International Journal of Automation and Computing

17(2), April 2020, 210-221
DOI: 10.1007/s11633-019-1216-5

Text-mining-based Fake News Detection Using Ensemble Methods

Harita Reddy Namratha Raj Manali Gala Annappa Basava

Department of Computer Science and Engineering, National Institute of Technology Karnataka, Mangalore 575025, India

Abstract: Social media is a platform to express one's views and opinions freely and has made communication easier than it was before. This also opens up an opportunity for people to spread fake news intentionally. The ease of access to a variety of news sources on the web also brings the problem of people being exposed to fake news and possibly believing such news. This makes it important for us to detect and flag such content on social media. With the current rate of news generated on social media, it is difficult to differentiate between genuine news and hoaxes without knowing the source of the news. This paper discusses approaches to detection of fake news using only the features of the text of the news, without using any other related metadata. We observe that a combination of stylometric features and text-based word vector representations through ensemble methods can predict fake news with an accuracy of up to 95.49%.

Keywords: Fake news, social media, stylometric features, word vectors, ensemble methods.

1 Introduction

The ease of access to the world wide web (WWW) has made it possible for people in every corner of the world to get real-time global news. With the advent of social media, rapid dissemination of news is possible; content can be shared with friends or followers, and thus information diffusion takes place on social networks[1]. However, this ease of accessibility to social media also leads to the prevalence of fake news, which is written in such a way that people are misled into believing false information presented by the news[2]. Some of the sources for fake news include people/bots that deliberately manipulate information for political agendas, and gossip stories on entertainment-related websites[2]. In the 2016 US presidential elections, fake news was found to have attracted greater engagement from social media users than the news published by conventional news sources[3]. Bovet and Makse[4] observed that 25% of the 30 million tweets that contained links to news sources during the 5 months till the election date were either highly biased or untrue. False information has been found to have faster diffusion, especially for politics-related news, and it evokes feelings like disgust and fear, as reflected in the replies given by the readers[5]. There have been many instances where readers believe such news without verifying the authenticity of the news content from trustworthy sources[6, 7].

This paper focuses on the improvement of the state-of-art techniques to identify fake news on social media by using stylometric (linguistic) features and word vector representations of the textual content. Finally, the stylometric features and the various word vector features are combined by applying ensemble methods: bagging, boosting and voting. No information related to the users or the media content in the news articles has been used, which is an advantage of our method because it does not require any other metadata and protects user privacy by using only the features of the text.

2 State-of-the-art overview

Many techniques have been proposed to identify fake news, which include data mining and social network analysis methods. Shu et al.[2] classify fake news detection models into news content models and social context models. Conroy et al.[8] propose operational guidelines for designing a system for verification of news. The authors see promising results by providing an innovative hybrid approach that combines linguistic cues and machine learning, with network-based behavioral data.

Natural language processing techniques for the detection of fake news have been evaluated by Gilda[9]. Term frequency-inverse document frequency (TF-IDF)[10] of bi-grams and probabilistic context-free grammar (PCFG) detection was used with various models including stochastic gradient descent and gradient boosting. TF-IDF of bi-grams with stochastic gradient descent model identified fake news with an accuracy of 77.2%. However, only the vector based approach cannot be used to analyze specific features and train the classifiers as these are specific to the particular training dataset.

Ruchansky et al.[11] proposed a model with three mod-

Research Article
Manuscript received June 13, 2019; accepted December 11, 2019; published online February 18, 2020
Recommended by Associate Editor Matjaz Gams
© Institute of Automation, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2020

ules: capture, score, and integrate. The first module is based on the response of the users and text present in the piece of news; it uses a recurrent neural network (RNN) to capture the temporal pattern of user activity on a given article. The second module learns the characteristics of the source based on the behavior of users, and in the third module, the previous two modules are integrated to classify an article as fake or not. This work combines text, response and source user information. This model detects fake news from the Twitter dataset with an accuracy of 89.2% and from Weibo with an accuracy of 95.3%.

Buntain and Golbeck[12] used structural, content-based, user and temporal features to design a system to detect fake news in popular Twitter threads. The content-based features include polarity, subjectivity and disagreement. Their system′s applicability is limited to highly re-tweeted threads of Twitter conversations, and in real-life, most tweets are rarely re-tweeted. Their high performing model applied on the BuzzFeed dataset achieved an accuracy of 65.29%.

A combination of textual and user features was used by Krishnan and Chen[13]. User features included number of friends, number of followers, friends to follower ratio and whether the user has a verified URL or not. Textual features include tweet length, word count, number of question marks, number of exclamation marks, number of URLs, number of capital letters, number of hashtags, etc. With an accuracy of 80.68% for the Hurricane Sandy dataset, they obtain a high recall without compromising too much on precision. However, their selected features are only applicable to social networks that have a concept of friends, followers and user verification.

Jin et al.[14] made one of the significant attempts to use images for verification of news by using visual features like clarity and coherence score, and statistical features of images like count and image ratio. Using these image features, they achieved the highest verification accuracy of 83.6%. This accuracy was boosted by more than 7% compared with other approaches that use non-image features only.

Deep learning approaches, which have gained ground in the past few years, have also been used in fake news detection. Yang et al.[15], have used both text and image information to train a model named as the text and image information based convolutional neural network (TI-CNN). They have used sentiment and lexical diversity for text. For images, they observed that real news had more images of faces whereas fake news had more irrelevant images. In their model, they used two parallel CNNs to extract latent features from both textual and visual information and achieved the highest precision of 0.92 and recall of 0.9227. CNNs however, require a large dataset and using them to analyze both text and images tends to be computationally expensive.

If the source user′s attributes and network information are available, that information can be also used to give a better judgment of the reliability of the news. However, sometimes it is not possible to obtain all this meta data about user and user connections, especially due to user privacy concerns. Though writers of fake news try to frame the text in such a way that it appears genuine, fake news can be detected by observing some generic textual features. This work uses stylometric features of the text, i.e., the features based on the style of writing, as well as word vector representations of the text for classifying the news.

In Section 3, we explain our proposed methodology, which includes information about the dataset used, data pre-processing steps, feature extraction, feature selection and classification. The detailed results, along with discussions are presented in the results and discussions section, which also includes comparison with other research. Finally, we conclude our work and suggest some future directions.

3 Proposed methodology

3.1 Dataset used

Two datasets have been combined for evaluating the proposed methodology. The FakeNewsNet[16] dataset is a result of a data collection project for fake news research at the Arizona State University, with the labeling done on the basis of fact-checking websites like PolitiFact. The McIntire Dataset[17] has been hosted by George McIntire and contains a balanced collection of fake and real news. The dataset mainly includes political news ranging from left-wing to right-wing sources in relation with the 2016 US elections. Political news is deliberately manipulated to spread political propaganda, which is often done through social media bots, with this practice being prevalent during elections[2]. By observing the fake news samples in the dataset, it can be noticed that many political articles are often intended to portray candidates from a certain political wing in a negative light, which can help shape people′s minds for electoral gains.

