International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
SKRGC Since 2012

http://www.ijcsjournal.com Volume 10, Issue 2, No 1, 2022. ISSN: 2348-6600
Reference ID: IJCS-423 PAGE NO: 2838-2842

Fake News Detection

Mohamed Abdullah[#1], Mr.N.Ganapathiram[*2]

[#]*Student, B.Sc Computer Science, Rathinam College of Arts and Science,*
Coimbatore, Tamil Nadu, India -641021
mohdabdullahsaeed98@gmail.com

[*]*Assistant Professor, Department of Computer Science,*
Rathinam College of Arts and Science, Coimbatore, Tamil Nadu, India -641021
ganapathiram.cs@rathinam.in

Abstract - The purpose of this thesis is to assist in automating the detection of Fake News by identifying which features are more useful for different classifiers. The effectiveness of different extracted features for Fake News detection are going to be examined. When classifying text with machine learning algorithms features have to be extracted from the articles for the classifiers to be trained on. In this thesis, several different features are extracted: word counts, ngram counts, term frequency-inverse document frequency, sentiment analysis, lemmatization, and named entity recognition to train the classifiers. Two classifiers are used, a Random Forest classifier and a Naïve Bayes classifier. Training on different features combined with different machine learning algorithms yields different accuracies. By testing the different features on different classifiers, it can be determined which features are the best for Fake News detection. Classifying news articles as either Fake News or as not Fake News is explored using three datasets, which in total contains over 40,000 articles. One of the datasets is used to partly to train the classifiers and partly to test the classifiers. The remaining two datasets are used purely for testing the classifiers.

Keywords—Random Forest classifier, Naïve Bayes classifier, Datasets.

I. INTRODUCTION

The purpose of this thesis is to assist in automating the detection of *Fake News* by identifying which features are more useful for different classifiers. The effectiveness of different extracted features for *Fake News* detection are going to be examined. When classifying text with machine learning algorithms features have to be extracted from the articles for the classifiers to be trained on. In this thesis, several different features are extracted: word counts, ngram counts, term frequency-inverse document frequency, sentiment analysis, lemmatization, and named entity recognition. Two classifiers are used, a Random Forest classifier and a Naïve Bayes classifier. Training on different features combined with different machine learning algorithms yields different accuracies. By testing the different features on different classifiers, it can be determined which features are the best for *Fake News* detection. Classifying news articles as either *Fake News* or as not *Fake News* is explored using three datasets, which in total contains over 40,000 articles. One of the datasets is used to partly to train the classifiers and partly to test the classifiers. The remaining two datasets are used purely for testing the classifiers. All the code used in conjunction with thesis can be found in Appendix B. The term *Fake News* has many definitions, for this paper we will be using Axel Galfert's. "Fake news is the deliberate presentation of (typically) false or misleading claims as *news*, where the claims are misleading *by design.*" Although some form of *Fake News* has been around for many years, it is now mainstream and is widely considered to be a major issue. The 2016 presidential election and Brexit are clear examples of the relevance of *Fake News* in modern society. With the nature of the Internet as it is, anybody can spread untrue and biased information. It is virtually impossible to prevent *Fake News* from being created. Therefore, the next best thing is to find a way to identify and differentiate *Fake News* from *real* news. One of the ways to determine validity is to fact check, but this is time consuming and requires skills that are not shared by everyone. The next best thing is to automate the detection of *Fake News* by using the methods and techniques of Data Science.

Data Science is an interdisciplinary field that tries to find patterns in data that, in this case, may enable society to differentiate between *Fake News* and *real* news. In addition, coupled with algorithms and large sets of data, Data Science can give the necessary insight to help find patterns within the data that would otherwise take a long time to discover or never be discovered at all. Artificial Intelligence (AI) and machine learning in particular, may be used to detect patterns that may characterize *Fake News* when the human eye cannot see it clearly.

