# Fake News Detection Using Sentiment Analysis

Bhavika Bhutani Neha Rastogi Priyanshu Sehgal Archana Purwar
*Department of Computer Science Engineering and Information Technology*
*Jaypee Institute of Information Technology*
Noida, India
bhavika.bhutani9@gmail.com rastogineha864@gmail.com sehgal.priyanshu3@gmail.com archana.purwar@jiit.ac.in

***Abstract*—Social media is one of the most revolutionary inventions of the present times. With its own set of advantages and disadvantages it is extremely essential for each one of us. Today Fake News has become a major problem wreaking havoc all over the world. Therefore building an algorithm with the best possible accuracy will be a revelation and it will have a massive impact on the social issues which are prevalent as well as on the current political scenario. Social Media and online news articles serve as a major source of news and for data for people since it can be approached easily , has a subsidized costing and is readily available-just a click away. However, it does have several negative impacts too such as no check on the source or authenticity and validity of the views being endorsed. Hence, we have proposed a new solution for fake news detection which incorporates sentiment as an important feature to improve the accuracy. It also investigates the performance of proposed method using three different data sets. Results show that proposed solution performs well. Moreover, the comparison is also made with other methods under this study.**

***Index Terms*—Fake News, Naive Bayes, Random Forest,Cosine similarity tf-idf, sentiment**

## I. Introduction

The information that we procure from the Internet and the Web cannot be relied on. Over the past few years spreading of rumors and fake information has reached a tipping point so much so that it has begun to affect social issues and political problems as well. The amount of time spent on social media platforms and online news sites has increased at an alarming rate. Thus most of the information that they have is acquired from these sources. Although social media can be accessed from anywhere and at any time and is also free but it provides anonymity while expressing out opinion therefore leading to a lack of accountability which greatly reduces the authenticity of data received from them as compared to a newspaper or any other trusted news source. Lack of constant supervision and an overseeing authority has allowed the wrongdoers to run amok and spread false information. Fake news is the deliberately falsified news which is sent out to fool people and make them believe in otherwise false information.

Misusing the news being broadcasted to various consumers can only result in confusion and chaos because various versions of the truth would be present. This disinformation may be spread in order to re-establish popularity or just for fun. In either case we need to figure out an effective way to identify fake and falsified information and prevent it from spreading further.

Hence this paper focuses on the detection of fake news accurately so that it helps to lessen the negative impact on individuals and society. The following sections of the paper give the relation work in that area, proposed solutions with experiments and results. Lastly, it concludes the paper with future directions.

## II. Related Work

Most challenging problem of deceptive language detection [4] is addressed by using automatic classification techniques. Other study [5] has used syntactic stylometry to deal with deceptive reviews. Hai et al. [6] has designed a semi supervised learning approach by using Laplacian regularized logistic regression) to improve the review spam detection performance. Fake news detection become more severe in case of reviews obtained from television interviews, posts on social networking system like Facebook and twitter. But, approaches based on statistics have helped in doing away with fake news but this has been a tough task especially due to the lack of labelled datasets. William Yang took care of this issue by releasing a new data set [1]. Moreover, different algorithms like Logistic Regression,Support Vector Machine(SVMs), Bi-directional Long Term Short Term Memory, Convolution Neural Network and hybrid CNN are used in the context of fake news. Recent study by Shu et al .[3] developed a false news collection methodology along with a deep learning solution in order to utilize metadata such as language based and replies, retweets etc. to further improve detection of fake news. Social article fusion model was adopted to classify fake news by them. [3]. However, fake news categorization may depend on whether the writer's attitude towards a particular topic like product, book, leader etc. is positive, negative, or neutral. For e.g., some opposition party made a statement on a political party, the sentiment of the statement is negative. It is an indication that the news is most probably fake. But, it is not 100 percent sure that a fake news is always negative but surely it plays an important part in determining whether the news is fake or not.

Therefore, this paper has proposed a solution to detect the fake news by incorporating a sentiment as a new column

978-1-7281-3591-5/19/$31.00 ©2019 IEEE

to dataset in order to detect fake news more precisely.

