Analysing Machine Learning Algorithms for Fake News Detection

Manush Bhatt
Purdue University
West Lafayette, Indiana, United States of America
bhatt16@purdue.edu

Shoban Kumar Rajamani Vimalarani
Purdue University
West Lafayette, Indiana, United States of America
srajaman@purdue.edu

ABSTRACT

In our modern era where internet is ubiquitous, everyone relies on various online resources for news. Along with the increase in use of social media platforms like Facebook, Twitter etc. news spread rapidly among millions of users and within a very short span of time. The model focuses on identifying fake news sources, based on multiple articles originating from a source. Once a source is labeled as a producer of fake news, we can classify it with high confidence that any future articles from that source will also be fake. Focusing on sources widens our article misclassification tolerance, because we then have multiple data points coming from each source.

1 INTRODUCTION

According to the definition on Wikipedia, Fake news is a type of yellow journalism or propaganda that consists of deliberate disinformation or hoaxes spread via traditional print and broadcast news media or online social media. Any news story written with an intent to mislead in order to cause damage can be classified as fake. This also includes the news articles or blogs containing a mixture of real as well as manipulated facts created with a malicious intent.

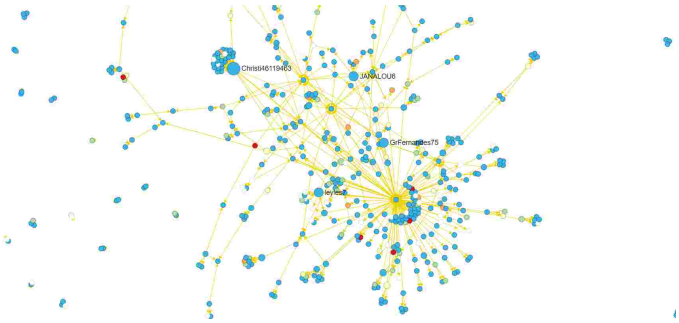
There are various reasons why this problem has become so eminent recently. With the growth of social networking websites such as Facebook and Twitter, it has become very easy to spread and share information. On top of that, there are various fake news websites created with the sole purpose of misleading people to generate revenue and publicity through ill-natured ways.

We have observed the impact of spreading of fake news several times in last decade including the latest US presidential elections. Fake news impact the readers by swinging their beliefs. It is also believed that circulation of fake news had material impact on the outcome of the 2016 US Presidential Election [1]. Zuckerberg accepted that identifying fake news is difficult, writing, "This is an area where I believe we must proceed very carefully though". Identifying the truth is complicated. It is increasingly becoming a threat to our society. It is typically generated for commercial and monetary purposes. However, people and groups with malicious agendas have used it as a tool to initiate fake news in order to influence events and policies around the world.

1.1 How does it spread?

Figure 1 shows the example of a single news story spreading via a subset of users on Twitter. Social connections between users are shown by yellow directed arrow. A Tweet is started by a single real user and re tweeted subsequently by various connections. The red dots denote that there is a high chance of that user being a bot based on the user's previous activity. Similarly, the blue dots denotes that the user is a human. And orange, yellow and green dots are the different levels of probability of the user being a bot.

Figure 1: Spreading of a single news article



Bots play a significant role in the spread of fake news. They enable low-credibility stories to gain enough momentum so that they can later go viral. Another strategy involves targeting people with many followers, either by mentioning those people specifically or replying to their tweets with posts that include links to low-credibility content.

1.2 Prior Works

Previous work on fake news detection (Rubin et al., 2016)[2] really have shown to be the distinguishing factor between true and fake news.The text understand ability features such as the number of characters, complex words, dominant words, long words, number of syllables, word types, and number of paragraphs, among others features played a major role in detecting the fake news (Papacharissi Oliveira, 2012)[3]. We extract a set of features derived from production rules based on context free grammars which improves the prediction as shown in (Feng et al., 2012)[4].Once the features are extracted, we refer to use machine learning algorithms for fake news detection(Shlok Gilda)[5] and Automatic Online Fake News Detection Combining Content and Social Signals (Marco L. Della Vedova et al.)[6] to evaluate multiple models and compare its results.

2 DATASET

The dataset for this project was built with a combination of both real and fake news.The datasets used for this project were drawn from Kaggle[7][8]. [7]contained all real news data collected from around 15 publications like The New York Times, Breitbart, CNN, Business Insider, The Atlantic, Fox News, The Washington Post, etc.The data primarily falls between the years of 2016 and July 2017.

