results are poor.
Wu& Liu [12]	2018	SVM, XGBoost, TM(DeepWalk), TM(LINE), Trace Miner	They proposed a system for classifying social media posts, Trace Miner. To detect the pathways of a message, they used the LSTM-RNN algorithm.	Real/fake news	Data collected from Twitter.	f1-score = 93.8	They proposed Trace-Miner, a new method for classifying social media messages based on diffusion traces in social networks. However, they did not utilize any other metrics beyond the f1-score as a statistic.

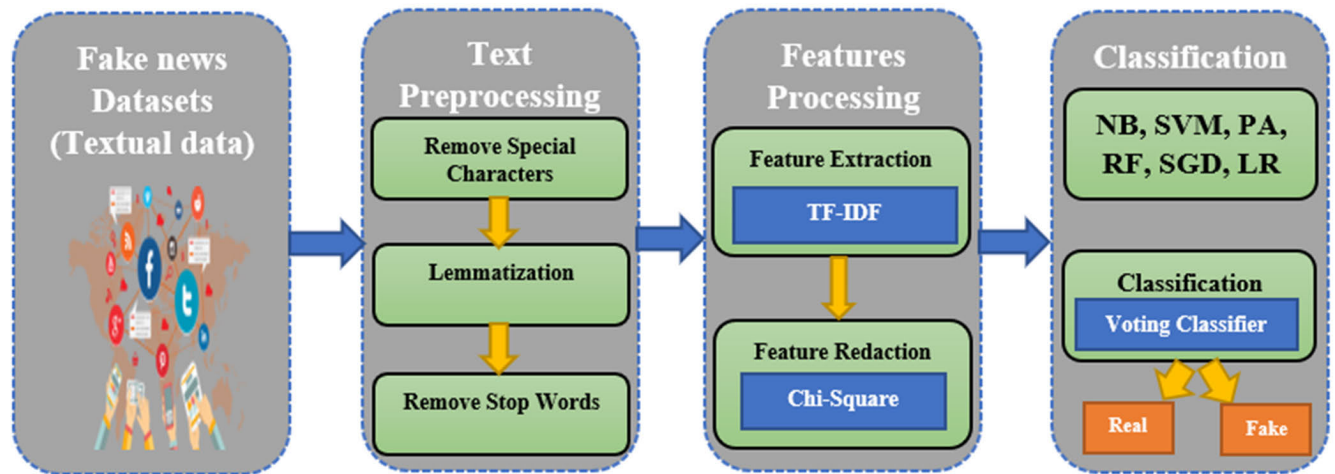


FIGURE 1. The proposed framework.

C. FEATURE EXTRACTION

This study works on textual data such as articles and posts with a large number of words and characters, resulting in high computational costs. This data is also redundant, has irrelevant words and meaning that the machine can't understand. Most of the text is unstructured and has high dimensionality. For that, it can be difficult to apply many classifiers to the text data. Thus, we need first to extract the most distinguishing features from the text so the dimensionality will reduce. Besides, it is important to take out a list of words from the text data and then transfer it into a feature set that the machine can use; this process is called feature extraction from text [34]. Using feature extraction algorithms also helps to enhance the performance of the ML algorithms. There are many ways to extract features from the text. In this paper, we used the TF-IDF algorithm and word embedding DOC2VEC algorithm.

TF-IDF algorithm is usually used to extract features in ML tasks because of its simplicity and robustness. TF-IDF algorithm split into two terms TF, which means how many words are in the current post, which is given by Eq. (1).

$$TF(word) = \frac{\text{number of repeated words appear in the document}}{\text{total number of words in the document}} \quad (1)$$

where *IDF* refers to how necessary any terms are in all posts. *IDF* gave a score to words, which is given by Eq (2). This score can highlight a useful or necessary word [35].

$$IDF(word) = \frac{\log(\text{total amount of documents})}{\text{number of document where the word appear}} \quad (2)$$

The stop-words parameter ignores the English stop-words such as a, about, above, after, again, at, as, and are. The (Min_df parameter), which we set to 5, means that the minimum number of posts containing the features means that we

only require the features that appear in at least five posts. The (max_df) is the same, but it will be set to 0.7 because the fraction corresponds to a percentage. This means we need the features that only appear in 70.0% of all the posts. Words that appear in almost every post are unsuitable for classification because they do not supply any unique features from the posts to be removed. Table 2 shows some examples of extracted features and their frequencies using the TF-IDF algorithm.

Algorithm 1 Text Preprocessing

Input: uncleaned documents/posts/tweets

Output: processed/cleaned documents/posts/tweets

Procedure:

- 1) For each document \in uncleaned document
- 2) Delete all the special characters
- 3) Delete all single characters
- 4) Delete single characters from the begin
- 5) Substituting multiple spaces with a single space
- 6) Delete prefixed 'b'
- 7) transforming to Lowercase
- 8) Lemmatization
- 9) End for of uncleaned document

Word embedding algorithms are a word representation type that gives the words of the same meaning the same representation. There are many types of word embedding, such as word2vector, Glove, and DOC2VEC. Word2vector algorithm takes every word in the article to classify and convert it to a unique vector [36]. Glove algorithm is the expansion of word2vec technique for better learning word vector. DOC2VEC algorithm uses three layers of the deep neural network to measure the document context and bind similar context together [37]. Figure 2 shows the histogram of extracted features using the DOC2VEC algorithm.

D. FEATURE REDUCTION

Feature reduction or selection is the process of extracting the most pertinent features from a dataset. Following that, ML algorithms will be used. This improves the performance of the classification model. Feature selection methods are used to reduce the risk of overfitting and training time.

TABLE 2. Examples of some extracted features and their frequencies using the TF-IDF algorithm.

Extracted Features	Frequency
Report	13082
Violation	16768
Think	15651
Office	10836
Fun	6713
Come	3235

Algorithm 2 TF Algorithm

Input:

- 1) D: documents/posts/tweets
- 2) T: the unique term in all documents

Output: weight matrix

Procedure:

- 1) For each $t_i \in T$ do
- 2) For each document $d_j \in D$ do
- 3) W_{ij} = number of appearances of term t_i in document d_j
- 4) End for of document
- 5) End for of term

Feature selection methods are classified into three groups which are: filter, wrapper, and embedded. The filter methods keep the same meaning of the selected features as the original features. These methods do not depend on the performance of any classifier. Therefore, in this paper, we used two algorithms of the filter method for feature selection. The first one is the chi-square algorithm which is used for categorical features. We calculated the chi-square between each feature in the dataset and target (label). Then we select the required number of features with the best score of chi-squares. The chi-square score is given by Eq. 3.

$$\chi^2 = \frac{(\text{observed frequency} - \text{expected frequency})^2}{\text{expected frequency}} \quad (3)$$

where the *observed frequency* is the number of class labels' observations. The *expected frequency* would be the class label's number of observations if there were no relevance amidst the feature and the target [38]. The second one is the ANOVA algorithm, a statistical technique used to check the meaning of two or more distinct sets.

E. CLASSIFICATION ALGORITHMS

The system will be trained using six distinct ML algorithms in this step, which will be discussed in the subsections below.

Algorithm 3 TF-IDF Algorithm

Input:

- 1) T: the unique term in all documents
- 2) D: documents/posts/tweets
- 3) weight matrix from TF step

Output:

TF-IDF weight for each term

Procedure:

- 1) for each term $t_i \in T$ do
- 2) for each document $d_j \in D$
- 3) If TF_{ij} not equal zero, then DF_i++
- 4) End for of document
- 5) $Idf = \log(D/DF_i)$
- 6) End for of term
- 7) For each term t_i do
- 8) For each document $d_j \in D$ do
- 9) $Tf-IDF = TF_{ij} * IDf_i$
- 10) End for of Document
- 11) End for of term

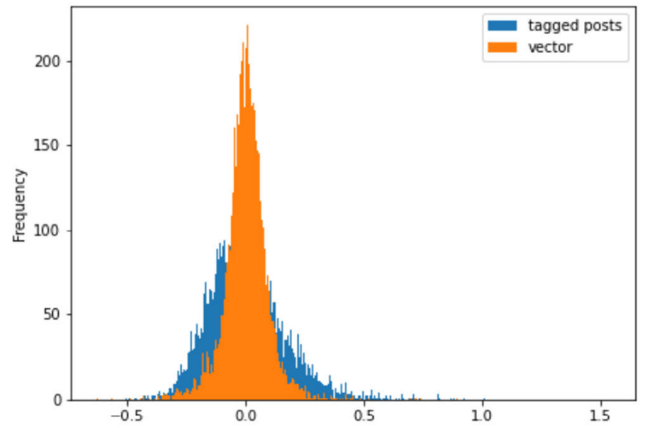


FIGURE 2. The extracted features in vector form using doc2vec algorithm.