## III. Proposed Solution

This section discusses the proposed solution for fake news detection by combining Fake News with Sentiment Analysis. Proposed solution is shown in Fig. 1. It consists of various steps as below:

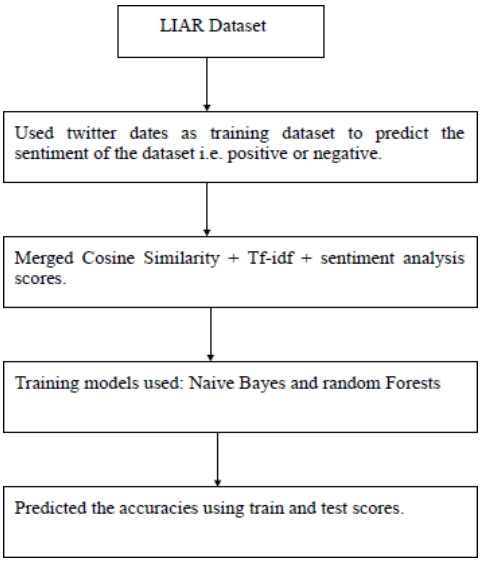


Fig. 1. Proposed Solution

**Step 1:** Merged data set was prepared using from different data sets namely Politifact, Kaggle and Emergent datasets.

**Step 2:** The different text preprocessing techniques like bigrams (series of two words taken from a given text) , trigrams (continuous series of three words taken from example text), CountVectorizer (count of terms in vector/ text , term frequency-inverse document frequency (tf-idf) vectorizer, tf-idf vectorizer with cosine similarity were used on the data set to choose best preprocessing technique as a result of Naive Bayes classifier. Results obtained from these techniques are shown in Table 1.

TABLE I
Comparison of text preprocessing techniques on Merged Dataset

| With Bi-grams | With Tri-grams | Count Vectorizer | Tf-Idf Vectorizer | Tf-Idf Vectorizer-Cosine Similarity |
|---|---|---|---|---|
| 0.676 | 0.676 | 0.683 | 0.801 | 0.816 |

**Step 3:** We have used tf-idf vectorizer on twitter dataset along with cosine similarity to build our vocabulary. Then Naive Bayes classifier was used to predict the sentiment of news statement of test data set (Merged data set) as shown in Fig. 2.

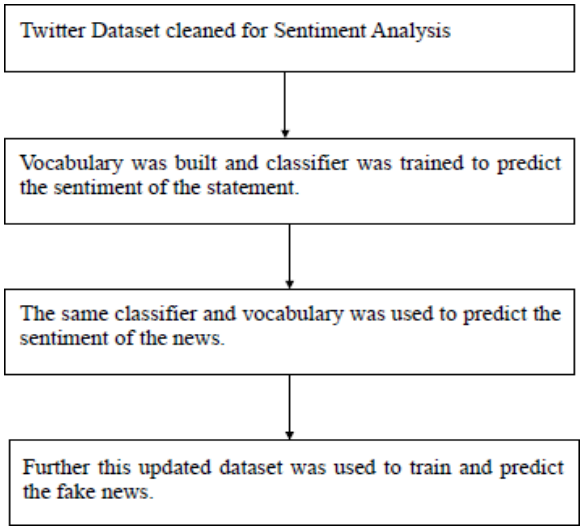


Fig. 2. Sentiment Prediction of news statement.

**Step 4:** We added additional columns: tf-idf scores, sentiments and Cosine similarity scores in Merged data set.

**Step 5:** Training model was built using Naive Bayes and Random Forest (train-test ratio: 3:1)

**Step 6:** Performance is evaluated and compared using accuracy.

Proposed solution consists of important steps 2 to 4 as preprocessing.It uses tf-idf Vectorizer with cosine similarities method for tokenizing a collection of text documents along with building a vocabulary of pre-existing words.Further we encoded the novel documents using that vocabulary.The encoded vector is returned with length of the entire vocabulary (bag of words) and an integer count for the number of times each word appeared had in the document. We have applied the CountVectorizer as well as tf-idf Vectorizer to text and title columns. We also applied the CountVectorizer with and without n-grams. The n-grams is a set of words occurring together within a given sub-text and when calculating the n-grams one typically moves ahead one word at a time. We ran the Naive Bayes algorithm with the Vectorizer on the Merged Dataset and highest accuracy was recorded using the CountVectorizer with one word at a time. After analysis, tf-idf was chosen as a best method to tokenize individual documents, to learn the vocabulary along with inverse document frequency weightings, allowing encoding the documents that we further encounter.