3.2 Data preprocessing

After combining the two datasets, the final combined dataset has only 2 columns, one containing the text and the other containing the label. The training set contains 5405 news articles and the test set contains 1352 news articles. The news articles are mostly based on US politics. The training set has a balanced distribution with 2696 real news samples (49.9%) and 2709 fake news samples (50.1%), as depicted in Fig. 1.

3.3 Feature extraction

3.3.1 Stylometric features

Stylometry is the study of linguistic features of a piece of text, usually used to verify the authenticity or author-

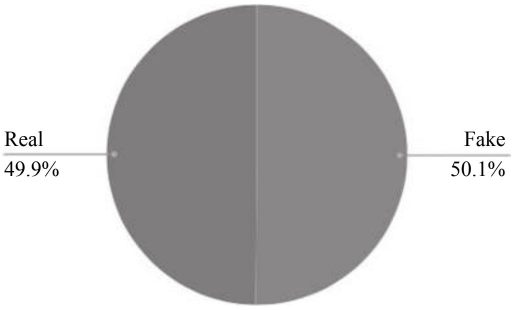


Fig. 1 Data distribution of real and fake news articles

ship of text based on the linguistic style of writing. Linguistic features can be useful because these features change while writing deceptive content to hide the author′s original writing style. Three feature sets were used as a reference for extracting stylometric features in the proposed methodology. Only the relevant features are extracted and used from the text-based dataset. The feature sets are:

Feature set 1: It is a minimal feature set used for authorship attribution in the work by Brennan and Greenstadt[18]. The features include number of unique words, complexity, Gunning-Fog index[19], character counts with and without whitespaces, average syllables per word, sentence count, average sentence length, and Flesch-Kincaid readability score. The gunning-fog index (GI) is a popularly used formula to determine the number of years of education required by an individual to understand a certain piece of text on the first reading itself. Flesch kincaid readability score (FS) is another readability test developed by Flesch and Kincaid to evaluate the difficulty in reading a piece of text. The higher the Flesch-Kincaid score, the easier it is to read the text.

$$GI = 0.4\left[\frac{\#words}{\#sentences} + 100\frac{\#difficultwords}{\#words}\right] \quad (1)$$

$$FS = 206.835 - 1.015\frac{\#words}{\#sentences} - 84.6\frac{\#syllables}{\#words}. \quad (2)$$

Feature set 2: This feature set is based on the lying detection dataset[20–22] and it includes features that were known to be quite effective in lie detection (Table 1).

We consider a word to be a big word if it has 6 or more characters, and a sentence to be a short one if it contains 10 or less words. Tentative words include words like "likely" and "probably" that express uncertainty.

Feature set 3: Zheng et al.[23] introduced the writeprints feature set for authorship attribution in short documents. The selection of features from writeprints is quite exhaustive and gives 73 attributes. The features selected from the writeprints set are listed in Table 2.

Function words are the words that contribute to the syntax of the sentence rather than its meaning. Greeting words include "hello", "good afternoon", "good evening", "good morning". *Yule′s K* measure[24] is a commonly used technique of evaluating the vocabulary difficulty of texts through the measurement of their lexical richness.

Table 1 Feature set 2 (Based on lying detection dataset)

Category	Features
Quantity	#syllables, #words, #sentences
Vocabulary	#big words, #syllables per word
Grammar	#short sentences, #long sentences, Flesh-Kincaid score, #words per sentence, sentence complexity, #conjunctions
Uncertainty	#certainty words, #tentative words, #modal verbs
Specificity	adjective and adverb rates, #affective terms
Verbal non-immediacy	self-references, #first, second and third person pronouns

Table 2 Feature set 3 (Based on writeprints features)

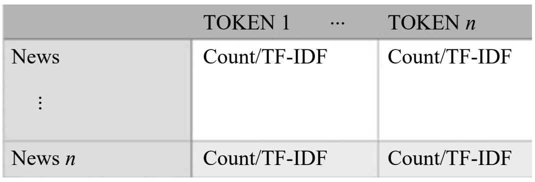
Category	Features
Character	#characters, % of digits, % of letters, % of uppercase letters, % of whitespace, letter frequencies, special character frequency
Word	words,% of short words (less than 4 chars), % of characters in words, avg. sentence length (chars), avg sentence length (words), #different words, once-occurring words′ frequency, twice-occurring words frequency, different length words frequency, Yule′s K measure
Synctactic	punctuation and function word frequency
Structural	#lines, #sentences, #of paragraphs, #sentences per paragraph, #characters per paragraph, #words per paragraph, Has a greeting, Has quoted content, Has URL
Content	Frequency of content specific keywords

3.3.2 Word vector features

The raw sequence of text data cannot be fed directly to the classifier. It has to be converted into vectors of numbers of a fixed size. Vector space models are often used to represent text documents in the form of vectors. Scikit-Learn[25] vectorization methods and also Gensim[26] Word2Vec, FastText (FT) models have been used to convert words to vectors.

Simple bag-of-words (BOW) count vector: Using scikit-learn package, the whole dataset is tokenized and the frequency of occurrence of each of these tokens in every document is calculated. The output would be a matrix M where every row represents a text document and every column represents a token (Fig. 2). $M[i,j]$ is the count of the token j in document i, i.e., the number of times token j occurs in document i. This is one of the simplest models for vectorization of documents.

BOW TF-IDF vector: Using scikit learn [25], the whole dataset is tokenized and the TF-IDF metric of each of these tokens in every document is calculated. It gives a matrix as the output (Fig. 2), where every row gives the



	TOKEN 1 ···	TOKEN n
News	Count/TF-IDF	Count/TF-IDF
⋮		
News n	Count/TF-IDF	Count/TF-IDF

Fig. 2 Count or TF-IDF matrix M

TF-IDF metric of every word present in that document (row), and if the word is not present then the value is taken as 0. TF-IDF of a token is calculated using the following two equations:

$$TF(t) = \frac{n_{td}}{n_d} \tag{3}$$

$$IDF(t) = \log\frac{D}{n_t} \tag{4}$$

where n_{td} is the number of times the token t appears in a document d, n_d is the total number of tokens in document d, D is the total number of documents and n_t is the number of documents with the token t in it. These 2 matrices suffer from the problem of sparsity with many rows containing 0.

Continuous bag of words (CBOW): It is a well known vector space model that generates dense vector representations for text. The CBOW architecture[27] predicts the target word from the context words given as input to a neural network. We can obtain the numerical vector form of each token in the text through this neural network architecture.