1) NAÏVE BAYES

Using Bayes Theorem, the NB classifier divides the data into classes based on their probability. When using the NB classifier, all predictors are believed to have the same influence on the class outcome. We can find the probability of an event X (which refers to fake news) when event D is true using Bayes Theorem. The formula of the Bayes Theorem is given by Eq. (4).

$$P(X|D) = \frac{P(D|X)P(X)}{P(D)} \quad (4)$$

where the variable X is the class label (fake/real), the variable D represents the word/features, $P(X|D)$ is the conditional probability that news articles are fake when that word D appears, $P(D|X)$ is the conditional probability of finding the word D in fake news articles, $P(X)$ is the overall probability that a given news article is fake and $P(D)$ is predictor prior probability.

Following that, the probability of the fact that a given news article is fake is calculated by combining the probabilities $P(X|D)$, which is known for each word in the news article, given by Eqs. (5-7).

$$P_1 = P(X | D_1) \cdot P(X | D_2) \cdot P(X | D_3) \cdots P(X | D_n) \quad (5)$$

$$P_2 = 1 - (P(X | D_1) \cdot P(X | D_2) \cdot P(X | D_3) \cdots P(X | D_n)) \quad (6)$$

$$P = \frac{P_1}{P_1 + P_2} \quad (7)$$

where X is the class label (fake/real), the variable D represents the word/features, n is the total number of words in the news article, P_1 is a product of the probability that news articles are fake given that they contain a specific word for all the words in the news article, P_2 is the same as P_1 , but complement probabilities are used instead, P is the overall probability of the fact that a given news article is fake and finally $P(X | D_1) \cdot P(X | D_2) \cdot P(X | D_3) \cdots P(X | D_n)$ is the conditional probability that a news article is fake given that the words D_1, D_2, D_3, D_n appear in it, respectively [39].

2) SUPPORT VECTOR MACHINES

Support vector machine is a supervised ML model used for classification and regression. Its goal is to find the best hyperplane that divides a dataset into two classes. There are three important concepts in the support vector machine: data points, hyperplane, and the margin. Data points are the support vectors closest to the hyperplane, and they can find the correct separate line. A hyperplane is a decision plane that divides a set of objects. A margin is defined as a gap between two lines on the closest data points of distinct classes; a large margin is desirable, while a small one is undesirable [40].

In applications such as document classification and fake news detection, the dimension of feature vectors is sufficiently large to find separable linear classifiers that can separate training feature vectors into non-overlapping regions (classes). Like perceptron and logistic regression, SVMs are linear classifiers. However, SVMs are more effective because they maximize the margin or the distance between the decision boundary (the hyperplane) and the closest training point (support vector). In order to find the decision boundary \mathbf{w} that maximizes the margin, the following two conditions must be satisfied, given by Eqs. (8-9).

$$\sum_{i=1}^n w_i d_i^+ \geq +1, \quad (8)$$

$$\sum_{i=1}^n w_i d_i^- \leq -1, \quad (9)$$

where \mathbf{d}^+ and \mathbf{d}^- are n -dimensional feature vectors that belong to positive and negative datasets, \mathbf{D}^+ (real news) and \mathbf{D}^- (fake news), respectively. The above conditions must hold for all feature vectors in \mathbf{D}^+ and \mathbf{D}^- .

Let \vec{w} defines the margin around the decision boundary \mathbf{w} , the form of \vec{w} can be defined by Eq. (10) as follows.

$$\vec{w} = \sum_{i \in \mathbf{D}^+} \alpha_i \mathbf{d}_i^+ - \sum_{j \in \mathbf{D}^-} \alpha_j \mathbf{d}_j^-, \quad (10)$$

where α_i and α_j are weights that are assigned to feature vectors in positive and negative datasets \mathbf{D}^+ and \mathbf{D}^- , respectively. Since some news is more important than others, these weights are assigned to each feature vector in order to keep only vectors that matter and neglect vectors (news) that do not matter by setting their weights to 0. Note that $\sum_{i \in \mathbf{D}^+} \alpha_i \mathbf{d}_i^+$ and $\sum_{j \in \mathbf{D}^-} \alpha_j \mathbf{d}_j^-$ represent the centroids of real news and fake news, respectively. By forming a polytope around all positive (real)/negative (fake) news, any point \vec{h} can be expressed as a linear combination of the corner points Δ of that polytope given by Eq. (11).

$$\vec{h} = \sum_{d \in \Delta} \alpha_d \vec{d}, \quad (11)$$

where \vec{d} is a corner point. Assuming that the nearest points to \mathbf{w} (support vectors) are \mathbf{h}^+ and \mathbf{h}^- , the margin width must be equal to $\|\mathbf{h}^+ - \mathbf{h}^-\|$, and the boundary must be located halfway between \mathbf{h}^+ and \mathbf{h}^- on the line linking them. Since there might be several support vectors, we should search for the minimum distance between support vectors on both sides of the boundary, as given by Eq. (12).

$$\min \|\mathbf{h}^+ - \mathbf{h}^-\| = \min \|\mathbf{w}\|, \quad (12)$$

where $\|\cdot\|$ is the Euclidean norm.

3) LOGISTIC REGRESSION

LR is a classification algorithm used for predicting discrete and binary values like “real and fake” using the logistic function, also known as the sigmoid function. The logistic sigmoid function converts the output of LR into a probability value. The formula of the sigmoid function is given by Eq. (13).

$$h(x) = \frac{1}{1 + e^{-x}} \quad (13)$$

where e is a mathematical constant known as Euler’s number and \mathbf{x} is a feature vector.

In LR, the output value is between 0 and 1 because it predicts an event’s appearance probability. We still need the result in binary form; therefore, we use a threshold to convert the probability to binary form [41]. If $y = 0$ and 1 are the class label, then (0) is the negative or fake news, and (1) is a positive class or actual news. The LR equation is given by Eq. (14).

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (14)$$

where θ^T is weight factor which is the transpose of θ and $h_{\theta}(x)$ is estimated probability when the class label $y = 0$ or 1 , 0 for fake news and 1 for real news, the estimated probability can be calculated by Eq. (15).

$$P(y = 0 | x; \theta) + P(y = 1 | x; \theta) = 1 \quad (15)$$

To calculate the estimated probability for the fake news $y = 0$, use Eq. (16).

$$P(y = 0 | x; \theta) = 1 - P(y = 1 | x; \theta) \quad (16)$$

where P is the estimated probability, y is the class label, x is the feature vector, and θ is the weight factor.

4) RANDOM FORESTS CLASSIFIER

It is a supervised ML algorithm, which is used for classification and regression. This algorithm generates the output by creating DT from data samples, takes a prediction from each tree, and then selects the superior solution through voting.

All labeled data is first assigned to a root node (N), after which the feature (F) is found within a random subset of features and a threshold (T) is determined. After that, divide the labeled data assigned to (N) root node into two subsets, left and right, and assign (F , T) to each. If the left and right splitting data are too small to be split again, attach leaf nodes (child) (L) left and (L) right to (N). And, to split data, label the leaves with the most existing labels in the left and right directions (S_{left} and S_{right}). Alternatively, attach child leaf nodes (N_{left}) and (N_{right}) to (N) and assign (S_{left}) and (S_{right}) to then repeat for ($N = N_{left}$ and N_{right}).