[8] contained all the fake news data collected from over 200 websites which mainly focused on domains pertaining to sports, business, entertainment, politics, technology, and education. The real and fake data were then merged and shuffled to get a CSV file containing a consolidated randomized dataset. From the consolidated randomized dataset we picked 20000 records at random which contained approximate 50% real news and 50% fake news articles. From these records, 80% was used for training and validating the detection model and 20% was reserved for testing the model.The challenge we faced was how do we consolidate the datasets.Fake dataset has information specific to few domains whereas the Complete dataset has lot of other fields.We tried to oversample and undersample the complete dataset and came up with the final consolidated csv.

2.1 Feature Extraction

The complete dataset consisted of a lot of features like title, publication, date, year, month, URL, content and author. But,the fake dataset had features like domain_rank, crawled, likes, comments, shares, spam_score in addition to the above features. So, we had to combine these two dataset and remove features which were not of higher importance like crawled, likes, comments and shares. The final data we had features like:

- id: unique id for a news article
- title: the title of a news article
- author: author of the news article
- text: the text of the article; incomplete in some cases
- label: a label that marks the article as 0: unreliable and 1: reliable
- spam_score : spam score of the fake data.

2.2 Text Preprocessing

Since most of the data was crawled and extracted manually, we had to first go through the data to understand organization and formatting of text. The data was made uniform and comparable by converting it into a uniform UTF-8 encoding. There were some cases where we encountered weird symbols and letters incompatible with the character set which had to be removed. We noticed that the data from news articles were often organized into paragraphs. So, we performed trimming to get rid of extra spaces and empty lines in text. Then, we removed unnecessary punctuation and stop words from the dataset. After this, we performed Lemmatization on the columns such as title, author and text instead of Stemming. We extracted uni-grams and bi-grams derived from the bag of words representation of each news articles.

3 FEATURE GENERATION

For generation of features from the given data, we first performed tokenization on the raw text of articles. We then generated tf-idf feature vectors as described below.

3.1 Term Frequency - Inverse Document Frequency

The tf-idf is a statistical measure that reflects the importance of a particular word with respect to a document in the corpus. It is often used in information retrieval and text mining as one of the components for scoring documents and performing searches. It is a weighted measure of how often a word occurs in a document relative to how often it occurs across all documents in the corpus. Term frequency is the number of times a term occurs in a document. Inverse document frequency is the inverse function of the number of documents in which it occurs.

$$W_{i,j} = f_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Hence a term like ”the” that is common across a collection will have lesser tf-idf values, as its weight is diminished by the idf component. Hence the weight computed by tf-idf represents the importance of a term inside a document. The tokenized data was used to generate a sparse matrix of tf-idf features for representation. This represented our feature vector and was used in subsequent prediction algorithms.

4 PREDICTION MODELS

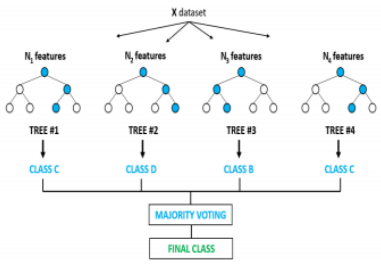
The most common algorithms used by fake news detection systems include machine learning algorithms such as Support Vector Machines, Random Forests, Decision trees, Stochastic Gradient Descent, Logistic Regression and so on. In this project we have attempted to implement four algorithms to train and test our result-Multinomial Naive Bayes, Random Forest, Adaboost Algorithm, Logistic Regression. The main challenge throughout the project has been to build a set of uniform clean data and to tune parameters of our algorithms to attain the maximum accuracy.

We performed k-fold cross validation with k=5 and split the 20% of training set into validation and compared the results to prevent overfitting of the models.

4.1 Random Forest Algorithm

Random Forests are a machine learning method of classification that work by building several decision trees while training the model. It is a kind of additive model that makes predictions from a combination of decisions from base models. Decision trees have huge depth and tend to overfit results. Random forest utilizes multiple decision trees to average out the results. The Random forest classifier creates a set of decision trees from a subset of the training data. It aggregates the results from different decision trees and then decides the final classification of the test data. The subsets of data used in the decision trees may overlap. In general, the more trees in the forest, the more robust the forest looks like.

Figure 2: Random Forest Model



In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. Decision tree concept is more to the rule based system. Given the training dataset with targets and features, the decision tree algorithm will come up with some set of rules. The same set rules can be used to perform the prediction on the test dataset.