Skip-gram (SG): In the skip-gram architecture[27], the current word is given as an input to a neural network to predict the context words. CBOW and skip-gram are implemented using Word2Vec and FastText in the Gensim[26] toolkit. In Word2Vec[27], a word is encoded into a vector using a relation between words and its surrounding words using a neural net, whose hidden layer encodes the vector. Proposed as an extension to Word2Vec for proper representation of rare words, FastText[28] breaks down words into several n-grams (subwords) and combines the values of all these n-grams to give a single vector for a word.

After getting vectors for every token in the vocabulary, two methods are followed.

Method 1: In this method, the mean of the vector of a token is assigned to every token of a news piece and all the mean values are combined in the order the tokens are present in the text document to feed the classification model. The size of the input vector is equal to the size of a news piece which has the maximum number of tokens in it. If the size of a vector is shorter, it is padded with zeros at the end.

Method 2: A matrix is created of dimension $(no\_of\_data\_samples \times vocabulary\_size)$. The mean of the j-th word embedded vector is added in the cell ij if that word is present in the i-th Newspiece or "0" if not present (similar to the count-vector, TF-IDF matrix given above). In this method, the infrequently used words are pruned and the max-limit of vocabulary size is taken as 10 000. Every row, being a vector, is given as input to the classifiers. The dataset is pruned to reduce the runtime and to extract only the main excerpt from the documents.

3.4 Feature selection

Feature selection is useful for selecting only those features that have an impact on the determination of whether a piece of news is fake or not. Feature selection is applied both on stylometric and word-vector features to reduce the dimensions of the dataset.

3.4.1 Stylometric features

Recursive feature elimination has been used for the purpose of selecting the most important features from the stylometric feature sets. It removes the weakest or the features with least importance till the number of features in the dataset have reduced to a particular value.

3.4.2 Word-vector features

Few words were removed from the word vector space or vocabulary using word net lemmatizer and port stemmer. Both stemming and lemmatization are used for reduction of words to their root form, but stemming is a crude method which does not take into consideration the part of speech (POS) or the context of the words[29]. The Chi-square test was used for feature selection to reduce the time complexity issues that arise when dealing with large vector spaces. Using this method, the top 25 000 important words were selected. Also, the Lemmatizer, Stemmer and Chi-square tests were combined to reduce the word vocabulary to a greater extent to ensure a good performance.

After every dimensionality reduction step, count-vectors, TF-IDF vectors and other word vectors are obtained and classification is performed.

3.5 Classifiers used

We have used random forest (RF)[30], naive bayes (NB)[31] (Gaussian and multinomial), support vector machine (SVM)[32, 33], KNN, logistic regression (LR), bagging[34] with general bagging classifier and extra trees classifier[35], and boosting with adaboost[36] and stochastic gradient boosting[37] for the classification purpose.

3.6 Combining stylometric features and word vectors

After applying classification methods directly on the stylometric and word vector features separately, we work on the combination of the two types of features. We combine the features using ensemble methods: bagging, boosting and voting.

Bagging: In the bagging methodology, n random subsets (with replacement) are selected from the training set. A machine learning model is trained on each of these subsets. For a given test example, all the classifiers are used to make a prediction, and the final prediction is the mode of all the predictions.

In this work, stylometric and word vector features are combined. In each of the n random subsets, a training example comprises of both stylometric and word vector features. For all the training examples within the subset, the stylometric features are accumulated into the list *stylometric_subset* and the word vector features are accumulated into the list *word_vectors_subset*. The random forest classifier is trained on the *stylometric_sub set* and the logistic regression is trained on *word_vectors_subset*. These two trained classifiers are applied on the stylometric features and word vector features of all the examples in the testing set respectively. The two predictions from the random forest classifier and the logistic regression classifier for each testing example are stored. This procedure is repeated with all the training subsets.

We obtain $2n$ predictions for each testing example. The final prediction for a particular testing example is the mode of all the $2n$ predictions for the example.

Boosting: In this method, the vector of stylometric features is combined with the word vector representation of texts to give one vector for each sample. Then, the boosting algorithm is applied for training and testing on these vectors.

Voting: In this method, the vector of stylometric features is combined with the word vector representation of texts to give one vector for each sample. Three classifiers are trained on all the training samples and the final prediction for testing is simply the vote of the predictions from the three classifiers.

3.7 Metrics

We measure the performance of the classifier in case of both stylometric and word vector features using accuracy, precision, recall and F-score. The classifier is trained on the training set (5 405 articles) and tested on the test set (1 352 articles). We give the accuracy, precision, recall and F-score obtained by the classification model on the test set.

4 Results and discussions

4.1 Stylometric features

The results obtained on applying naive Bayes and random forest classifier on stylometric features (three feature sets) without performing feature selection are tabulated in Table 3.

Table 3 Results with RF and NB on the 3 feature sets

Feature set	Classifier	Acc. (%)	Prec.	Rec.	F1
Set 1	RF	59	0.63	0.51	0.57
	NB	62	0.59	0.96	0.73
Set 2	RF	78	0.79	0.75	0.77
	NB	59	0.67	0.34	0.45
Set 3	**RF**	**83**	**0.85**	**0.79**	**0.82**
	NB	69	0.71	0.63	0.67

The feature set 1 gives a very poor accuracy with both random forest and naive Bayes, indicating that the 9 features belonging to feature set 1 are not enough to detect the nature of the news. The writeprints-based feature set 3 has an exhaustive set of features and gives much better accuracy, especially with the random forest classifier. We thus continue further only with feature set 3. In further sections, references to stylometric features imply reference to feature set 3.

4.2 Analysis of feature set 3

All the features are not equally important in determining the authenticity of a piece of news. The extra trees classifier[35] is used to analyse the importance of the stylometric features. It fits a number of randomized decision trees on subsamples of the dataset. Then, averaging is used to improve the predictive accuracy and combat the problem of overfitting. The forest obtained from the classifier is used to get the importance of all the features in the dataset. In feature set 3 (Table 2), the ten features with highest importance values are "has quoted content", "has URL", "% of uppercase letters", "frequency of punctuation", "frequency of words of length 15", "% of whitespaces", "frequency of words of length 14", "average sentence length in words", "frequency of words of length 12" and "frequency of words of length 11". In Fig. 3, it is observed that real news has a very high average number of quotes compared to fake news. This might be because real news is substantiated with quotes, thus verifying its authenticity. On the other hand, fake news does not contain any evidence and hence lacks enough quotes.

In the case of uppercase letters (Fig. 4), the average percentage of uppercase characters in fake news is much higher than that in real news. This is because fake news is more dramatic with copious use of uppercase letters to make it a click-bait for the readers.

The most unimportant features which add no value to the dataset are frequency distribution of words with length of 21 characters, total number of lines and total numberof paragraphs. This is because the information regarding the lines and paragraphs is lost while collecting and integrating information from various websites.