Only in the past few decades, there has been an effort to use AI to detect types of deception. Additionally, in the past few years there has been an effort to use AI to detect *Fake News*. The majority of the research has not focused on full news articles, but on short statements. Most of the research that has been done is on small pieces of text that may vary in length; a sentence to a few sentences generally derived directly from tweets or text messages. One of the more predominant datasets, LIAR, is derived from politifact's database of statements. The LIAR dataset has 6 levels of truth values, and includes author data Datasets that use full sized articles are not as prevalent. This is because it is much easier to label a single statement rather than a full-length article.

There are a few datasets that contain full length articles such as FakeNewsNet which is a small dataset with supplementary data. This data was collected from articles posted to twitter and contains data such as the profile of the user who posted the article and other social media context]. Another dataset, called BS DETECTOR, lists websites and their labels; the labels include, among others, fake, conspiracy, and bias . Only URL, no articles, are provided and many of these sites are no longer operational or even available. Therefore, gathering articles from this dataset's sources is complicated and sometimes impossible. Lastly, there is the ISOT Fake News Dataset which contains over 40,000 articles and is far larger than all other datasets readily available . However, all articles in this dataset that are labeled "true" are from Reuters. These "true" articles skew the data because machine learning algorithms may detect the style of Reuters authors or editors and "learn" to label news as "not fake" if it fits this pattern.

In the next section I will discuss of previous research into *Fake News* detection. Afterword, we will examine the datasets used for both training and testing. Then we will go over feature extraction, and the different methods of feature extraction will be discussed. Then a basic introduction into the classifiers that are used is presented. Next, we will examine and discuss the results from the trained classifiers. Finally, we will explore future research.

II. METHODS

A) Data Sets:

To find patterns in Fake News, first news needs to be collected and labeled. Both Fake News and legitimate news needs to be represented in roughly equal amounts. This is to avoid the frequency of Fake News in the dataset being used as a determining factor in classifying. Having good data is essential producing valid results. Good data in this context is data that is representative of the real world and is generalizable.

The dataset used to train the classifiers is the ISOT Fake News Dataset, the largest available dataset of full length Fake News articles. The ISOT dataset contains 21,417 articles labeled Real and 23,481 that are labeled Fake, totaling 44,898. FakeNewsNet is another data set containing full length articles, however there are only 422 labeled articles in it. And lastly there is a set of 180 articles, 90 Fake and 90 Real, collected by the author which, will be referred to as the Original Data. These two additional datasets will be used to test the accuracy of the trained classifiers.

Each model will initially be trained with 80% of the ISOT data. The remaining 20% of the ISOT data will be used to test the accuracy of the trained classifiers. As mentioned, FakeNewsNet and the Original Data will be used for testing as well. The reasoning behind using these additional tests is to make sure we are detecting Fake News and not some other pattern of the ISOT dataset, such as a style of a particular news organization.

Each article labeled as Real in the ISOT dataset was collected from Reuters; all articles their started with the word "Reuters". This pattern could easily be picked up by humans and machines alike. To avoid this issue the beginning "Reuters" phrase was removed from each article.

B) Features:

To find patterns, several different features should be tested. Features are numeric values that describe the text. Examples of these numeric values are word count or the number of times a particular punctuation mark is used. Some features will be more helpful than others, for instance the number of verbs is more likely to be useful compared to the number of times a particular word is used, such as 'kitten'. The goal is to

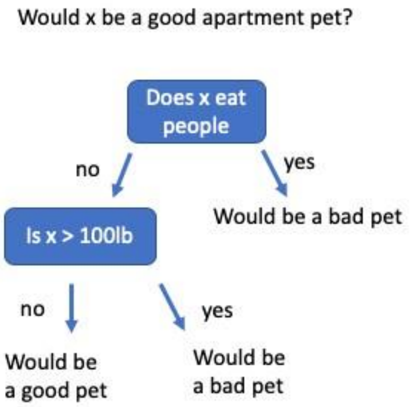
International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
SKRGC Since 2012
http://www.ijcsjournal.com Volume 10, Issue 2, No 1, 2022. ISSN: 2348-6600
Reference ID: IJCS-423 PAGE NO: 2838-2842

find the features that are most helpful in detection of Fake News. Next, each extracted feature will be discussed in detail.