The main motive behind text preprocessing is to consider each document as a feature vector i.e. is to divide entire text into separate words (i.e. a bag of words or dictionary). TF-IDF Vectorizer [7] was used to convert the words into tokens accompanied by their frequencies in the document. Further, Cosine Similarity score between the statement $s_i$ and $s_j$ using (1) was used to show the degree of similarity between two vectors of a class ultimately providing us with conversion of text into vectors.

The sentiment that we obtain as a result from the prediction will be saved as a column and added to our dataset. The addition of this feature serves an essential role for us because it serves to add another feature to our existing dataset which is derived from our existing entries. We hypothesize that the sentiment that is put forth on writing a news article could serve as a pivotal deciding factor in the process of characterizing the news into fake or real. For example when a person is posing allegations against someone else or giving a false statement about someone else the sentiment behind it is generally negative thus helping in classifying it as Fake. Hence it will serve to further increase the accuracy and understand the emotion and sentiment associated with it.

$$(s_i, s_j) = \frac{\sum_{k=1}^{c} w_{ik} w_{jk}}{|s_i||s_j|} \quad (1)$$

Where ,

$s_i$ and $s_j$ are the corresponding weighted term vectors,
$s_i$ and $s_j$ are the length of statement vectors $s_i$ and $s_j$,
c is the count of terms in statements,
and $w_{ik}$ of $i^{th}$ statement is defined as :

$$w_{ik} = tf_{ik} \times log\frac{N}{n} \quad (2)$$

Where,

$w_{ik}$ = Weight of term $t_k$ in statement $s_i$
$tf_{ik}$ = Frequency of term $t_k$ in statement $s_i$
N = Number of statements in the news dataset
n = Number of statements where term $t_k$ occurs

On applying the Naive Bayes Algorithm on the dataset after it has been vectorized using TF-IDF Vectorizer and calculating cosine similarity score comes to the highest accuracy as 0.816 as compared to other text preprocessing ones.

Later, these tf-idf scores, sentiments as well as cosine similarity columns were added in merged data set and model was trained using Naive Bayes as well as Random Forest classifier. Finally the performance of proposed was evaluated and compared using different data set.

## IV. Experimental Settings

This section shows the experimental work and data sets used to evaluate the proposed solution.

### A. Datasets-The Potpourri

We have made experiments using three data sets. The first dataset which we will call the Merged Dataset has a combination of 3 separate ones namely Kaggle[1], PolitiFact[2] and the Emergent datasets.These datasets were combined by taking a median of the present data rows which resulted in an increase in overall accuracy by 0.5% . There were certain columns which were not common in all three datasets has been deleted before combining them. Among the rest of the columns we combined page-order from Emergent and domain-rank from Kaggle to form the page-order column which basically specifies what the ranking of that particular website is from among those mentioned in the dataset. Since PolitiFact did not have any such column we assigned a mean value to all those entries. Similarly we combined page-shares from PolitiFact and shares from Kaggle to form the page-shares column which basically specifies the number of shares for that particular article. Finally Merged dataset is described by 14 columns namely Title, Text, author, Language, country, page-order , domain rank , replies , replies count , participation count, likes, comments , page-shares and spam score. [1] [2] [3] [4]

Second dataset is the George McIntire dataset[3]. This dataset includes fake news and real news in 1:1 ratio. It has 10,558 rows and described by three columns namely title, text and category of news (fake /real). Third dataset is the LIAR dataset[4]. It includes 13 column as well as 12.8K human labeled short statements from APIs. Its category attribute has 6 six labels namely true, false, barely true, pantsfire, mostly true and half true. These class labels have been distributed quite evenly. With 1050 cases of pants-fire all other class labels were from 2,063 to 2,638.

### B. Preprocessing

We have pre-processed the data set by eliminating main_img_url and thread_title that have no impact on their performance. Moreover, data set was consisting of missing values. Missing values were replaced using mean method or deletion method [8] .Text column is one of the important features based on which all classification is done. Hence rows having null value for this column were deleted from the data set. We have carried out the experiments using python language .

## V. Results and Analysis

The performance of proposed solution has been evaluated on three different data sets using accuracy as an evaluation metric.