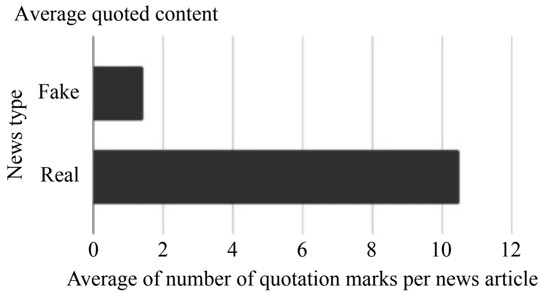


Fig. 3 Statistics for quoted content

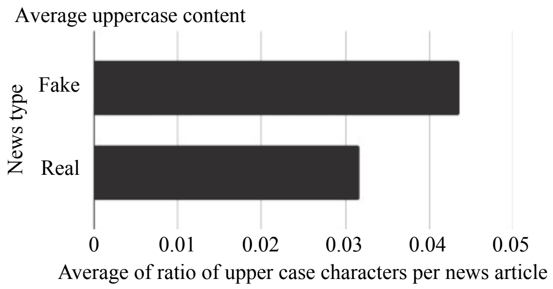


Fig. 4 Statistics for upper case letters

4.2.1 Feature selection on feature set 3

Recursive feature elimination is used for selecting the 50 most important features in the feature set 3 (Table 4). We observe that the maximum accuracy with the random forest classifier on feature set 3 is obtained on selecting 50 features.

Table 4 Accuracies for random forest obtained after feature selection using recursive feature elimination

#Selected Features	Acc. (%)	Prec.	Re.c	F1
35	82	0.82	0.80	0.81
40	82	0.84	0.78	0.81
45	83	0.84	0.80	0.82
50	**84**	**0.87**	**0.79**	**0.82**

4.2.2 Classification

After selecting only 50 most important features, classification methods are applied on the dataset. The results obtained with RF, NB, SVM, LR and KNN are shown in Table 5. It is observed that RF gives the best performance. SVM overfits the data. The accuracy obtained on the training is almost 100%, but on the test set it is poor. The results of SVM did not improve despite varying the regularization parameter. SVM might be overfitting the highly noisy training data. One more disadvantage of the SVM is its high time complexity to train on large datasets. The results obtained by using ensemble methods have been tabulated in Table 6. Please note that GBC = general bagging classifier, ETC = extra trees classifier and GB = gradient boosting classifier.

Ensemble methods are known to give better accuracy than their constituents and thus bagging and boosting have been used to improve the accuracy. Gradient boosting gives an accuracy of 86%.

Table 5 Results with basic classifiers

Classifier	Acc. (%)	Prec.	Rec.	F1
RF	**82.5**	**0.838**	**0.80**	**0.819**
NB-Mutinomial	67	0.64	0.76	0.69
NB-Gaussian	70	0.71	0.66	0.68
SVM	53	0.51	1.0	0.676
LR	75.7	0.716	0.84	0.77
KNN	67	0.68	0.62	0.65

Table 6 Bagging and boosting

Ensemble model	Acc. (%)	Prec.	Rec.	F1
GBC with KNN	69.6	0.726	0.614	0.665
GBC with RF	84.3	0.84	0.84	0.84
ETC	83	0.83	0.82	0.82
GB	**86**	**0.86**	**0.85**	**0.86**
AdaBoost	85	0.85	0.83	0.84

4.3 Word-vector features

The performance of different classifiers on count, TF-IDF, CBOW and skip-gram vectors is evaluated. Table 7 shows the performance of NB, RL and LR classifiers on the dataset without pruning the vocabulary. It can also be inferred that the precision, recall and F1 scores of both NB and LR classifiers are good. LR performs better than the other two classifiers on both count and TF-IDF vectors.

Tables 8–10 give the performance of the three classifiers on the vocabulary with different pruning and dimension reduction methods. For the results obtained in Table 8, the vocabulary dimension is reduced by using only the text pre-processing methods of lemmatization and stemming. LR performs better than the other two classifiers, especially with the TF-IDF vectors. In the case of Table 9, the chi-square test has also been used for dimensionality reduction along with stemming and lemmat-

Table 7 Results without any reduction in vocabulary dimension

Feature	Classifier	Acc. (%)	Prec.	Rec.	F1
	NB	87.50	0.90	0.83	0.87
BOW count	RF	80.32	0.85	0.73	0.79
	LR	92.08	0.92	.92	0.92
	NB	78.92	0.97	0.59	0.73
BOW TF-IDF	RF	80.55	0.84	0.74	0.79
	LR	89.49	0.88	0.91	0.895

Table 8 Results with reduction in vocabulary dimension using only lemmatization and stemming

Feature	Classifier	Acc. (%)	Prec.	Rec.	F1
BOW count	NB	82.25	0.88	0.74	0.80
	RF	72.85	0.88	0.52	0.65
	LR	83.73	0.88	0.77	0.824
BOW TF-IDF	NB	78.99	0.97	0.59	0.74
	RF	79.73	0.83	0.73	0.78
	LR	89.72	0.88	0.913	0.897

Table 9 Results with reduction in vocabulary dimension using lemmatization, stemming, chi-square test

Feature	Classifier	Acc. (%)	Prec.	Rec.	F1
BOW count	NB	84.17	0.87	0.795	0.83
	RF	74.85	0.85	0.598	0.70
	LR	82.91	0.82	0.83	0.83
BOW TF-IDF	NB	82.84	0.96	0.68	0.796
	RF	82.02	0.85	0.77	0.81
	LR	89.13	0.87	0.91	0.89

Table 10 Classifier results for mentioned features with reduction in vocabulary dimension using chi-square test

Feature	Classifier	Acc. (%)	Prec.	Rec.	F1
BOW count	NB	84.98	0.88	0.80	0.84
	RF	75.81	0.89	0.58	0.70
	LR	85.05	0.90	0.78	0.84
BOW TF-IDF	NB	82.91	0.96	0.68	0.79
	RF	81.88	0.85	0.77	0.81
	LR	89.28	0.88	0.91	0.89

ization. In Table 10, only the chi-square test is used for vocabulary dimension reduction. It is observed that LR performs better on TF-IDF vectors compared to count vectors after vocabulary dimension reduction. This difference in accuracy is profound in the case when all the three methods are used for dimensionality reduction (Table 9).

Figs. 5 and 6 are plots for count vectors and TF-IDF respectively that show the accuracy of NB and LR in the 4 cases: (1) without vocabulary dimension reduction, (2) dimension reduction using lemmatization and stemming, (3) dimension reduction using only the chi-square test for feature selection, and (4) dimension reduction using lemmatization, stemming and chi-square test. Observing the accuracy of NB and LR in the case of count vectors (Fig. 5), it is concluded that both the classifiers are performing well when the vocabulary is not pruned. Logistic regression is performing well on all the cases. Random forest′s performance is average but over-fitting is found in all the cases (not-pruned and pruned), hence it is not compared with LR and NB in the plots (Figs. 5 and 6).