Word counts are among the most easily obtained features that can be extracted from raw text. It is simply a count of all the terms in a body of text. Word counts are also called a ‘bag of words’, however, to keep names descriptive, we shall call this type of feature a count. To get the word count in texts, scikit-learn’s CountVectorizer is used; the CountVectorizer tokenizes the data and then counts each term . The data can be tokenized by word or by n-gram. N-grams are series of n items, such as words or characters. In this thesis n-grams refers to groupings of two and three characters. For instance, the n-grams of the word ‘feature’ would be as follows: ‘fe’, ‘ea’, ‘at’, ‘tu’, ‘ur’, ‘re’, ‘fea’, ‘eat’, ‘atu’, ‘tur’, and ‘ure’. These features will be referred to as countword and count-ngram respectively.

C) Classifiers

As previously mentioned, the extracted features were used to train classifiers. The classifiers used are now discussed. Naïve Bayes, NB, is a type of classifier that takes each feature and treats it as unrelated to any other feature. It then calculates the probability that the particular feature belongs to a classification. It does that for each feature and then aggerates each individual probability to calculate the final classification. For example, with a count-word it would calculate the probability that the count of the first word would belong to *Fake News* as opposed to not. This process will continue for every word and these probabilities a final decision would be made.

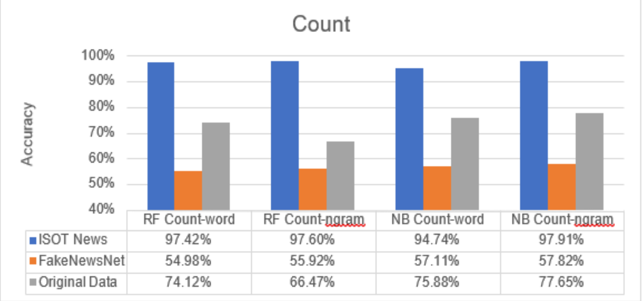


Before describing the next classifier, we will consider decision trees. A decision tree classifier takes the values of the features and splits them into two groups such that each group is as close as possible to only having a single classification. This is repeated until each group consists of a single classification. See Figure 1, for a visual example of a main tree. The main issue with decision trees is that they do not generalize very well. They tend to fit the training data so well that the general patterns in the data are over looked.

This is where the next classifier comes in. Random Forests are a type of classifier built out of a collection of decision trees. But, instead of each decision tree training on all of the data, each decision tree gets a random subset of the data to train on. Making each decision tree in the forest unique. When classifying, each decision tree in the forest gives its own classification, then whichever classification gets the most votes of the decision trees wins. *DataFunctions.py* contains function for reading datasets from files, splitting training and testing data, training classifiers, testing classifiers, and printing results. *FeatureExtraction.py* contains functions for feature extraction. *FeatureTest.py* uses the function from *FeatureExtraction.py* and *DataFunctions.py* to test the different features. *RemoveReuters.py* simply contains the code used to remove the Reuters headers from the news articles.

III. EXPERIMENTAL RESULTS

Using two different models, each extracted feature was tested. The models used were Random Forest (RF) and Naïve Bayes (NB). There is some difference between the two classifiers. There is a much larger difference between datasets. The following is a detailed discussion of each set of features. We will compare features and classifiers by their accuracy, which is the percentage of correct classification made by the classifier.