4.2 Logistic Regression

Logistic Regression is a Machine Learning technique used to estimate relationships among variables using statistical methods. This algorithm is great for binary classification problems as it deals with predicting probabilities of classes, and hence our decision to choose this algorithm as our baseline run. It relies on fitting the probability of true scenarios to the proportion of actual true scenarios observed. Also, this algorithm does not require large sample sizes to start giving fairly good results. It uses gradient descent to converge onto the optimal set of weights (θ) for the training set. For our model, the hypothesis used is the sigmoid function:

$$h_\theta(x) = g\left(\theta^T x\right) = \frac{1}{1+\exp(-\theta^T x)}$$

4.3 Multinomial Naive Bayes Algorithm

Multinomial Naive Bayes is a specialized version of Naive Bayes that is designed more for text documents. Whereas simple naive Bayes would model a document as the presence and absence of particular words, Multinomial Naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with in. So, we tried to use this model to find the performance on our fake news detection problem. The term frequencies described above can then be used to compute the maximum-likelihood estimate of the Multinomial Naive Bayes Classifier based on the training data to estimate the class-conditional probabilities of the fake news detection. This can be represented as:

$$\hat{P}\left(x_i|\omega_j\right) = \frac{\sum tf-idf(x_i, d\in\omega_j)+\alpha}{\sum N_{d\in\omega j}+\alpha\cdot V}$$

where,

- x_i: A word from the feature vector x of a particular sample.
- $\sum tf - idf(x_i, d \in \omega_j)$: The sum of raw term frequencies of word xi from all documents in the training sample that belong to class ω_j.
- $\sum N_{d\in\omega_j}$: The sum of all term frequencies in the training dataset for class ω_j.
- α: An additive smoothing parameter (α=1 for Laplace smoothing).
- V: The size of the vocabulary (number of different words in the training set)

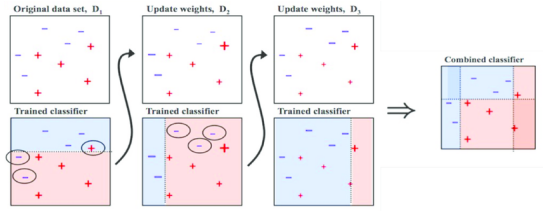
The class-conditional probability of encountering the text x can be calculated as the product from the likelihoods of the individual words:

$$P(\mathbf{x} \mid \omega_j) = P(x_1 \mid \omega_j)\cdot P(x_2 \mid \omega_j)\cdot \ldots \cdot P(x_n \mid \omega_j) = \prod_{i=1}^{m} P(x_i \mid \omega_j)$$

4.4 Adaboost Algorithm

Boosting is a general technique by which multiple "weak" classifiers are combined to produce a "super strong" single classifier. The idea behind boosting technique is very simple. Boosting consists of incrementally building a final classifier from an ensemble of classifiers in a way such that the next classifier chosen should be able to perform better on training instances that the current classifier is not able to do.

Figure 3: Adaboost



The concept of a "weak" learner is that the classifier is able to do classification with an error rate of less than $\frac{1}{C}$, where C is the number of classes. Adaboost is one such boosting algorithm, which "greedily" builds up the final classifier from a pool of weak classifiers. It assigns weights to each weak classifier such that classifiers having higher accuracy are given more weights. More importantly in each iteration of the Adaboost algorithm, it assigns weight to each training example. Training examples which are hard to classify are assigned more weights. Thus, it is represented as:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

where,

- $h_t(x)$: The output of weak classifier t for input x
- α_t: weight assigned to classifier

5 METHOD EVALUATION

5.1 Evaluation Metrics

We used the following three metrics for the evaluation of our results. The use of more than one matrix helped us evaluate the performance of the models from different perspectives.

(1) **Accuracy** This depicts the number of accurate predictions made out of the total number of predictions made. Classification accuracy is calculated by dividing the total number of correct result by the total number of test data records and multiplying by 100 to get the percentage.