The algorithm is randomly drawing repeated at each node for D -dimensional samples, typical subset size, for a random subset of features is given by Eq. (17).

$$round\sqrt{D} = round(\log_2 x) \quad (17)$$

where x is the features and $round$ means simplifying the number but remaining its value near to what it was.

RF uses a modified tree learning algorithm that selects a random subset of features at each candidate split in the learning process. This random selection of features is sometimes referred to as “feature bagging” [42].

5) STOCHASTIC GRADIENT DESCENT (SGD CLASSIFIER)

SGD classifier is used to fit the linear model. We calculate gradient costly based on the whole trained data, referred to as “batch gradient descent.” Using this algorithm will be very expensive if there is a large data set because it can be done for a single point in the trained data. So, updating the weights will be slow, and it will take time to converge the global minimum cost. To update the weight, we do the next step after training samples, and the cost minimum is reached one or more times [43].

Calculate the gradient of the feature first by determining the slope of the objective function with respect to each feature parameter. Second, choose a random initial value for the parameters and update the gradient function by putting the parameter value into the gradient function. Then, using the following formula, determine the step size is given by Eq. (18).

$$Step\ size = gradient \times learning\ rate \quad (18)$$

and finally, determine the new weight is given by Eq. (19).

$$new(w) = old(w) - step\ size \quad (19)$$

For our own datasets, let D_n is a set of the posts in the dataset. $D_n = \{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)\}$ where (n) is the total number of training examples (posts). The problem is to find the correct class for each post. First, the algorithm initializes with ($w = 0, b = 0$). Second for iteration (t) belongs to $[1, \dots, t]$, draw a random example (x_n, y_n) belongs to (D), compute the prediction is given by Eq. (20).

$$P_n = h(x_n) \quad (20)$$

And to compute the loss function is given by Eq. (21).

$$L_n = L(P_n, y_n) \quad (21)$$

where (x_n) is the features, and (y_n) is the class labels (real and fake labels), (w) is the weight, and (b) is the bias.

6) PASSIVE-AGGRESSIVE CLASSIFIER

It is an online learning algorithms family used for either classification or regression tasks [26]. It attempts to settle the issues of the perceptron rule. Let x_t be a set of posts $x_t = \{x_1, x_2, x_3 \dots, x_n\}$, where x_t is the instance presented to the algorithm on round (t) and let (y_t) is a set of labels $y_t = \{y_1, y_2\}$, where (y_t) belongs to $-1, 1$. Then, take each instance label pair (x_t, y_t) as an example.

This algorithm predicts using a classification function. So, it takes an array of sizes $[n_samples, n_features]$ as input, and the output is an array of sizes $[n_samples]$ because of its binary classification [44]. So, when applying PA in our data first, initialize $w = (0 \dots, 0)$. Then receive a post, a vector of words $d = (d_1, \dots, d_n)$ and d^T is the transpose of d . Predict positive (real) if, $d^T w > 0$, suppose that (y) is class label $y = +1$ or -1 so, $d^T w \geq +1$ if positive (real) $y = +1$ and $d^T w \leq -1$ if negative (fake) $y = -1$.

Note that taking both equations and multiplying it by (y), the answer will be the same, $y(d^T w) \geq 1$. To compute the loss function is given by Eq. (22).

$$L = \max(0, 1 - y(d^T w)) \quad (22)$$

And to calculate updated weight is given by Eq. (23).

$$w_{new} = w + yLd \quad (23)$$

where L is the loss function, (w) is the weight vector, d is a word vector, d^T is the transpose of d and (n) is the number of words. Many of these algorithms were combined using the voting classifier method to get the best ACC depending on trial and error.

7) VOTING CLASSIFIER

We generated a set of weak classifiers and then merged their output into a single final decision via the ensemble technique [45]. One of the ensemble approaches is the voting classifier utilized in this study. Multiple weak ML techniques are used in the same dataset to accomplish this. The Voting Classifier ($voting = 'hard'$) used in this study means that each model (from the previous classification models) votes for every instance in the data set. The last output prediction is

the one that receives more than half of the votes. The instance would be classified based on the majority class label; check the outcome tables to compare the models (Tables 4 - 12).

IV. EXPERIMENTAL RESULTS

This section is divided into four subsections: dataset description, hardware and software specifications, evaluation metrics, and discussion results. The first represents the three benchmark datasets: Fake-or-Real-News dataset, Media-Eval dataset, and ISOT dataset. The second represents the hardware and software specifications that we used in our experiments. The third is the evaluation metrics used to evaluate our work. The fourth subsection represents the results and the discussion. This subsection presents the classification results and ensemble-voting classifier on three different benchmark datasets.

We applied three different experiments to the three datasets. We then compared the results of six different ML algorithms and the proposed voting classifier. There are some tables and figures included to clear our point. We supported our idea with five different performance measures. The used performance measures are explained in detail in the evaluation metrics subsection. In the discussion subsection, we discuss the experimental results and provide them with a ROC curve.

A. DATASET DESCRIPTION

We tested our work on three datasets published online Table 3. The first is called the Fake-or-Real-News dataset. The second dataset is called the Media-Eval dataset. The third dataset is called ISOT fake news dataset.

Fake-or-Real-News dataset: contains three columns, which are title, text, and label. In our experiment, we only need the text and label columns. The total posts in this dataset are 6335 posts collected from social media. This dataset is divided into two types: fake with 3164 posts and real with 3171 posts. The dataset is available for free online [48].

Media-Eval dataset: this dataset contains fake and real posts from Twitter. It also contains images shared online, but we only need the posts in our work so, the images were ignored. This dataset is designed especially for information retrieval tasks and new technologies depending on social media. In this dataset, we worked on 15629 posts, 9404 for fake posts, and 6225 for actual posts [47].

ISOT dataset: this dataset contains two types of articles, fake and real news. The data was collected from accurate world sources in the period from 2016 to 2017. The real articles were collected from Reuters.com, and the fake articles were collected from unreliable websites. This dataset contains different topics. There are 23481 fake articles and 21417 for real articles, with 21417 total articles [24], [49], [50]. The main difference between the three datasets is the source of data collected. The real-or-fake news was collected from Facebook. The Media-Eval dataset was collected from Twitter. ISOT Dataset was collected from accurate world sources in the period from 2016 to 2017.

TABLE 3. Dataset description.

Dataset	Fake	Real	Total articles
1-Fake-or-Real-News dataset	3164	3171	6335
2- Media-Eval Dataset	9404	6225	15629
3-ISOT Dataset	23481	21417	44898

B. HARDWARE AND SOFTWARE SPECIFICATIONS

This work was implemented by using an anaconda 3.8/Jupyter notebook, python 3. We ran our experiments in a Core (TM) i5/2.6 GHz machine with 8 GB RAM. To implement the ML models, feature extraction, and feature selection algorithms, we used the scikit learn package. Natural Language Toolkit (NLTK) used for preprocessing. The matplotlib library was used to visualize the ROC curve. For the benchmark datasets, we used 10-fold cross-validation to split the data.

C. EVALUATION METRICS

We measured performance ACC, precision, recall, f1-score, and the ROC curve [46]. ACC is how we are relative to the right value. $ACC = \text{number of correct predictions} / \text{total number of predictions}$, ACC's equation is given by Eq. (24).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (24)$$

Precision is how close the measurements are, which is given by Eq. (25).

$$Precision = \frac{TP}{TP + FP} \quad (25)$$

Recall (Sensitivity) is how many correctly actual positive is defined, is given by Eq. (26).

$$Recall / Sensitivity / TPR = \frac{TP}{TP + FN} \quad (26)$$

The f1-score definition is that if the cost of false positives and false negatives varies, we require precision and recall, as given by Eq. (27).

$$F1-Score = 2 \times \frac{(Sensitivity \times Precision)}{(Sensitivity + Precision)} \quad (27)$$

ROC curve is known as (Receiver Operating Characteristics). It is an important performance measure that shows how much the model can distinguish between various classes, where the AUC is the area under the curve. Note that the high of this area means the better of the model to make a detection. The ROC curve computes using true positive rate (TPR) and false positive rate (FPR). Figures 2-9 show the applied system's ROC curve on the three data sets used in this research, as given by Eq. (28).

$$FPR = \frac{FP}{TN + FP} \quad (28)$$

where $TP = \text{True Positives}$, $TN = \text{True Negatives}$, $FP = \text{False Positives}$ and $FN = \text{False Negatives}$.