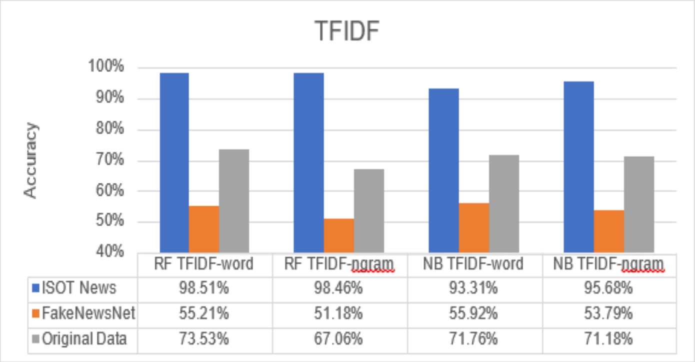


	RF Count-word	RF Count-ngram	NB Count-word	NB Count-ngram
■ISOT News	97.42%	97.60%	94.74%	97.91%
■FakeNewsNet	54.98%	55.92%	57.11%	57.82%
■Original Data	74.12%	66.47%	75.88%	77.65%

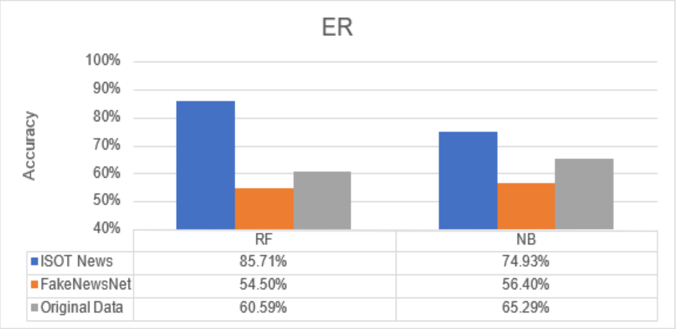
Count-word and Count-ngram: First, most notable the ISOT testing data is getting way higher accuracy results than either the Original dataset or the FakeNewsNet dataset. After the ISOT, the Original dataset is getting the next

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
SKRGC Since 2012

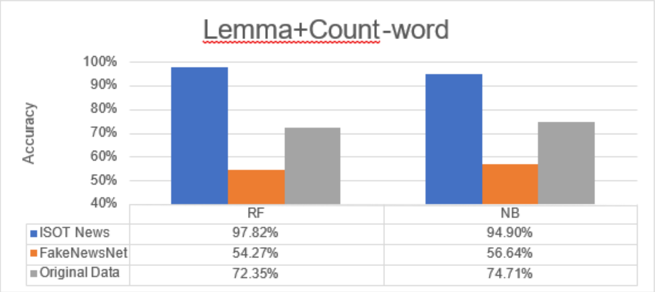
highest accuracy rates. This suggests that the Original dataset is closer in makeup to the ISOT dataset than the FakeNewsNet is. Next the data shows that the NB classifier generalizes better than the RF classifier. The NB classifier gets better accuracy rates with count-ngram. The RF has no clear winner between count-word and count-ngram.



TFIDF-word and TFIDF-ngram: As seen in Figure 3, the ISOT testing data has the highest accuracies again. The random forest classifiers get better results with the ISOT dataset than the Naive Bayes. However, the NB does generalize better to the Original dataset and the FakeNewsNet dataset. TFIDF-word is getting better accuracy rates over TFIDF-ngram. In the case of the RF's classification of the Original dataset, the TFIDF-word is getting 6.47% more accuracy. Again, the Original dataset is being classified better than the FakeNewsNet dataset. Between TFIDF and Count, the Count-ngram is getting the best accuracy results.



Lemma: Once again, ISOT accuracies are the highest, with Original coming in second. The NB classifier is still generalizing better than the RF classifier. The results from lemma are better than some of the other features. However, lemma with a Count-word is not as accurate as a Countword. Suggesting that the different forms of a word are helpful to the classifier.



From the results a few more general conclusions can be made. The most notable is that the accuracy on the ISOT test data is much higher than the accuracies of the other datasets. From this, it can be concluded that there is a pattern in the ISOT dataset that is being picked up by the two classifiers. However, it appears that these patterns do not generalize well to the other available datasets. The patterns that the classifiers are picking up on could be a pattern found in Reuters articles, or could be another pattern that exists mainly in the ISOT dataset Such as article topic, or political leaning. All of this suggests that ISOT is not a good dataset to train with. Next, it can be seen that the Original dataset is classifying with better accuracy than the FakeNewsNet dataset. The Original dataset does not contain articles from Reuters hence, this does not explain the jump in accuracy. Therefore, it is possible that the *Fake News* within the ISOT and Original dataset are closer in underlaying structure.