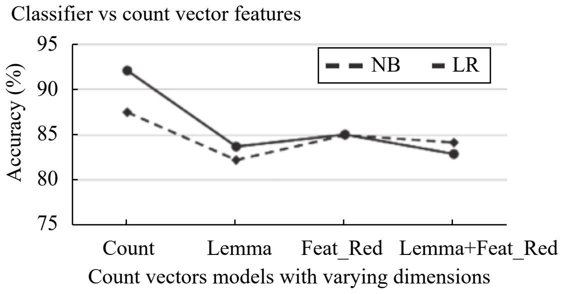


Fig. 5 Performance of classifiers on count vector features

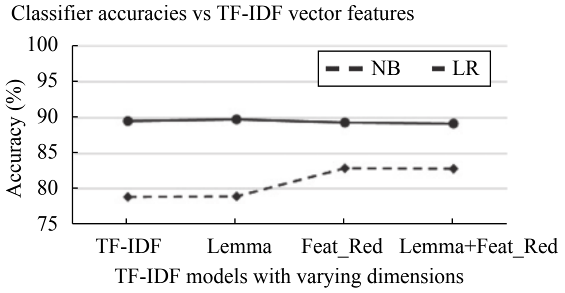


Fig. 6 Performance of classifiers on TF-IDF vector features

The TF-IDF curve (Fig. 6) implies that LR is performing better than NB in all the four cases and it is highest in the feature-selection and dimension reduction model. Performance of NB improves with the pruning of vocabulary. From Tables 7-10, it is clear that precision, recall and F1 scores of LR classifier are good.

The classifiers do not perform well on word vector features when method 1 structure for skip-gram and CBOW (as described in Section 3.3.2) is used as input to classifiers (Table 11).

Fig. 7 shows the performance of word embedding models Word2Vec and FastText. This plot is obtained using the results of Table 12, where method 2 (described in Section 3.3.2) is used to give input for the classifier. CBOW-W2V, CBOW-FT, SG-W2V and SG-FT embedding methods are found to be performing very well with LR compared to all other methods. Overall, the results obtained with skip-gram and CBOW are better than

Table 11 Classifier results for CBOW, skip-gram (poor performance with method 1 vector structure as input)

Feature	Classifier	Acc. (%)	Prec.	Rec.	F1
CBOW-W2V	NB	50.89	0.67	0.01	0.01
	RF	64.57	0.67	0.54	0.60
CBOW-FT	NB	50.89	0.67	0.01	0.01
	RF	64.94	0.67	0.56	0.61
SG-W2V	NB	50.89	0.67	0.01	0.01
	RF	71.23	0.74	0.65	0.69
SG-FT	NB	50.89	0.67	0.01	0.01
	RF	65.38	0.67	0.58	0.62

those obtained by using count and TF-IDF vectors. NB's performance is found to be average. Random forest was also used on the obtained word vector values but it has an average performance due to over-fitting on the training set.

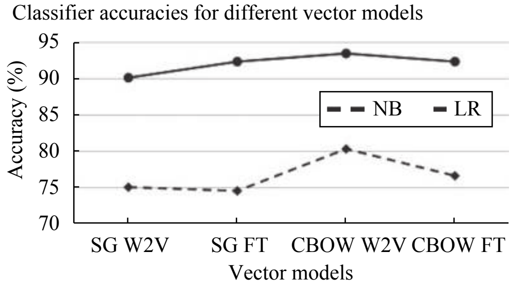


Fig. 7 Performance of classifiers on word to vector embedded features

Table 12 Results for CBOW, skip-gram features (Good performance: Method 2 vector structure as input)

Feature	Classifier	Acc. (%)	Prec.	Rec.	F1
CBOW-W2V	NB	80.32	0.75	0.90	0.82
	RF	86.39	0.90	0.81	0.85
	LR	93.42	0.94	0.93	0.93
CBOW-FT	NB	76.63	0.70	0.92	0.80
	RF	86.61	0.89	0.83	0.86
	LR	92.3	0.93	0.92	0.92
SG-W2V	NB	75.07	0.68	0.93	0.79
	RF	86.39	0.90	0.81	0.85
	LR	90.09	0.88	0.92	0.90
SG-FT	NB	74.56	0.67	0.93	0.78
	RF	86.17	0.88	0.83	0.86
	LR	92.3	0.92	0.93	0.92

The reduction of vocabulary dimension also did not solve this problem.

4.4 Ensemble methods to combine word-vectors and feature set 3 for classification

Both stylometric features (feature set 3) and the word vector features are used to get the combined prediction. The features are combined using ensemble methods: bagging, boosting and voting.

1) Bagging

In the bagging methodology, n random subsets (with replacement) are selected from the training set. m is the number of samples in each subset. m is taken as 700 for all cases. In our experiment, we take n subsets of the training set, each containing 700 samples (news pieces). For each subset, we train a RF classifier on the stylometric features of the samples in the subset and LR classifier on the word-vector representation of the samples. Hence, for n subsets, we obtain a total of n RF classifiers and n LR classifiers. These classifiers are then used to predict the label for the test set samples, with each test set sample getting $2n$ predictions (n RF classifiers applied to the stylometric features and n LR classifiers applied to the word vector representations of the test set samples). The mode of the predictions is taken as the final predicted label for each test sample. The results of this methodology of classification are tabulated in Table 13, with varying values of n. The details have been explained in Section 3.6.

Table 13 Results of bagging on both feature set 3 and word vectors

Word vectors	n	Acc. (%)	Prec.	Rec.	F1
SG (W2V)	10	86.76	0.87	0.86	0.86
	15	87.5	0.87	0.87	0.87
CBOW (FT)	10	90.01	0.91	0.88	0.89
	15	90.90	0.92	0.89	0.91
CBOW (W2V)	10	90.24	0.91	0.88	0.90
	15	**91.20**	**0.93**	**0.89**	**0.91**
SG (FT)	10	88.83	0.89	0.88	0.89
	15	90.09	0.90	0.90	0.90

Skip-gram and CBOW (both Word2Vec and FastText) have been considered for word vector representations to applying bagging. The highest accuracy of 91.20% has been observed in the case of CBOW (W2V) with good precision, recall and F-score.

2) Boosting

Gradient boosting (Grad) and AdaBoost (Ada) are used to combine the feature set 3 and word vector based features, as shown in Table 14. It is observed that the gradient boosting classifier works the best with CBOW(W2V) and AdaBoost works well with all vector models.