TABLE 4. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for Fake-or-Real-News dataset according to TF-IDF for feature extraction and CHI-SQUARE for feature selection using 10-fold cross-validation split way.

FAKE-OR-REAL-NEWS								
Methods	TF-IDF+ CHI-SQUARE							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	88.9	88.7	92	87	84	93	88	90
PA	93.8	93.8	93	95	94	94	94	94
LSVM	94.2	94.2	93	95	95	94	94	94
SGD	93.4	93.5	91	96	96	91	93	94
LR	91.9	92	92	87	84	93	88	90
RF	85.9	86.2	82	91	91	82	86	86
The proposed voting classifier	94.5	94.4	96	98	98	96	94	95

TABLE 5. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for Fake-or-Real-News dataset according to TF-IDF for feature extraction and ANOVA for feature selection using 10-fold cross-validation split way.

FAKE-OR-REAL-NEWS								
Methods	TF-IDF+ ANOVA							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	88.6	88.6	93	85	84	93	88	89
PA	93.7	93.7	93	94	94	93	94	94
LSVM	94.3	94.3	94	94	94	94	94	94
SGD	94	94	95	93	93	95	94	94
LR	91.8	91.8	93	85	84	93	88	89
RF	92.7	92.7	92	93	93	92	93	93
The proposed voting classifier	94.6	94.6	96	96	96	96	94	94

TABLE 6. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for Fake-or-Real-News dataset according to DOC2VEC for feature extraction selection using 10-fold cross-validation split way.

FAKE-OR-REAL-NEWS								
Methods	DOC2VEC							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	73.8	73.6	71	78	83	65	76	71
PA	84	83.9	79	91	93	75	86	82
LSVM	84	88.1	85	92	93	83	89	87
SGD	84	83.8	77	96	97	70	86	81
LR	88.6	88.5	86	93	93	84	89	88
RF	86.9	86.8	86	88	89	85	87	86
The proposed voting classifier	88.8	88.7	86	94	96	85	89	88

D. RESULTS AND DISCUSSION

This part will represent the results of the classification process using six different ML algorithms and an ensemble voting classifier. The data is split into two parts, training and testing, using the 10-fold cross-validation method. The evaluation metrics ACC, precision, recall, f1-score, and ROC-AUC are used to evaluate the results.

We presented the results of the three datasets, which are: Fake-or-Real-News dataset (the first dataset), Media-Eval dataset (the second dataset), and ISOT dataset (the third dataset) in this section. Tables 4-12 represent the results.

ML algorithms play a decisive role in fake news detection. With the help of an ensemble voting classifier and feature extraction and selection methods, our system should detect fake news from real news with high ACC. We used three datasets from various sources, including Facebook, Twitter, and other blogs/websites. On each dataset, we did three experiments and compared the results.

In the first experiment, we used the TF-IDF technique for feature extraction and the chi-square approach for feature selection. The number of retrieved features using the TF-IDF algorithm for the Fake-or-Real-News dataset was 18138. After removing the stop words, this method selects

TABLE 7. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for Media-Eval dataset according to TF-IDF for feature extraction and CHI-SQUARE for feature selection using 10-fold cross-validation split way.

Media-Eval								
Methods	TF-IDF+ CHI-SQUARE							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	89.1	87.8	88	92	95	80	91	86
PA	90.5	89.1	88	95	97	81	92	87
LSVM	90.7	89.5	89	94	96	83	92	88
SGD	90.5	89	88	95	97	81	92	87
LR	90.7	89.3	88	95	97	82	92	88
RF	90.4	89.5	90	91	94	85	92	88
The proposed voting classifier	91.2	90	89	99	99	83	93	89

TABLE 8. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for Media-Eval dataset according to TF-IDF for feature extraction and ANOVA for feature selection using 10-fold cross-validation split way.

Media-Eval								
Methods	TF-IDF+ ANOVA							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	90.7	88.7	89	93	97	81	93	87
PA	85.9	87.5	96	75	81	94	88	83
LSVM	91.9	89.9	90	95	98	82	94	88
SGD	90.4	87.8	88	96	98	78	93	86
LR	91.1	89	89	95	97	81	93	87
RF	90.9	89.8	92	90	94	85	93	88
The proposed voting classifier	92.3	91.6	93	96	98	89	94	90

TABLE 9. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for Media-Eval dataset according to DOC2VEC for feature extraction selection using 10-fold cross-validation split way.

Media-Eval								
Methods	DOC2VEC							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	56.5	60.8	77	46	41	80	54	59
PA	66.1	66.5	76	55	65	68	70	61
LSVM	66.1	60.7	68	59	83	39	75	47
SGD	66.1	61.4	68	63	86	37	76	46
LR	65.9	60.5	68	60	85	36	75	45
RF	69.5	65	71	66	85	45	77	53
The proposed voting classifier	68.9	66.2	73	64	96	54	76	57

only the unique words from the data. Table 2 lists some of the collected features and their frequencies using this approach. We implemented a feature selection method to make the classification algorithms in our task more understandable and the training phase move faster. Using the chi-square technique, a total of 10000 characteristics were selected. When the features are categorical, the chi-square algorithm is utilized. It is a statistic that analyses the relationship between two categorical variables. When compared to before using the methods, ACC improved by 0.8% after using feature reduction approaches.

Table 4 represents the results of the first dataset. The voting classifier system achieved ACC equals 94.5%, AUC equals 94.4%, precision equals 97.0%, recall equals 96.0%, and f1-score equals 95.0%. For LSVM, the ACC is 94.2%, the AUC is 94.2%, the precision is 95.0%, the recall is 94.0%, and the f1-score is 94.0%. Figure 3 represents the ROC curves for the current state of the arts and the proposed voting classifier. The ROC curve indicates the compromise between sensitivity (or TPR) and specificity (1 – FPR). Classifiers with curves closer to the upper left indicate better performance. Therefore, by noticing the curve, the voting classifier performs best.

TABLE 10. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for ISOT dataset according to TF-IDF for feature extraction and CHI-SQUARE for feature selection using 10-fold cross-validation split way.

ISOT								
Methods	TF-IDF+ CHI-SQUARE							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	94.5	94.5	95	94	95	94	95	94
PA	100	100	100	100	100	100	100	100
LSVM	100	100	100	100	100	100	100	100
SGD	99.5	99.5	100	99	99	100	100	100
LR	99.3	99.3	95	94	95	94	95	94
RF	100	100	100	100	100	100	100	100
The proposed voting classifier	100	100	100	100	100	100	100	100

TABLE 11. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for ISOT dataset according to TF-IDF for feature extraction and ANOVA for feature selection using 10-fold cross-validation split way.

ISOT								
Methods	TF-IDF+ ANOVA							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	94.5	94.5	95	94	95	94	95	94
PA	100	100	100	100	100	100	100	100
LSVM	100	100	100	100	100	100	100	100
SGD	99.5	99.5	100	99	99	100	100	99
LR	99.3	99.3	95	94	95	94	95	94
RF	100	100	100	100	100	100	100	100
The proposed voting classifier	100	100	100	100	100	100	100	100

TABLE 12. Comparing ACC, AUC, precision, recall, and f1-score using machine learning algorithms for ISOT dataset according to DOC2VEC for feature extraction selection using 10-fold cross-validation split way.