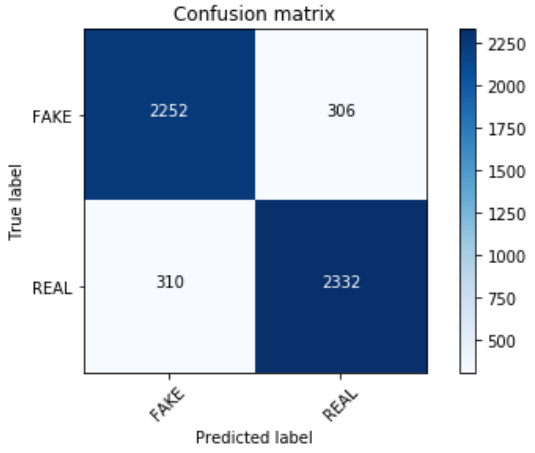
(2) **Confusion Matrix** This is a great visual way to depict the predictions as four categories: 1. False Positive: Predicted as fake news but are actually true news. 2. False Negative: Predicted as true news but are actually fake news. 3. True Positive: Predicted as fake news and are actually fake news. 4. True Negative: Predicted as true news and are actually true news

(3) **Precision and Recall** Precision which is also known as the positive predictive value is the ratio of relevant instances to the retrieved instances. Precision = No. of True Positives / (No. of True Positives + No. of False Positives) Recall which is also known as sensitivity is the proportion of relevant instances retrieved among the total number of relevant instances. Recall = No. of True Positives / (No. of True Positives + No. of False Negatives)

5.2 Evaluation Results

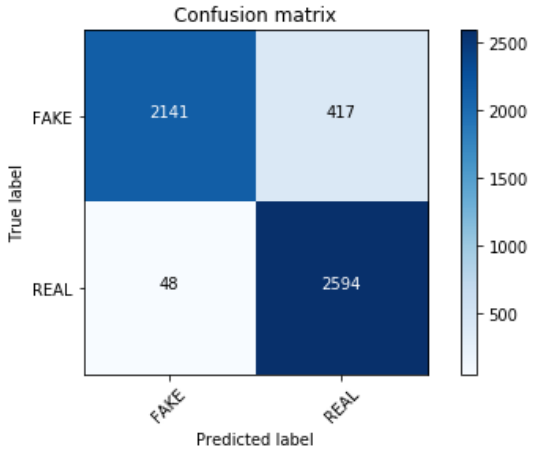
We use confusion matrix to describe the performance of each of the models and later compare their accuracies, precision and recall scores

Figure 4: Confusion Matrix for Random Forest



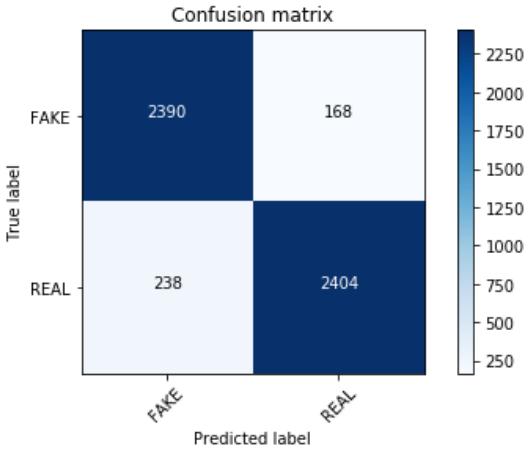
The hyper parameters used for the model includes number of estimators as 10, gini impurity to find the model split and maximum depth of 30. The accuracy of random forest was 0.86. The precision, recall and f1 scores were 0.855, 0.85 and 0.853 respectively. From this data, we can say that random forest gives us good accuracy. Therefore, we decided to use random forest as the base model for all other model comparisons.

Figure 5: Confusion Matrix for Multinomial Naive Bayes



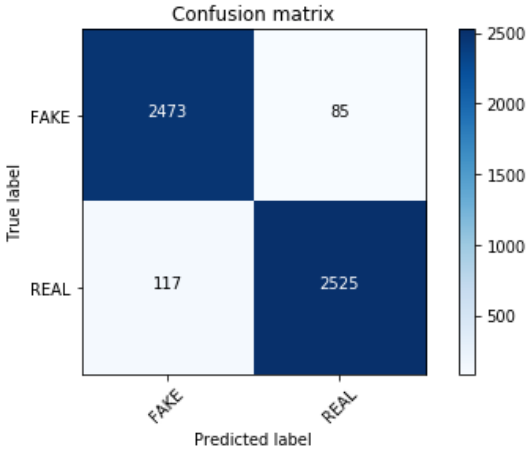
We chose to use multinomial naive bayes as it is highly suitable for text classification with discrete features (words in our dataset). Our accuracy score was 0.91. The interesting thing to note was that we received a very high precision score of 0.978 but a very low recall score of 0.836.