3) Voting

All the voting methods used are "hard" voting as soft voting results are average (Tables 15 and 16).

The ensemble of logistic regression, random forest and Adaboost is applied on the combination of writeprints, stylometric features and skip-gram (FastText) to obtain voting results. This ensemble performed the best when compared to an ensemble of other classifiers with respect to voting.

The maximum accuracy obtained by using bagging is 91.20% and voting on the combination of skip-gram (FT) and writeprints based features gives 91.94%, with precision, recall and F1-score above 90%. Boosting using gradient boosting algorithm on the combination of CBOW (Word2Vec) and stylometric features gives and accuracy of 95.49%, with a precision, recall and F-score of

Table 14 Using boosting on feature set 3 + word vectors

Model	Word vector model	Acc. (%)	Prec.	Rec.	F1
Grad	CBOW (FT)	94.82	0.95	0.95	0.95
	CBOW (W2V)	**95.49**	**0.95**	**0.95**	**0.95**
	SG (FT)	95.12	0.95	0.95	0.95
	SG (W2V)	95.12	0.95	0.95	0.95
Ada	CBOW (FT)	94.53	0.95	0.94	0.94
	CBOW (W2V)	94.53	0.95	0.94	0.94
	SG (FT)	94.53	0.95	0.94	0.94
	SG (W2V)	94.53	0.95	0.94	0.94

Table 15 Results for voting on feature set 3 + TF-IDF (post feature selection)

Classifiers	Weights	Acc. (%)	Prec.	Rec.	F1
NB+LR+Ada	1 1 1	80.84	0.77	0.88	0.82
	2 2 1	80.84	0.77	0.88	0.82
Bag+LR+Ada	1 1 1	84.02	0.82	0.87	0.84
	2 2 1	84.026	0.82	0.87	0.84
LR+RF+Ada	1 1 1	90.09	0.899	0.899	0.899
	2 2 1	90.38	0.89	0.92	0.903

Table 16 Results for voting on feature set 3 + wordvector features (WV)

Classifiers	WV	Acc. (%)	Prec.	Rec.	F1
NB+LR+RF	CBOW (W2V)	82.40	0.77	0.92	0.84
	CBOW (FT)	83.21	0.77	0.93	0.84
	SG (W2V)	78.03	0.72	0.92	0.80
	SG (FT)	80.70	0.74	0.94	0.83
NB+LR+Ada	CBOW(W2V)	84.62	0.78	0.96	0.86
	CBOW(FT)	85.28	0.79	0.96	0.86
	SG(W2V)	80.25	0.73	0.95	0.826
	SG (FT)	82.17	0.74	0.97	0.84
LR+RF+Ada	CBOW(W2V)	90.83	0.90	0.91	0.91
	CBOW(FT)	90.90	0.90	0.92	0.91
	SG(W2V)	91.20	0.90	0.92	0.91
	SG (FT)	**91.94**	**0.91**	**0.93**	**0.92**

Table 17 Comparison with other works (only text based)

Work	Method	Acc. (%)
Hybrid CNN and RNN[38]	Automatic Feature Identification Using LSTM and CNN	82%
TF-IDF bigrams[9]	Stochastic Gradient LSTM and CNN	77.2%
CNNs on Text[39]	Unimodal deception detection through text with CNNs	84%
Proposed Method	Boosting on Stylometric and Word Vector Features	95.49%

95%. On studying the fake news which are mis-labeled as real news in the test set, we note that those mis-classified samples have a greater average quoted content than usual. On manually observing some of those mis-classified texts, we note that in some places quotes have been used to emphasize certain words or phrases.

Table 17 gives the comparison of the proposed method with other text based methods. Hybrid CNN and RNN[38] identify fake news with an accuracy of 82% on a dataset based on tweets during five major events: Charlie Hebdo, Sydney Siege, Ottawa Shooting, Ferguson Shootings and Germanwings-Crash. However, deep learning models need large datasets. TF-IDF of bi-grams with stochastic gradient descent identifies pieces of fake news in the dataset published by Signal Media with an accuracy of 77.2%[9]. Gogate et al.[39] achieved an accuracy of 84% in unimodal deception detection using CNNs. In their work, they use audio, textual and visual cues but we compare only with the result obtained on text. In relation with the PolitiFact based news of FakeNewsNet, Shu et al.[16, 40] achieved an accuracy of 69% with Social Article Fusion. In another work that uses a combination of publisher-news relationships and user-news interactions, PolitiFact gives an accuracy of around 88%[41]. Paschalides et al.[42] developed a browser plugin using linguistic, stylistic, complexity and psychological features of news that gives an accuracy of 72% on the PolitiFact data.

A positive research outcome has been obtained in the proposed work. By using boosting methodology on a combination of word vectors and stylometric features, a precision of 95% and recall of 95% are obtained. This implies that 95% of the fake news is detected successfully (high recall) with only textual analysis. The advantage of the proposed methodology is that only textual features are used. Features related to user information may not be available in some cases and media-based features increase the time complexity of processing.

5 Conclusions and future work

The writeprints-based feature set is exhaustive for stylometric fake news detection but needs to be combined with word vectors to get better accuracy. The most important stylometric features for differentiating fake and real news include the amount of quoted content and uppercase letters. Ensemble methods including random forests, stochastic gradient descent and extra trees classifier worked the best on the stylometric features. In vectorized representation of text, TF-IDF vectors and skip-gram Word2Vec features are giving good results with a lower run time. Logistic regression is performing well for all types of word vector features, while the performance of the naive Bayes classifier is fluctuating. Individually, word vectors give much better accuracy compared to the accuracy obtained by applying classification models on stylometric features, thus underlining the importance of the implicit information present in the vectorized representation of text. Though word vector representations of text give better accuracies, we are able to achieve a good accuracy of 86% for stylometric features with gradient boosting classifier. We are also able to see that some stylometric features including the presence of quoted content and uppercase letters are significant for differentiating real and fake news, highlighting the importance of the style of writing news. We observe that random forest gives increasing accuracy as we go from feature set 1 to feature set 3. The writeprints feature set, which involves the combination of structural, syntax-based, lexical and content-specific features used for attributing authorship in short documents is the most exhaustive set of stylometric features[23]. These features have been used in the literature for authorship attribution of online content to achieve accuracies between 70% and 95%. These features work well in the case of fake news detection and they can capture the author's style of writing without knowing anything about the source of the news, thus helping in alleviating the problem of not using any metadata related to the news.