ISOT								
Methods	DOC2VEC							
	ACC (%)	AUC (%)	Precision (%)		Recall (%)		F1-score (%)	
			Fake	Real	Fake	Real	Fake	Real
NB	55.5	56.1	61	52	44	68	51	59
PA	55.7	54.6	56	55	74	36	64	43
LSVM	55.7	63.8	65	63	70	57	67	60
SGD	55.7	64.2	65	64	71	58	68	61
LR	64.4	64.1	65	64	70	58	67	61
RF	73.7	73.2	72	77	83	64	77	70
The proposed voting classifier	68.9	68.5	71	66	77	60	72	68

For the Media-Eval dataset, the extracted features using the TF-IDF algorithm were 2282, and the number of selected features was 1500 using the chi-square algorithm. The system achieved ACC equals 91.2%, AUC equals 90.0%, AUC equals 99.0%, recall equals 99.0%, and an f1-score equals 93.0% which appeared in Table 7. Figure 6 represents the ROC curves for the current state of the arts and the proposed voting classifier. The ROC curve of the voting classifier has a better performance.

For the ISOT dataset, The number of extracted features was 31095, and the number of selected features was

15000 features. The performance was 100% of ACC, AUC, precision, recall, and f1-score, as shown in Table 10.

In this experiment, we observed that the voting classifier technique has the highest ACC among all ML algorithms. Because the voting classifier technique combines two or more weak standard ML algorithms, each classifier votes for a class, and the class with the most votes wins.

The second experiment was applied the TF-IDF algorithm for feature extraction and the ANOVA algorithm for feature selection on the three datasets. For the Fake-or Real-News dataset, the number of extracted features using the

TF-IDF algorithm was 17378. The number of the selected features was 10000 features using the ANOVA algorithm. The results are shown in Table 5. The voting classifier achieved average ACC equals 94.6%, AUC equals 94.6%, precision equals 96.0%, recall equals 96.0%, and f1-score equals 95.0%. When compared to, for example, LSVM, which achieved ACC of 94.0%, AUC of 94.0%, precision of 94.0%, recall of 94.0%, and f1-score of 94.0%, the proposed voting classifier performed better. Figure 4 represents the ROC curve for the traditional ML algorithms and the proposed voting classifier. As shown from the curve, the proposed method has a better performance when compared with the current state of the arts.

For the Media-Eval dataset, The extracted features were 1852 by the TF-IDF algorithm and the selected features were 1000 features using ANOVA algorithms. The proposed voting classifier achieved average ACC equals 92.3%, AUC equals 91.6%, precision equals 93.0%, recall equals 98.0%, and f1-score equals 94.0%. Comparing with, for example, LSVM that achieved ACC equals 91.9%, AUC equals 89.9%, precision equals 95.0%, recall equals 82.0%, and f1-score equals 88.0%, the proposed voting classifier performed better. Table 8 represents the results of this dataset. From the results and Figure 7, the performance of the proposed voting classifier is better than the traditional ML algorithms.

For the ISOT dataset, the voting classifier system achieved 100% of ACC, AUC, precision, recall, and f1-score. Table 11 represents the results.

The third experiment was applying the word embedding techniques. We used the DOC2VEC algorithm for feature extraction and reduction. For the Fake-or Real-News dataset, the voting classifier achieved average ACC equals 88.8%, AUC equals 88.7%, precision equals 86%, recall equals 96%, and f1-score equals 89.0%. For example, LSVM, which achieved ACC equals 84.0%, AUC equals 88.1%, precision equals 85.0%, recall equals 93.0%, and f1-score equals 89.0%, which appeared in Table 6, the proposed voting classifier performed better.

For the Media-Eval dataset, the proposed voting classifier achieved average ACC equals 68.9%, AUC equals 66.2%, precision equals 73.0%, recall equals 96.0%, and f1-score equals 76.0%. Comparing to, for example, LSVM that achieved ACC equals 66.1%, AUC equals 60.7%, precision equals 68.0%, recall equals 83.0%, and f1-score equals 75.0%, as shown in Table 9. Figure 8 represents the ROC curves for classical ML techniques and the proposed voting classifier for this dataset.

For the ISOT dataset, the voting classifier achieved average ACC equals 68.9%, AUC equals 68.5%, precision equals 71.0%, recall equals 77.0%, and f1-score equals 72.0%. When compared to, for example LSVM that achieved ACC equals 55.7%, AUC equals 63.8%, precision equals 65.0%, recall equals 70.0%, and f1-score equals 67.0%, as shown in Table 12, the proposed voting classifier performed better.

Also, we noticed that the RF algorithm achieved better ACC than any algorithms for the ISOT dataset, as shown

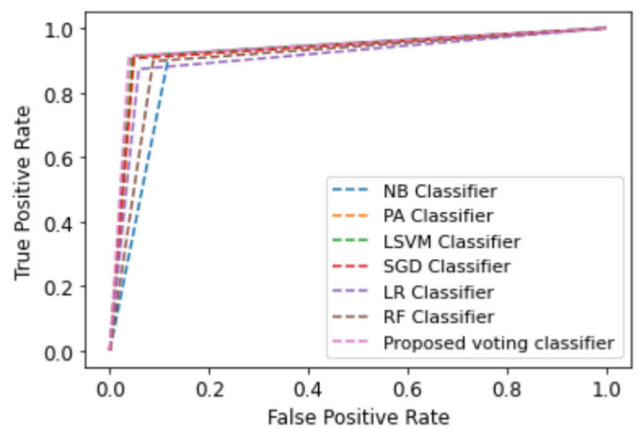


FIGURE 3. The ROC curve for the classification models applied on the first dataset depending on 10-fold cross-validation using TFIDF and chi-square algorithms.

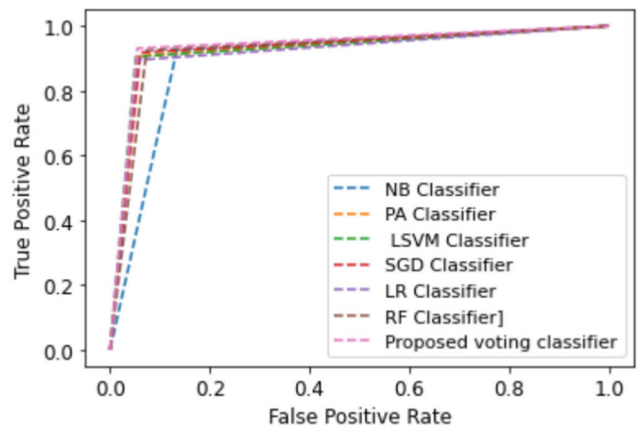


FIGURE 4. The ROC curve for the classification models applied on the first dataset depending on 10-fold cross-validation using TFIDF and ANOVA algorithms.

in Table 12. The reason could be that the RF algorithm won't overfit the model if there are enough trees in the forest, which causes better performance. Figure 9 represents the ROC curves for classical ML techniques and the voting classifier for this dataset.

After displaying the results and explaining each experiment separately, we notice that the first and the second experiments have higher ACC than the third experiment. With an ACC of 94.6%, the suggested voting classifier has the best classification performance of all ML algorithms, as shown in Table 5. The third experiment depends on Word embedding techniques which require massive amounts of data. Our data is small for this strategy to handle. After applying the TF-IDF technique for feature extraction, we enhanced the results by employing the feature reduction approach from the filter type.

We utilized a feature extraction technique in the first and second experiments, then a feature selection algorithm to select the important features from the text. The feature extraction algorithm was the same in both trials; the selection algorithm was different.

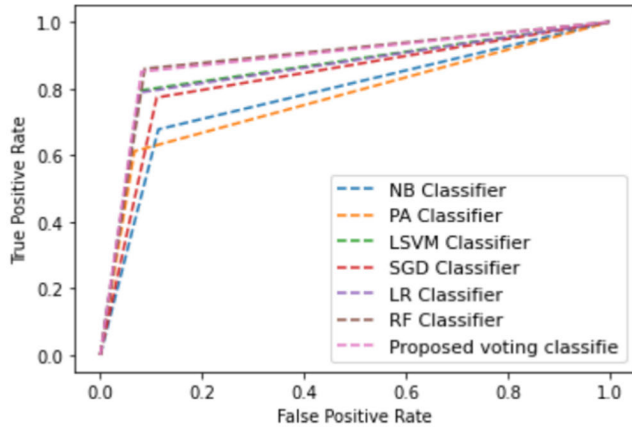


FIGURE 5. The ROC curve for the classification models applied on the first dataset depending on 10-fold cross-validation using DOC2VEC algorithm.