[1]https://www.kaggle.com/mrisdal/fake-news
[2]https://www.kaggle.com/arminehn/rumor-citation/version/3
[3]https://github.com/GeorgeMcIntire/fake_real_news_dataset
[4]https://www.cs.ucsb.edu/william/ data/liar_dataset.zip

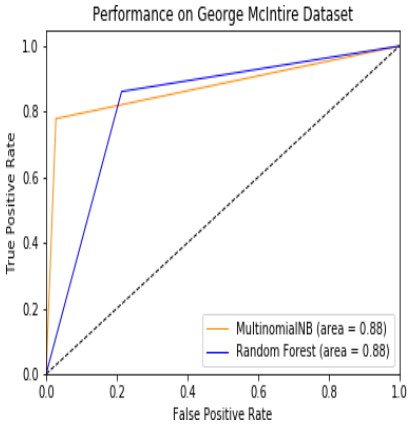


Fig. 3. ROC Curve for George McIntire dataset

Graph shown in Fig. 3 shows the ROC curves obtained on applying Random Forest and Naive Bayes algorithm respectively. Both of them have an AUC of 0.88 each. Here, True Positive Rate represents correctly classified news while False Positive Rate represents incorrectly positive classified news.

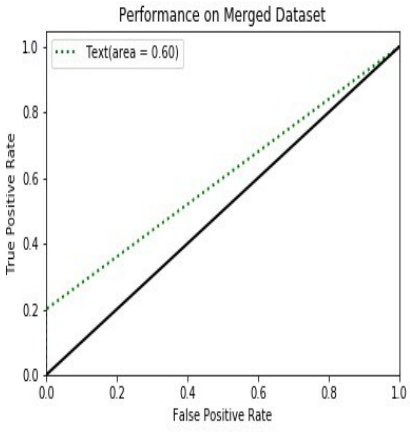


Fig. 4. ROC Curve for Merged dataset

Graph shown in Fig. 4 shows the ROC curves obtained on applying Random Forest and Naive Bayes algorithms respectively on the Merged dataset . We obtain an AUC of 0.60. The True Positive Rate and False Positive Rate represent the same news as mentioned above. The dotted black line aligned at 45 degree angle represents the base line for ROC Curves.

Graph shown in Fig. 5 shows the ROC curves obtained on applying Multinomial Naive Bayes algorithm on the LIAR dataset . We observe several curves with varying AUC due to different combinations of columns being used. We obtain an AUC of 0.60 for Text+Content which is the maximum value for this algorithm on LIAR dataset .

Graph shown in Fig. 6 shows the ROC curves obtained on applying Random Forest algorithm on the LIAR dataset

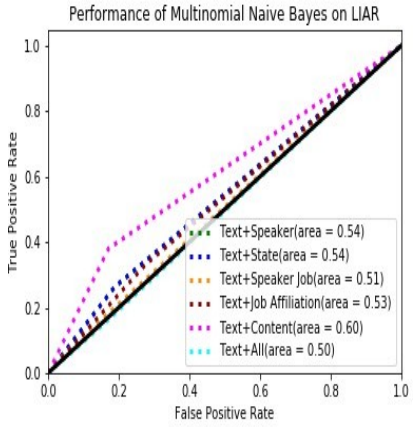


Fig. 5. ROC Curve for Multinomial Naive Bayes on LIAR dataset

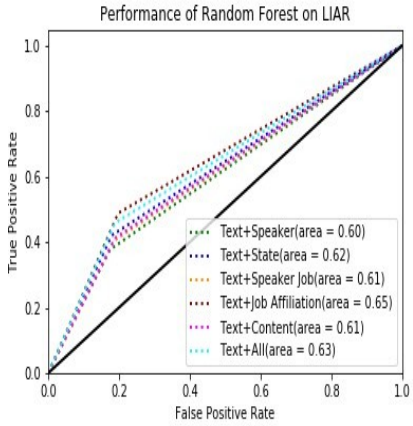


Fig. 6. ROC Curve for Random Forest on LIAR dataset

. Here again, we observe 6 different curves with varying AUC showing the different combinations of columns that we used. We obtain a peak in AUC 0.65 for Text+Affiliation which is the maximum value on the LIAR dataset .

Experiments were carried out using validation as well as test data set to evaluate the performance of proposed solution. Results cited in Table 2 shows better accuracy using Random Forest as classifier as compared to those given in Table 3 on LIAR[4] dataset. Further experiments are performed on Merged data set as well as George McIntire datasets as well. Table 4 shows the results on all three datasets as a result of Naive Bayes classifier. It gives accuracies by considering tf-idf with and without cosine similarity. Hence use of tf-idf with cosine similarity and sentiment analysis improves prediction result. Table 5 depicts accuracies as a result of Random Forest classifier in place of Naive Bayes. Thereby, proposed solution gives better results on three different data sets taken under study with Random Forest classifier.