Figure 6: Confusion Matrix for Adaboost using Decision Tree



For adaboost, we chose the Decision Tree Classifier as the base estimator and kept the number of estimators to 50. The observed accuracy was 0.92. The precision and recall scores were 0.91 and 0.93 respectively.

Figure 7: Confusion Matrix for Logistic Regression



We got the best results using logistic regression. We used l2 normalization with tolerance of 10^{-3}. The accuracy score was 0.96. Our precision and recall scores were 0.954 and 0.966 respectively.

From the results, we concluded that logistic regression gave the best accuracy among the 4 models.

5.3 Model Comparison

Figure 9 explains the results of comparison between the four models when the news are preprocessed with Stemming and then tf-idf approach is used.

Figure 8: Feature Importance

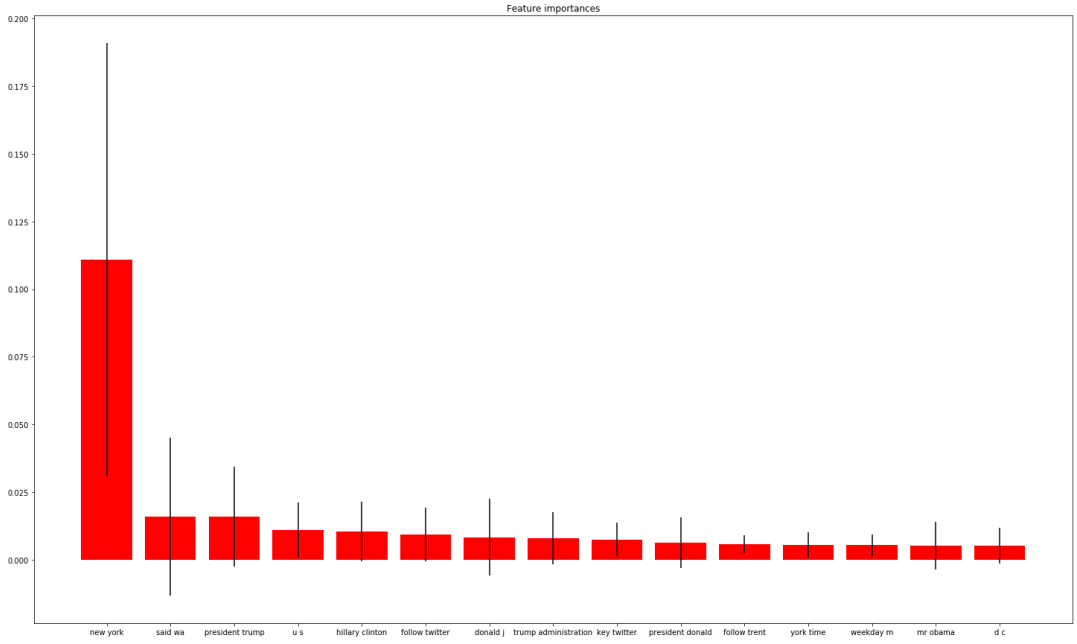


Figure 9: Model Comparison with Stemming and TF-IDF

Models	Accuracy	Precision	Recall	F1 Score
Random Forest	0.76	0.755	0.77	0.77
Multinomial Naïve Bayes	0.84	**0.901**	0.766	0.832
Adaboost	0.83	0.83	0.84	0.84
Logistic Regression	**0.9**	0.89	**0.9**	**0.9**

The results are not that accurate and so we tried with Lemmmatization instead of Stemming and used same tf-idf vector approach in top of that. Here are the results we got from this approach.

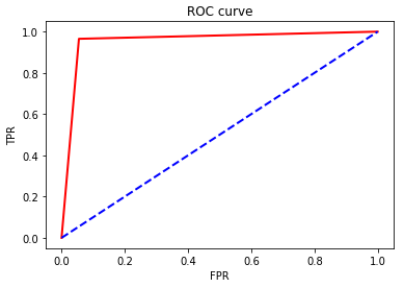
Figure 10: Model Comparison with Lemmatization and TF-IDF

Models	Accuracy	Precision	Recall	F1 Score
Random Forest	0.86	0.855	0.85	0.853
Multinomial Naïve Bayes	0.91	**0.978**	0.836	0.902
Adaboost	0.92	0.91	0.93	0.922
Logistic Regression	**0.96**	0.954	**0.966**	**0.96**

The results above are very much promising in general but if we see in depth the number of fake news classified as real is still in higher range which needs to effectively reduced to less than one percent so that people or consumer are not misguided by these news.