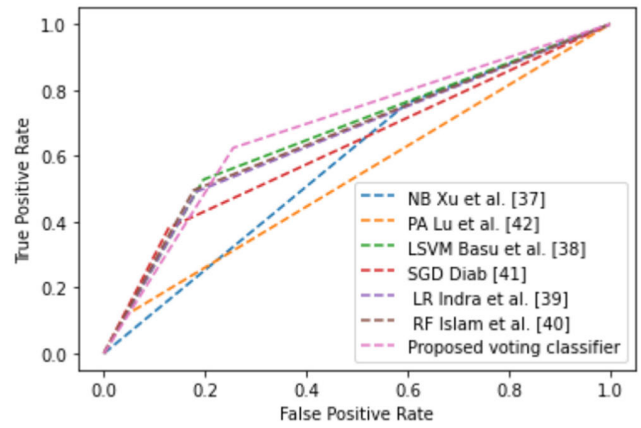


FIGURE 8. The ROC curve for the classification models applied on the second dataset depending on 10-fold cross-validation using DOC2VEC algorithm.

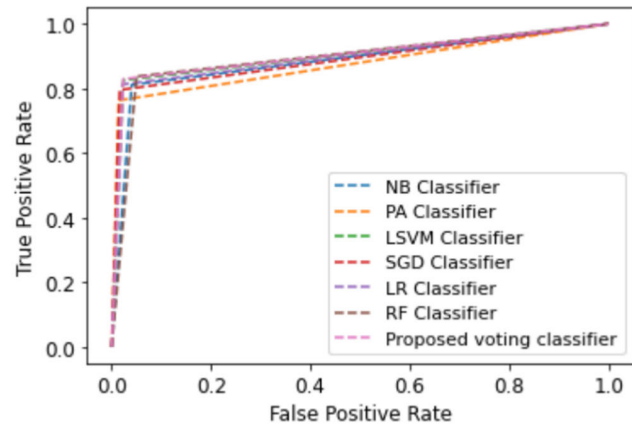


FIGURE 6. The ROC curve for the classification models applied on the second dataset depending on 10-fold cross-validation using TFIDF and chi-square algorithms.

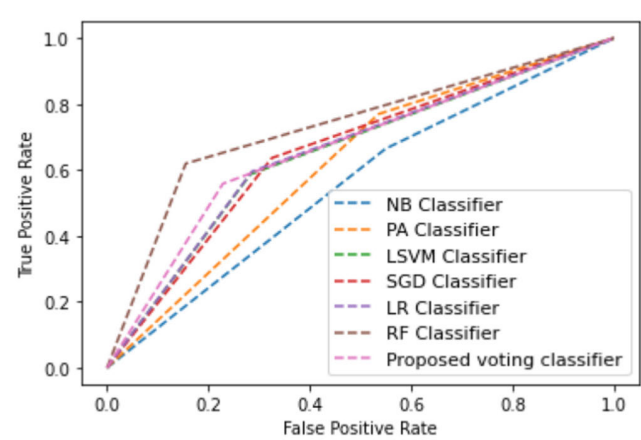


FIGURE 9. The ROC curve for the classification models applied on the third dataset depending on 10-fold cross-validation using DOC2VEC algorithm.

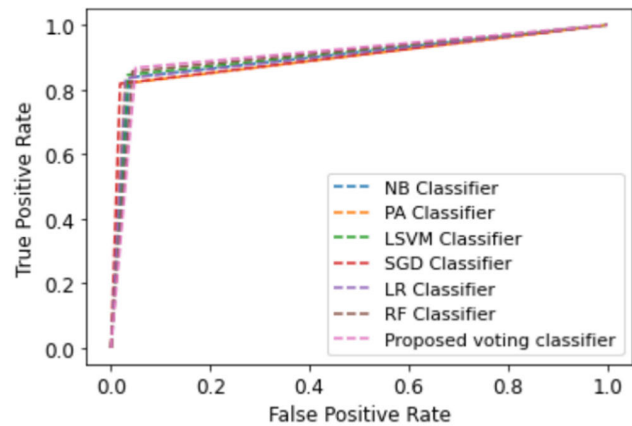


FIGURE 7. The ROC curve for the classification models applied on the second dataset depending on 10-fold cross-validation using TFIDF and ANOVA algorithms.

For the ISOT dataset, Table 13 shows a comparison between the suggested technique and previous work. As indicated in the table, the proposed strategy outperforms the other works in terms of ACC by 0.2% for the traditional ML algorithms [24]. This may be due to the usage of feature

TABLE 13. Performance comparison of the proposed method with the existing work for the ISOT dataset.

Study	ACC (%)
Ahmed et al. [3]	92.0
Hakak et al. [24]	99.8
Nasir et al. [26]	99.0
Jiang et al. [27]	99.94
Saleh et al. [28]	100
Proposed method	100

extraction and feature selection techniques. When comparing our work to [27] and [28], who used the DL techniques, our work outperforms [27] in terms of ACC and is equal to [28]. The reason behind using ML is that DL models require a large number of parameters to be trained and a large amount of data, which causes bad performance for the small datasets. The other two datasets, Fake-or-Real-News and Media-Eval, were used to demonstrate that the suggested voting classifier outperformed existing ML methods.

V. CONCLUSION

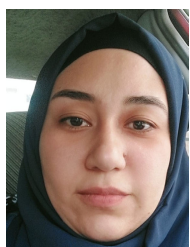
The fake news problem is not new, as disinformation has long been circulated in newspapers and radio. Because of the

internet, false news spreads quickly through social media and blogs. This type of information might be harmful. Thus we must be able to distinguish between fake and actual news. For better performance, we preprocessed the data. We used k-fold cross-validation to split the data. The TF-IDF and DOC2VEC algorithms were used to extract features, and the best features were chosen using the chi-square and ANOVA procedures. This study used six different ML algorithms: NB, LSVC, LR, RF, PA, and SGD. Three separate datasets are used in this experiment, all of which are freely available online. In three datasets, the suggested technique outperforms existing traditional methods in terms of ACC. The ACC was 94.5% for the Fake-or-Real-News dataset and 91.2% for the Media-Eval dataset, and 100% for the ISOT dataset. On the other hand, the presented study can only deal with textual data. So, in the future, datasets in images and videos with textual information will be collected and preserved from Facebook and other social media platforms. The annotated dataset can be used to detect fake photos and videos. In addition, for the proposed false news detection system, a real-time system for detecting fake news will be used on Facebook, Twitter, Instagram, and other platforms. The suggested method has the ability to bring benefits to a variety of new activities, including preventing the spread of fake news during elections, terrorism, natural disasters, and criminal activity for the good of humanity.