For accuracy rates, it can be concluded that Counts and TFIDF generalize better than ER, PoS, and VADER. More tests should be done with ER, PoS, and VADER features before any of them are discarded for providing lower accuracy rates. They still may be a benefit to accuracy rates when combined with other features, despite not doing well by themselves.

IV. CONCLUSION

With the nature of the Internet as it is, Fake News is easily created and distributed. Fact checking is tedious and time consuming, so automating Fake News detection is critical. Thus Fake News classifiers should be created. However, a classifier does not come out of thin air, it must be trained on already existing data. The quality and quantity of the data is important. Three datasets were used for the research in this thesis. ISOT, a huge dataset of over 40,000 articles. FakeNewsNet is another, much smaller dataset containing

422 articles. Lastly, the Original dataset, containing 180 articles, that was gathered specifically for this research. However, a classifier cannot read, so it must have features extracted for the articles. A feature is a numeric value extracted from the article. Such as a word count, or a count of parts of speech, or more complicated features. Such as a count of the named entities, like businesses or organizations.

Two different classifiers, Random Forests and Naive Bayes, were trained on the 80% of the ISOT dataset reserved for testing using each of the ten different features: Count-word, Countngram, TFIDF-word, TFIDF-ngram, PoS, ER, Lemma, VADER, NLTKStop, and spaCyStop. Then each classifier was tested on the remaining 20% from ISOT, all of FakeNewsNet, and all of the Original dataset. The accuracy results where then examined and conclusions were drawn. The ISOT dataset did not generalize well to the other two datasets used for testing. Making the test results for the 20% testing portion of ISOT get way higher results than the other two datasets. This could be found the fact that ISOT got all of its real news from Reuters and the classifiers ended up being a Reuters vs not-Reuters classifier. Next it was discovered that Count/TFIDF are better standalone features than PoS, ER, and VADER. However, these features still have potential to be used in conjunction with other features. Although Lemma was one of the better features, it was outperformed by Count-word, suggesting that some of the removed data was improving the classification.

SCOPE OF FUTURE WORK

A different dataset should be used to train classifiers to verify the result obtained with ISOT. The size of ISOT makes it a valuable dataset, however, it is probably best as a testing dataset than a training dataset. Using combinations of the features should be explored. For instance, combining ER with lemma. Even more testing with VADER scores could be beneficial. The best accuracy rate on the Original data set was achieved with a Naïve Bayes classifier with a word count feature after NLTK stop words were removed. These results should be explored more, for example which words when removed provide the greatest increase in accuracy. Additionally, it should be look into if the removal of any the NLTK stop words actually harm the overall accuracy of the classification. One thing that has not been tried is differentiating and classifying real news, satire, and Fake News. This would be valuable because satire is a type of deceptive news that isn't Fake News. Hence, we should avoid labeling it as such.

REFERENCES

[1] A. Gelfert, "Fake news: A definition," Informal Log., vol. 38, no. 1, pp. 84–117, 2018.

[2] K. Rogers and J. Bromwich, "The Hoaxes, Fake News and Misinformation We Saw on Election Day," NY Times, New York, 08-Nov-2016.

[3] I. Witten, E. Frank, and M. Hall, Data Mining 4th Edition. 2016.

[4] W. Y. Wang, "'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017, pp. 422–426.

[5] D. Byrd, "The science of fake news gets a boost," 2018. [Online]. Available: http://earthsky.org/human-world/fake-news-mar-2018-article-science-calling-for-studies.

[6] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media," 2018.

[7] "BS DETECTOR." [Online]. Available: https://github.com/selfagency/bs-detector.

[8] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, 2017, pp. 127–138.