5.4 ROC Curve

Figure 11: ROC curve



In a Receiver Operating Characteristic (ROC) curve the true positive rate (Sensitivity) is plotted in function of the false positive rate (Specificity) for different cut-off points. The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between the two classes real and fake. Figure 11 shows ROC curve for logistic regression

6 EVALUATION RESULTS

The following are the results for this project:

- Surprisingly, Logistic Regression fared far better when compared to other models such as Multinomial Naive Bayes and Random Forest.This is mainly because of the type of data we were dealing with.

- However,Multinomial Naive Bayes has more precision among all the models which makes it quite interesting.
- Tf-idf shows promising potential predictive power, even when ignoring named entities, but we remain skeptical that this approach would be robust to changing news cycles. This would require a more complete corpus however.
- Lemmatization gives better accuracy than the Stemming as Lemmatization gives better logical grouping of tf-idf vector than Stemming.

7 NOTEWORTHY INSIGHTS

7.1 Feature Importance

Figure 8 gives us top 15 important features from the data set. Our data comprised mostly of content taken during US presidential elections. Therefore, words like president, Donald Trump and Hilary Clinton are more prominent. Apart from that some of the words like Twitter and follow show that the importance of role of social media during the elections.

7.2 Frequency Distribution

We used n-gram range of 1 to 2 in our implementation. This helps in getting words containing full names as well as some city names which we observed in the data set.

Figure 12: n-gram distribution

The uni-gram and bi-grams can be seen in figure 12. As inferred from feature importance, many of the words are related to the elections. Words like New York, United States are also more notable among the few. But we observer some of the outliers as well such as even now, still well and said which need to be taken care off in our future work. We only used n-gram range of 1 to 2 because of the poor performance of the models and performance constraints.

8 FUTURE WORK

In our future work, We plan to aggregate the well-performed classifiers to achieve better accuracy. For example, using bootstrap aggregating for Neural Network, and using LSTM and SVM models among others to get a better prediction. An ambitious work would be to search the news on the Internet and compare the search results with the original news. Since the search result is usually reliable, this method(human crowd sourcing) should be more accurate, but it also involves natural language understanding because the search results are not exactly same as the original news. So, we would need to compare the meaning of both contents and decide whether they describe the same thing or not.

9 CONTRIBUTIONS

The contributions of individual group members were as follows:

Shoban was responsible for extraction of the fake and real news dataset, preprocessing the text and the collection and randomization of the cleaned data into CSV files.He also implemented the training of data using Random Forest classifier and Multinomial Naive Bayes classifier using Sci-kit learn library and tested the results for various input sizes using the evaluation measures described.

Manush was responsible for the implementation of the Naive Bayes classifier and Adaboost classifier using open source Sci-kit learn library. He was also involved in consolidating and visualization of results for the same. He performed few other insights such as top k words from the dataset using word cloud.

Both of us were involved in the topic selection, analysis, design and algorithm selection and documentation phases

REFERENCES

[1] Alexandre Bovet and HernÃąn A MakseInfluence of fake news in Twitter during the 2016 US presidential election Nature Communications, 10(1):7, 2019.

[2] Victoria L Rubin, Niall J Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news In Proceedings of NAACL-HLT, pages 7âĂŞ17,2016.

[3] Papacharissi, Z. & Oliveira, M.The Rhythms of News Storytelling on Egypt. Journal of Communication. 62. pp. 266âĂŞ282,2012.

[4] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 171âĂŞ175. Association for Computational Linguistics,2012

[5] Shlok GildaEvaluating machine learning algorithms for fake news detection IEEE 15th Student Conference on Research and Development (SCOReD),2017.

[6] Marco L. Della Vedova,et al.Automatic Online Fake News Detection Combining Content and Social Signals 22nd Conference of Open Innovations Association (FRUCT),2018.

[7] kaggle All the News. https://www.kaggle.com/snapcrack/all-the-news

[8] kaggle Fake News NLP Stuff. https://www.kaggle.com/rksriram312/fake-news-nlp-stuff/notebook

[9] Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre,Rada Mihalcea. Automatic Detection of Fake News Proceedings of the 27th International Conference on Computational Linguistics, pages 3391âĂŞ3401 Santa Fe, New Mexico, USA, August 20-26, 2018