REFERENCES

- [1] A. M. Guess, B. Nyhan, and J. Reifler, "Exposure to untrustworthy websites in the 2016 U.S. election," *Nature Hum. Behav.*, vol. 4, no. 5, pp. 472–480, May 2020, doi: [10.1038/s41562-020-0833-x](https://doi.org/10.1038/s41562-020-0833-x).
- [2] *How Finland is Fighting Fake News in the Classroom*. Accessed: Aug. 21, 2021. [Online]. Available: <https://www.weforum.org/agenda/2019/05/how-finland-is-fighting-fake-news-in-the-classroom/>
- [3] H. Ahmed, I. Traore, and S. Saad, "Intelligent, secure, and dependable systems in distributed and cloud environments," in *Proc. 1st Int. Conf. Intell., Secur. Dependable Syst. Distrib. Cloud Environ.*, vol. 10618, 2017, pp. 169–181, doi: [10.1007/978-3-319-69155-8](https://doi.org/10.1007/978-3-319-69155-8).
- [4] D. Keskar, S. Palwe, and A. Gupta, *Fake News Classification on Twitter Using Flume, N-Gram Analysis, and Decision Tree Machine Learning Technique*. Singapore: Springer, 2020, pp. 139–147, doi: [10.1007/978-981-15-0790-8_15](https://doi.org/10.1007/978-981-15-0790-8_15).
- [5] A. Bali, M. Fernandes, S. Choubey, and M. Goel, "Comparative performance of machine learning algorithms for fake news detection," 2019, doi: [10.1007/978-981-13-9942-8_40](https://doi.org/10.1007/978-981-13-9942-8_40).
- [6] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big Data*, vol. 8, no. 3, pp. 171–188, Jun. 2020, doi: [10.1089/big.2020.0062](https://doi.org/10.1089/big.2020.0062).
- [7] K. Shu, S. Wang, and H. Liu, "Beyond news contents: The role of social context for fake news detection," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 312–320, doi: [10.1145/3289600.3290994](https://doi.org/10.1145/3289600.3290994).
- [8] J. T. Pintas, L. A. F. Fernandes, and A. C. B. Garcia, *Feature Selection Methods for Text Classification: A Systematic Literature Review*, no. 123456789. Cham, Switzerland: Springer, 2021.
- [9] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in *Proc. 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (ACL)*, vol. 2, 2017, pp. 422–426, doi: [10.18653/v1/P17-2067](https://doi.org/10.18653/v1/P17-2067).
- [10] S. Yang, K. Shu, S. Wang, R. Gu, F. Wu, and H. Liu, "Unsupervised fake news detection on social media: A generative approach," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5644–5651, doi: [10.1609/aaai.v33i01.33015644](https://doi.org/10.1609/aaai.v33i01.33015644).
- [11] K. Shu, D. Mahudeswaran, and H. Liu, "FakeNewsTracker: A tool for fake news collection, detection, and visualization," *Comput. Math. Org. Theory*, vol. 25, no. 1, pp. 60–71, Mar. 2019, doi: [10.1007/s10588-018-09280-3](https://doi.org/10.1007/s10588-018-09280-3).
- [12] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 637–645, doi: [10.1145/3159652.3159677](https://doi.org/10.1145/3159652.3159677).
- [13] J. Zhang, B. Dong, and P. S. Yu, "FAKEDETECTOR: Effective fake news detection with deep diffusive neural network," 2018, *arXiv:1805.08751*.
- [14] S. Gilda, "Notice of violation of IEEE publication principles: Evaluating machine learning algorithms for fake news detection," in *Proc. IEEE 15th Student Conf. Res. Develop. (SCORED)*, Dec. 2017, pp. 110–115, doi: [10.1109/SCORED.2017.8305411](https://doi.org/10.1109/SCORED.2017.8305411).
- [15] H. Ko, J. Y. Hong, S. Kim, L. Mesicek, and I. S. Na, "Human-machine interaction: A case study on fake news detection using a backtracking based on a cognitive system," *Cognit. Syst. Res.*, vol. 55, pp. 77–81, Jun. 2019, doi: [10.1016/j.cogsys.2018.12.018](https://doi.org/10.1016/j.cogsys.2018.12.018).
- [16] C.-S. Atodiresei, A. Tănăsescu, and A. Ifte, "Identifying fake news and fake users on Twitter," *Proc. Comput. Sci.*, vol. 126, pp. 451–461, Jan. 2018, doi: [10.1016/j.procs.2018.07.279](https://doi.org/10.1016/j.procs.2018.07.279).
- [17] J. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu, "Content based fake news detection using knowledge graphs," in *Proc. 17th Int. Semantic Web Conf.*, Monterey, CA, USA, Oct. 2018, doi: [10.1007/978-3-030-00671-6_39](https://doi.org/10.1007/978-3-030-00671-6_39).
- [18] M. Pothast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist. Proc. Conf. (ACL)*, vol. 1, 2018, pp. 231–240, doi: [10.18653/v1/p18-1022](https://doi.org/10.18653/v1/p18-1022).
- [19] M. Ghafari, S. Yakhchi, A. Beheshti, and M. Orgun, "Social context-aware trust prediction: Methods for identifying fake news," in *Web Information Systems Engineering—WISE 2018*. Cham, Switzerland: Springer, 2018, pp. 161–177. [Online]. Available: <https://researchers.mq.edu.au/en/publications/social-context-aware-trust-prediction-methods-for-identifying-fake-news>, doi: [10.1007/978-3-030-02922-7_11](https://doi.org/10.1007/978-3-030-02922-7_11).
- [20] F. Fernández-Reyes and S. Shinde, "Evaluating deep neural networks for automatic fake news detection in political domain," in *Proc. 16th Ibero-Amer. Conf. AI*, Trujillo, Peru, Nov. 2018, doi: [10.1007/978-3-030-03928-8_17](https://doi.org/10.1007/978-3-030-03928-8_17).
- [21] M. De Grandis, G. Pasi, and M. Viviani, "Multi-criteria decision making and supervised learning for fake news detection in microblogging," in *Proc. ROME, SIGIR Workshop Reducing Online Misinformation Exposure*, Paris, France, Jul. 2019, doi: [10.13140/RG.2.2.32071.04009](https://doi.org/10.13140/RG.2.2.32071.04009).
- [22] P. Ksieniewicz, M. Choraś, R. Kozik, and M. Woźniak, "Machine learning methods for fake news classification," in *Intelligent Data Engineering and Automated Learning (Lecture Notes in Computer Science)*, vol. 11872. Manchester, U.K.: Springer, 2019.
- [23] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, 2006.
- [24] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Gener. Comput. Syst.*, vol. 117, pp. 47–58, Apr. 2021, doi: [10.1016/j.future.2020.11.022](https://doi.org/10.1016/j.future.2020.11.022).
- [25] Y. Madani, M. Erritali, and B. Bouikhalene, "Using artificial intelligence techniques for detecting COVID-19 epidemic fake news in Moroccan Tweets," *Results Phys.*, vol. 25, Jun. 2021, Art. no. 104266, doi: [10.1016/j.rinp.2021.104266](https://doi.org/10.1016/j.rinp.2021.104266).
- [26] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *Int. J. Inf. Manage. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100007, doi: [10.1016/j.jjime.2020.100007](https://doi.org/10.1016/j.jjime.2020.100007).
- [27] T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, "A novel stacking approach for accurate detection of fake news," *IEEE Access*, vol. 9, pp. 22626–22639, 2021, doi: [10.1109/ACCESS.2021.3056079](https://doi.org/10.1109/ACCESS.2021.3056079).
- [28] H. Saleh, A. Alharbi, and S. H. Alsamhi, "OPCNN-FAKE: Optimized convolutional neural network for fake news detection," *IEEE Access*, vol. 9, pp. 129471–129489, 2021, doi: [10.1109/ACCESS.2021.3112806](https://doi.org/10.1109/ACCESS.2021.3112806).
- [29] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," *PLoS One*, vol. 15, no. 5, pp. 1–22, 2020, doi: [10.1371/journal.pone.0232525](https://doi.org/10.1371/journal.pone.0232525).
- [30] H. Woo, J. Kim, and W. Lee, "Validation of text data preprocessing using a neural network model," *Math. Problems Eng.*, vol. 2020, pp. 1–9, May 2020, doi: [10.1155/2020/1958149](https://doi.org/10.1155/2020/1958149).
- [31] G. Santafe, I. Inza, and J. A. Lozano, "Dealing with the evaluation of supervised classification algorithms," *Artif. Intell. Rev.*, vol. 44, no. 4, pp. 467–508, 2015, doi: [10.1007/s10462-015-9433-y](https://doi.org/10.1007/s10462-015-9433-y).
- [32] F. Y. Ahmed, Y. H. Ali, and S. M. Shamsuddin, "Using K-fold cross validation proposed models for spikeprop learning enhancements," *Int. J. Eng. Technol.*, vol. 7, nos. 4–11, pp. 145–151, 2018, doi: [10.14419/ijet.v7i4.11.20790](https://doi.org/10.14419/ijet.v7i4.11.20790).