One important thing we notice in our work is the power of ensemble methods. The use of ensemble methods: bagging, boosting and voting have helped to improve the results significantly with both stylometric and word vector features. As we observe in the case of feature set 3 (in stylometric features), random forest gives an accuracy which is 6.8% more than the next best classifier, which is logistic regression. Gradient boosting and AdaBoost further outperform the random forest classifier. Even with a modestly sized training set, boosting is able to achieve a huge improvement because of its focus on news samples that are difficult to classify. This can be explained by the fact that boosting has helped to improve the predictive performance on several benchmark datasets spanning across different fields as it iteratively fits and gives more weight to samples that are misclassified[43, 44]. An accuracy of 95.49% is obtained on using boosting method on the combination on both stylometric and word vector features. This also highlights that using only textual features can give a good accuracy in the identification of fake news without the need of relying on user-related information, which is often private.

Our work is mainly based on the political domain and few other fields because of the minimal availability of curated and labelled datasets. Future research can focus on creation of more diverse datasets and their analysis with reduced timed complexity to make it ideal for real-time applications.

Online training could be used to incorporate real time data. Images could be taken into consideration along with the features considered here in order to improve overall results, preferably using pretrained models. Newer datasets covering more domains can be used for better generalization.

References

[1] B. Chang, T. Xu, Q. Liu, E. H. Chen. Study on information diffusion analysis in social networks and its applications. *International Journal of Automation and Computing*, vol. 15, no. 4, pp. 377–401, 2018. DOI: 10.1007/s11633-018-1124-0.

[2] K. Shu, A. Sliva, S. H. Wang, J. L. Tang, H. Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017. DOI: 10.1145/3137597.3137600.

[3] C. Silverman. Viral Fake Election News Stories Outperformed Real News on Facebook, [Online], Available: http://www.buzzfeed.com/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook?utm_term=.kq3Zz2Wxa#.rbBZBjgdx, December 15, 2018.

[4] A. Bovet, H. A. Makse. Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications*, vol. 10, no. 1, Article number 7, 2019. DOI: 10.1038/s41467-018-07761-2.

[5] S. Vosoughi, D. Roy, S. Aral. The spread of true and false news online. *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. DOI: 10.1126/science.aap9559.

[6] C. Silverman, J. Singer-Vine. Most Americans Who See Fake News Believe It, New Survey Says, [Online], Available: https://www.buzzfeednews.com/article/craigsilverman/fake-news-survey, December 15, 2018.

[7] C. Kang, A. Goldman. In Washington Pizzeria Attack, Fake News Brought Real Guns, [Online], Available: https://www.benton.org/headlines/washington-pizzeria-attack-fake-news-brought-real-guns, December 15, 2018.

[8] N. J. Conroy, V. L. Rubin, Y. M. Chen. Automatic deception detection: methods for finding fake news. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, American Society for Information Science, Silver Springs, St. Louis, USA, Article number 82, 2015.

[9] S. Gilda. Notice of violation of IEEE publication principles: Evaluating machine learning algorithms for fake news detection. In *Proceedings of the IEEE 15th Student Conference on Research and Development*, IEEE, Putrajaya, Malaysia, pp. 110–115, 2017. DOI: 10.1109/SCORED.2017.8305411.

[10] J. Ramos. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the 1st Instructional Conference on Machine Learning*, pp. 133–142, 2003.

[11] N. Ruchansky, S. Seo, Y. Liu. CSI: A hybrid deep model for fake news detection. In *Proceedings of ACM on Conference on Information and Knowledge Management*, ACM, Singapore, pp. 797–806, 2017. DOI: 10.1145/3132847.3132877.

[12] C. Buntain, J. Golbeck. Automatically identifying fake news in popular twitter threads. In *Proceedings of 2017 IEEE International Conference on Smart Cloud*, IEEE, New York, USA, pp. 208–215, 2017. DOI: 10.1109/SmartCloud.2017.40.

[13] S. Krishnan, M. Chen. Identifying tweets with fake news. In *Proceedings of 2018 IEEE International Conference on Information Reuse and Integration*, IEEE, Salt Lake City, USA, pp. 460–464, 2018. DOI: 10.1109/IRI.2018.00073.

[14] Z. W. Jin, J. Cao, Y. D. Zhang, J. S. Zhou, Q. Tian. Novel visual and statistical image features for microblogs news verification. *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 598–608, 2017. DOI: 10.1109/TMM.2016.2617078.

[15] Y. Yang, L. Zheng, J. W. Zhang, Q. C. Cui, Z. J. Li, P. S. Yu. TI-CNN: Convolutional Neural Networks for Fake News Detection, [Online], Available: https://arxiv.org/abs/1806.00749, August 1–20, 2018.

[16] K. Shu, D. Mahudeswaran, S. H. Wang, D. Lee, H. Liu. FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media, [Online], Available: https://arxiv.org/abs/1809.01286, December 15, 2018.

[17] G. McIntire. Fake and Real News Dataset, [Online], Available:https://github.com/GeorgeMcIntire/fake_real_news_dataset, July 10, 2018.

[18] M. Brennan, R. Greenstadt. Practical attacks against authorship recognition techniques. In *Proceedings of 21st Conference on Innovative Applications of Artificial Intelligence*, Association for the Advancement of Artificial Intelligence, Pasadena, USA, pp. 60–65, 2009.

[19] R. Gunning. The fog index after twenty years. *Journal of Business Communication*, vol. 6, no. 2, pp. 3–13, 1969. DOI: 10.1177/002194366900600202.

[20] J. K. Burgoon, J. P. Blair, T. T. Qin, J. F. Jr. Nunamaker. Detecting deception through linguistic analysis. In *Proceedings of the 1st NSF/NIJ Symposium on Intelligence and Security Informatics*, Springer, Tucson, USA, pp. 91–101, 2003. DOI: 10.1007/3-540-44853-5_7.

[21] S. Afroz, M. Brennan, R. Greenstadt. Detecting hoaxes, frauds, and deception in writing style online. In *Proceedings of IEEE Symposium on Security and Privacy*, IEEE, San Francisco, USA, pp. 461–475, 2012. DOI: 10.1109/SP.2012.34.

[22] J. T. Hancock, L. E. Curry, S. Goorha, M. Woodworth. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, vol. 45, no. 1, pp. 1–23, 2007. DOI: 10.1080/01638530701739181.

[23] R. Zheng, J. X. Li, H. Chen, Z. Huang. A framework for authorship identification of online messages: Writing - style features and classification techniques. *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 378–393, 2006. DOI: 10.1002/asi.20316.

[24] G. U. Yule. *The Statistical Study of Literary Vocabulary*, Cambridge, UK: Cambridge University Press, 2014.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] R. Řehůřek, P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 Workshop New Challenges for NLP Frameworks*, Valletta, Malta, pp. 46–50, 2010.

[27] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient estimation of Word Representations in Vector Space, [Online], Available: https://arxiv.org/abs/1301.3781, September 20, 2018.

[28] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. DOI: 10.1162/tacl_a_00051.

[29] A. G. Jivani. A comparative study of stemming algorithms. *International Journal of Computer Technology and Applications*, vol. 2, no. 6, pp. 1930–1938, 2011.