TABLE II
ACCURACIES OBTAINED ON LIAR DATASET

| TYPE | Random Forests | | M-Naive Bayes | |
|---|---|---|---|---|
| | VALID | TEST | VALID | TEST |
| Text+Speaker | 0.995 | 0.351 | 0.343 | 0.236 |
| Text+State | 0.995 | 0.357 | 0.345 | 0.251 |
| Text+Job | 0.995 | 0.349 | 0.301 | 0.259 |
| Text+Context | 0.994 | 0.370 | 0.276 | 0.224 |
| Text+All | 0.995 | 0.356 | 0.212 | 0.185 |

TABLE III
ACCURACIES OBTAINED ON LIAR DATASET

| Hybrid CNNs | VALID | TEST |
|---|---|---|
| Text+Subject | 0.263 | 0.235 |
| Text+Speaker | 0.277 | 0.248 |
| Text+Job | 0.270 | 0.258 |
| Text+State | 0.246 | 0.256 |
| Text+Party | 0.259 | 0.248 |
| Text+Context | 0.251 | 0.243 |
| Text+History | 0.246 | 0.241 |
| Text+All | 0.247 | 0.274 |

## VI. CONCLUSION AND FUTURE WORK

With the mushrooming vogue of social media, more and more people are continuously consuming news from social media rather than the traditional media. However, the growing concern for us today is the proliferation of fake news, which has strong negative impacts on individual users and the society as a whole. Therefore this paper analyzes different text preprocessing techniques and selects tf-idf with similarity score as the best approach using accuracy as an evaluation metric. Further it enriches the merged data set using sentiment to increase the accuracy of fake news detection. The proposed approach is evaluated using three data sets and found better as compared to the approach without using tf-idf and cosine similarity as text preprocessing technique.

Future work will focus to apply different neural network algorithms as classifier in our proposed approach to enhance the accuracy. Further, we would include expanding our dataset to focus not only for textual but also for images as well as videos.

TABLE IV
ACCURACIES OBTAINED ON DIFFERENT DATA SET USING NAIVE BAYES CLASSIFIER

| Datasets | TF-IDF Vectorizer without Cosine Similarity | TF-IDF Vectorizer with Cosine Similarity |
|---|---|---|
| Merged Dataset | 0.801 | 0.816 |
| LIAR dataset | 0.172 | 0.185 |
| George McIntire | 0.807 | 0.843 |

TABLE V
ACCURACIES OBTAINED ON DIFFERENT DATA SETS USING RANDOM FOREST CLASSIFIER

| Datasets | TF-IDF Vectorizer without Cosine Similarity | TF-IDF Vectorizer with Cosine Similarity |
|---|---|---|
| LIAR dataset | 0.348 | 0.356 |
| George McIntire | 0.761 | 0.839 |

## REFERENCES


[1]W. W.Yang. " liar, liar pants on fire: A new benchmark dataset for fake news detection," arXiv preprint arXiv:1705.00648, 2017.

[2]https://www.kdnuggets.com/2017/04/machine-learning-fake-news-accuracy.html

[3]K. Shu, D. Mahudeswaran, and H. Liu. "FakeNewsTracker: a tool for fake news collection, detection, and visualization," Computational and Mathematical Organization Theory, pp. 1-2, 2018.

[4]R. Mihalcea, and C. Strapparava. "The lie detector: Explorations in the automatic recognition of deceptive language." Proceedings of the ACL-IJCNLP Conference Short Papers. Association for Computational Linguistics, 2009.

[5]S. Feng, R.Banerjee, and C. Yejin, "Syntactic stylometry for deception detection." Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012.

[6]H. Zhen, et al. "Deceptive review spam detection via exploiting task relatedness and unlabeled data." Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2016.

[7]S. Gilda, "Evaluating machine learning algorithms for fake news detection," IEEE 15th Student Conference on Research and Development (SCOReD), 2017.

[8]A. Purwar,, S. K. Singh, and P. Kesarwani. "A Decision Model to Predict Clinical Stage of Bladder Cancer." Soft Computing: Theories and Applications. Springer, Singapore, 829-838, 2018.