- [33] D. Berrar, "Cross-validation," *Comput. Biol.*, vols. 1–3, pp. 542–545, Jan. 2018, doi: [10.1016/B978-0-12-809633-8.20349-X](https://doi.org/10.1016/B978-0-12-809633-8.20349-X).
- [34] Q. Liu, J. Wang, D. Zhang, Y. Yang, and N. Wang, "Text features extraction based on TF-IDF associating semantic," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018, pp. 2338–2343, doi: [10.1109/CompComm.2018.8780663](https://doi.org/10.1109/CompComm.2018.8780663).
- [35] R. Dzisevic and D. Sesok, "Text classification using different feature extraction approaches," in *Proc. Open Conf. Electr., Electron. Inf. Sci. (eStream)*, Apr. 2019, pp. 1–4, doi: [10.1109/eStream.2019.8732167](https://doi.org/10.1109/eStream.2019.8732167).
- [36] S. Li and B. Gong, "Word embedding and text classification based on deep learning methods," in *Proc. MATEC Web Conf.*, vol. 336, 2021, p. 06022, doi: [10.1051/mateconf/202133606022](https://doi.org/10.1051/mateconf/202133606022).
- [37] H. I. Celenli, S. T. Ozturk, G. Sahin, A. Gerek, and M. C. Ganiz, "Document embedding based supervised methods for Turkish text classification," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 477–482, doi: [10.1109/UBMK.2018.8566326](https://doi.org/10.1109/UBMK.2018.8566326).
- [38] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved chi-square for Arabic text classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 32, no. 2, pp. 225–231, Feb. 2020, doi: [10.1016/j.jksuci.2018.05.010](https://doi.org/10.1016/j.jksuci.2018.05.010).
- [39] S. Xu, Y. Li, and Z. Wang, "Bayesian multinomial naïve Bayes classifier to text classification," in *Advanced Multimedia and Ubiquitous Engineering (Lecture Notes in Electrical Engineering)* vol. 448, Singapore: Springer, May 2017, pp. 347–352, doi: [10.1007/978-981-10-5041-1_57](https://doi.org/10.1007/978-981-10-5041-1_57).
- [40] A. Basu, C. Walters, and M. Shepherd, "Support vector machines for text categorization," in *Proc. 36th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2003, pp. 1–8, doi: [10.1109/HICSS.2003.1174243](https://doi.org/10.1109/HICSS.2003.1174243).
- [41] S. T. Indra, L. Wikarsa, and R. Turang, "Using logistic regression method to classify tweets into the selected topics," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2016, pp. 385–390, doi: [10.1109/ICACSIS.2016.7872727](https://doi.org/10.1109/ICACSIS.2016.7872727).
- [42] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, "A semantics aware random forest for text classification," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1061–1070, doi: [10.1145/3357384.3357891](https://doi.org/10.1145/3357384.3357891).
- [43] S. Diab, "Optimizing stochastic gradient descent in text classification based on fine-tuning hyper-parameters approach?: A case study on automatic classification of global terrorist attacks," Feb. 2019, *arXiv:1902.06542*.
- [44] J. Lu, P. Zhao, and S. C. H. Hoi, "Online passive aggressive active learning and its applications," *J. Mach. Learn. Res.*, vol. 39, no. 2014, pp. 266–282, 2014.
- [45] L. Moreira-Matias, J. Mendes-Moreira, J. Gama, and P. Brazdil, "Text categorization using an ensemble classifier based on a mean co-association matrix," in *Machine Learning and Data Mining in Pattern Recognition (Lecture Notes in Computer Science)*, vol. 7376, Berlin, Germany: Springer, May 2014, pp. 525–539, doi: [10.1007/978-3-642-31537-4_41](https://doi.org/10.1007/978-3-642-31537-4_41).
- [46] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process.*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: [10.5121/ijdkp.2015.5201](https://doi.org/10.5121/ijdkp.2015.5201).
- [47] *Mediaeval Dataset*. Accessed: Aug. 22, 2021. [Online]. Available: <https://www.multimediaeval.org/datasets/>
- [48] *Fake-or-Real-News Dataset*. Accessed: Aug. 22, 2021. [Online]. Available: https://github.com/NavePnow/Google-BERT-on-fake_or_real-news-dataset/tree/master/data
- [49] *ISOT Fake News Dataset*. Accessed: Aug. 22, 2021. [Online]. Available: <https://www.uvic.ca/engineering/ece/isot/datasets/>
- [50] *Explanation for ISOT Fake News Dataset*. Accessed: Aug. 22, 2021. [Online]. Available: https://www.uvic.ca/engineering/ece/isot/assets/docs/ISOT_Fake_News_Dataset_ReadMe.pdf



EMAN ELSAED received the B.Sc. degree from the Department of Information Technology, Faculty of Computers and Information, Mansoura University, Egypt, in 2014. She is currently pursuing the M.Sc. degree with the Faculty of Computers and Information, Mansoura University. Her research interests include artificial intelligence, machine learning, and data mining.



OSAMA OUDA (Member, IEEE) received the bachelor's degree in computer science from the College of Computer and Information Sciences, Mansoura University, Mansoura, Egypt, in 2000, the master's degree in computer science from the College of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, in 2007, and the Ph.D. degree from the Graduate School of Advanced Integration Science, Chiba University, Chiba, Japan, in 2011. Since 2011, he has been an Assistant Professor with the Department of Computer Science, Mansoura University. From 2013 to 2014, he was a Postdoctoral Research Scientist with the iProBe Lab, Department of Computer Science and Engineering, Michigan State University, Lansing, USA. In 2017, he was appointed as an Associate Professor with the Department of Information Technology, Mansoura University. He is currently an Associate Professor with the College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia. His research interests include biometric template security, pattern recognition, and machine learning. He is serving as a reviewer for different journals and conferences.



MOHAMMED M. ELMOGHY (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Faculty of Engineering, Mansoura University, Mansoura, Egypt, and the Ph.D. degree from the Department of Informatics, MIN Faculty, University of Hamburg, Hamburg, Germany, in 2010. He worked as a Visiting Researcher with the Department of Bioengineering, University of Louisville, Louisville, KY, USA, from July 2016 to August 2019. He is currently a Professor and the Chair of the Department of Information Technology, Faculty of Computers and Information, Mansoura University. He has authored/coauthored over 220 research publications in peer-reviewed reputed journals, book chapters, and conference proceedings. He advised more than 30 master's and Ph.D. graduates. His current research interests include computer vision, medical image analysis, machine learning, pattern recognition, and biomedical engineering. He is also a Professional Member of the ACM Society. He has also served as a technical program committee member for many workshops and conferences. He has served as a reviewer for various international journals.



AHMED ATWAN received the B.Sc., M.Sc., and Ph.D. degrees in electronics and communications engineering from the Faculty of Engineering, Mansoura University, Egypt, in 1988, 1998, and 2004, respectively. From 2000 to 2004, he worked as a Lecturer with the Faculty of Technology, Al Bahah, Saudi Arabia. From 2005 to 2009, he worked as an Assistant Professor with the Faculty of Computers and Information Sciences (FCIS), Mansoura University. From 2010 to 2015,

he worked as an Associate Professor with FCIS, Mansoura University. Since 2016, he has been working as a Professor with FCIS, Mansoura University. Since 2019, he has also been working as a Professor with the Faculty of Computer and Information Technology, Northern Border University, Saudi Arabia. His research interests include pattern recognition, networking, signal processing, and image processing.

EMAN EL-DAYDAMONY received the B.S., M.S., and Ph.D. degrees in electrical communications from the Faculty of Engineering, Mansoura University, Egypt, in 1998, 2003, and 2008, respectively. She worked as a Teaching Assistant, from 1998 to 2002, a Teacher, from 2003 to 2007, and a Lecturer, from 2008 to 2010. From 2010 to 2017, she worked as an Assistant Professor at the Information Technology Department, Faculty of Computers and Information, Mansoura University, Egypt. Since 2018, she has worked as an Associate Professor at the Faculty of Computers and Information, Mansoura University. Her current research interests include computer vision, pattern recognition, medical image processing, biomarker discovery, and bioinformatics.

...