[30] L. Breiman. Random forests. *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.

[31] I. Rish. An empirical study of the naive Bayes classifier. In *Proceedings of IJCAI Workshop on Empirical Methods in Artificial Intelligence*, Seattle, USA: 2001.

[32] C. C. Chang, C. J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, Article number 27, 2011. DOI: 10.1145/1961189.1961199.

[33] M. Goudjil, M. Koudil, M. Bedda, N. Ghoggali. A novel active learning method using SVM for text classification. *International Journal of Automation and Computing*, vol. 15, no. 3, pp. 290–298, 2018. DOI: 10.1007/s11633-015-0912-z.

[34] L. Breiman. Bagging predictors. *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. DOI: 10.1023/A:1018054314350.

[35] P. Geurts, D. Ernst, L. Wehenkel. Extremely randomized trees. *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006. DOI: 10.1007/s10994-006-6226-1.

[36] Y. Freund, R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997. DOI: 10.1006/jcss.1997.1504.

[37] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002. DOI: 10.1016/S0167-9473(01)00065-2.

[38] O. Ajao, D. Bhowmik, S. Zargari. Fake news identification on twitter with hybrid CNN and RNN models. In *Proceedings of the 9th International Conference on Social Media and Society*, ACM, Copenhagen, Denmark, pp. 226–230, 2018. DOI: 10.1145/3217804.3217917.

[39] M. Gogate, A. Adeel, A. Hussain. Deep learning driven multimodal fusion for automated deception detection. In *Proceedings of IEEE Symposium Series on Computational Intelligence*, IEEE, Honolulu, USA, pp. 1–6, 2017. DOI: 10.1109/SSCI.2017.8285382.

[40] K. Shu, D. Mahudeswaran, H. Liu. FakeNewsTracker: A tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, vol. 25, no. 1, pp. 60–71, 2019. DOI: 10.1007/s10588-018-09280-3.

[41] K. Shu, S. H. Wang, H. Liu. Beyond news contents: The

role of social context for fake news detection. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, ACM, New York, USA, pp. 312–320, 2019. DOI: 10.1145/3289600.3290994.

[42] D. Paschalides, C. Christodoulou, R. Andreou, G. Pallis, M. D. Dikaiakos, A. Kornilakis, E. Markatos. Check-It: A plugin for detecting and reducing the spread of fake news and misinformation on the web. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE, Thessaloniki, Greece, pp. 298–302, 2019.

[43] G. Ridgeway. The state of boosting. *Computing Science and Statistics*, vol. 31, pp. 172–181, 1999.

[44] R. E. Schapire. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification*, D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, B. Yu, Eds., New York, USA: Springer, pp. 149–171, 2003. DOI: 10.1007/978-0-387-21579-2_9.

Harita Reddy received the B. Tech. degree in computer science and engineering from National Institute of Technology Karnataka, India in 2019. She is currently working as a software engineer at Uber, India.

Her research interests include data mining, machine learning and social network analysis.

E-mail: harita.nitk@gmail.com (Corresponding author)
ORCID iD: 0000-0002-3314-7880

Namratha Raj received the B. Tech. degree in computer science and engineering from National Institute of Technology Karnataka, India in 2019.

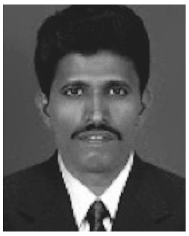
Her research interests include data science, machine learning, natural language processing and bioinformatics.

E-mail: namratha.mraj@gmail.com
ORCID iD: 0000-0002-2114-1553

Manali Gala received the B. Tech. degree in computer science and engineering from National Institute of Technology Karnataka, India in 2019. She is currently an analyst at Goldman Sachs, India.

Her research interests include machine learning and data analysis.

E-mail: manaligala7@gmail.com
ORCID iD: 0000-0002-5982-9062



Annappa Basava received the B. Eng. degree in computer science and engineering from the Govt. B.D.T. College of Engineering, Davangere affiliated to Mysore University, India in 1991, and received the M. Tech. and Ph. D. degrees in computer science and engineering from National Institute of Technology Karnataka, India in 2003 and 2012, respectively. Currently, he is a professor in the Department of Computer Science and Engineering, National Institute of Technology Karnataka, India. He has published more than 100 research papers in international conferences and journals. He has more than 20 years of experience in teaching and research. He was the Organizing Chair of *International Conference on Advanced Computing* 2013 and he is in the Technical Progam Committee of many international conferences and reviewer of journals. Currently, he is the Chair of India Council of the IEEE Computer Society and he was the Chair of IEEE Mangalore Subsection during 2018. He was the Secretary of IEI Mangaluru Local Centre. He is a Fellow of Institution of Engineers (India) and senior member of IEEE, ACM. Four research scholars completed their Ph. D. under his supervision and 7 scholars are currently enrolled for research under his supervision.

His research interests include cloud computing, big data analytics, distributed computing, software engineering and process mining.

E-mail: annappa@ieee.org
ORCID iD:0000-0002-4049-3677

IJAC International Journal of Automation & Computing

Cite this article as Reddy Harita, Raj Namratha, Gala Manali, Basava Annappa. Text–mining–based fake news detection using ensemble methods. *International Journal of Automation and Computing*. doi: 10.1007/s11633–019–1216–5

View online: https://doi.org/10.1007/s11633–019–1216–5

Articles may interest you

Mfsr: maximum feature score region-based captions locating in news video images. *International Journal of Automation and Computing*, vol.15, no.4, pp.454, 2018.
DOI: 10.1007/s11633-015-0943-5

A novel active learning method using svm for text classification. *International Journal of Automation and Computing*, vol.15, no.3, pp.290, 2018.
DOI: 10.1007/s11633-015-0912-z

Study on information diffusion analysis in social networks and its applications. *International Journal of Automation and Computing*, vol.15, no.4, pp.377, 2018.
DOI: 10.1007/s11633-018-1124-0

An integrated mci detection framework based on spectral-temporal analysis. *International Journal of Automation and Computing*, vol.16, no.6, pp.786, 2019.
DOI: 10.1007/s11633-019-1197-4

Step-based feature recognition system for b-spline surface features. *International Journal of Automation and Computing*, vol.15, no.4, pp.500, 2018.
DOI: 10.1007/s11633-018-1116-0

Dual-modal physiological feature fusion-based sleep recognition using cfs and rf algorithm. *International Journal of Automation and Computing*, vol.16, no.3, pp.286, 2019.
DOI: 10.1007/s11633-019-1171-1

An overview of contour detection approaches. *International Journal of Automation and Computing*, vol.15, no.6, pp.656, 2018.
DOI: 10.1007/s11633-018-1117-z

WeChat: IJAC

Twitter: IJAC_Journal