Model Analysis to Provide Aid in Detecting Fake News

Chia-Hsien Ho
Min Kim
Sayaka Kuwayama

Abstract

The presence of fake news on social media and news platforms online has been on the rise in the recent years. In order to come up with a solution to this problem, an effective method of detecting fake news has been needed. The method that we have come up with uses 5 different machine learning baselines - Logistic Regression, Logistic Regression CV (Skip-gram), Logistic Regression (CBOW), Neural Network and Decision Tree. Our proposed models are built upon the 2 separate ISOT datasets with over 20,000 rows each with articles from 2016 to 2017, one holding all the false news articles and one with all the true news with three main columns of 'title' and 'text' for the article and the 'subject' of the article. The true set of articles are gathered from the news website Reuters.com. The false news set is collected from various websites flagged by Politifact, a fact-checking organization, and Wikipedia. In this paper, we compare our models by looking at each of its accuracy, as it provides us with how appropriate the model would be in detecting for fake news. The dataset is split into 80/20 training and test sets. For our result, we choose the logistic regression CV as our final model with accuracy of 99.96%.

1 Introduction

Recently, there has been a surge in the number of psychological crimes involving fake news that has been created to target innocent people for reasons from political control, promoting an ideology, marketing products/ideas, to making money, to stir up a commotion or simply for no reason at all. Especially with the rise of social media, it has become too easy to put information unaccounted for for the public to see. In 2020, there has been over 3.6 billion users on social media, and over the next 5 years, we are expected to see an increase to 4.41 billion users [7] which maximizes the need to stop fake news from being published or take it down before its negative influence expands.

To define false news, it differs from a regular news report that states only facts and is also different from news reports that show factual information but biased information. False news is identified as biased information that is one of *disinformation*, which is unproven information purposely spread to deceive people and misinformation, false or misleading information.[8]

As more fabricated news becomes prevalent, it will more likely cause conflicted opinions that will lead to larger chaos not just amongst readers but also within the news industry. In a Harper's Magazine story published back in 1925, it was warned that, 'Once the news faker obtains access to the press, all honest editors alive will not be able to repair the mischief he can do.'.[4]

Fraudulent news can be the easiest thing for companies or individuals to produce, but once it has been published and spread, it can cause very impactful damage to all around. Within our paper, we try to build a tool through testing on various models to aid this pivotal issue by being able to detect fake news.

Our method of trying to detect fake news in this paper is organized as followed:

- We study a work that is relative to ours (Step 2)
- We demonstrate exploratory data analysis on our data by pre-processing - cleaning the data, running a text analysis as well as a sentiment analysis to discover what our data holds (Step 3)
- We develop numerous models to compare which one would be most efficient in using it to detect fraudulent news.

However, we first classify the processed data into different categories to thoroughly test our data. (Step 4)

- We implement logistic regression to look at the effects of sentiments on the title and text as well as both variables individually
- We implement logistic regression cross validation to look at the effects of variables of just ‘title’ and ‘text’ individually
- For an additional dataset that we created of including both the title and the text, we implement logistic regression cross validation with both skip-gram and CBOW models as well as a decision tree and a neutral network model
- We establish our findings from all of the models that we created and compare them to visualize which model would be the best fit for our case in being able to distinguish fake news (Step 5)
- We explore our final model by testing for its accuracy on another dataset
- We successfully conclude the study by summarizing our findings (Step 6)

2 Related Work

With the problem of prevalence in fraudulent news, there have been many studies done in the past to come up with solutions on ways to detect it better.

Tacchini et al [14] introduced two classification methods to decide whether a news is false or true based on interactions with the users. The first method was to create a logistic regression model using the user interaction as its feature. The second was using the boolean label crowdsourcing technique. This method is where for a certain set of terms, the users would provide boolean labels of true and false. After, the crowdsourcing algorithm would give an estimate of truth or false to each term and to individual users it would give the probability of likelihood that it would be true or false.[3][12] The result was successful as the accuracy for both models concluded that interaction of users had an impact on the type of news.

Prez-Rosas et al [11] tried a different approach with instead of text that just carried pieces of an article, used actual news excerpts to test their models. The classifier they chose was the linear support vector machine classifier to indicate that linguistic features like lexical, syntactic and semantic levels would aid in filtering the types of news. However, unlike many studies, the result was disappointing as the system seemed was not able to catch more than what humans could.

Wang [17] used a LIAR dataset holding just over 12,000 short texts. These articles were gathered from open public sources like Facebook posts and tweets as well as from political debates and etc. Similar to one of our methods, they estimated with a neural network model. The model that they proposed was the Convolutional Neural Network model implementing text and metadata together, where the feature extraction was done from the text and the Bi-directional Long Short Term Memory was used for the extraction of metadata. These two extractions were compiled into a connected softmax layer to be used in making the final conclusion about the types of news. The outcome stated that the concatenation of metadata with text did result in higher accuracy.

3 Preliminaries

Before beginning to build our models, to get familiar with our datasets, we clean and explore the data to make sure it is suitable to run on various baselines. The two datasets that we use, one consisting of the fake news articles with 23481 rows and the other with true news of 21417 rows, are first given an additional column of ‘isTrue’, to define whether or not the article is false or true with a 0 and 1 accordingly. We put the two different datasets together to create one whole dataset carrying both false and true news articles to deliver thorough analysis.

3.1 Exploratory Data Analysis

Upon looking through the whole dataset ‘news’ with 44898 rows, we encounter some empty context in few of the text of the article. The title exists without any text to it and we discover that excluding one true news that just had an image as its text, they are all fake news, where ‘isTrue’ equals 0. We can assume that the fake news was used to target the audience just with the title itself as titles do have an impact since it is most likely what catches the readers' eye first. In addition to empty text, we identify a line of text that is made up of unknown wording within brackets rather than phrases. Summing up the 631 rows of empty text and 1 row of unreasonable text, we create a new dataset excluding these rows. The final cleaned dataset that we implement into our models was of 44266 rows, 22850 counts of fake news and 21416 of true news.

In manipulating the subject column to check for each source of news articles, we group by subject and 'isTrue'. The inputs of subjects are - 'politicsNews', 'worldnews', 'News', 'politics', 'left-news', 'Government News', 'US_News', 'Middle-east'. However, we detect that for 'isTrue' with 1, meaning that it's true news, the only types of subjects are 'politicsNews' and 'worldnews'. The rest of the subjects are made up of false news. Due to this result, we decide to not use the column of subjects into our model for detection since the prediction will show a perfect accuracy of 100% based on the given subject.

		counts
subject	isTrue	
politicsNews	1	11271
worldnews	1	10145
News	0	9050
politics	0	6433
left-news	0	4308
Government News	0	1498
US_News	0	783
Middle-east	0	778

Table 1: Count fake/true news by subject

3.2 Preprocessing

Before being able to carry out any analysis, we pre-process our data first. Pre-processing of the data is converting the dataset so that it becomes a language that the computer can understand. Even though we already got rid of unuseful data, we filter the dataset once more to get rid of useless words, known as 'stop words' within natural language processing (NLP).

'Stop words' are words that are commonly used daily but are ignored when the search engine indexes entries for searching as well as in results from a search query. Examples of these words are "the", "a", "an" and "in". We choose to take these words out as it would just take up processing time if included.

Once we've taken out the English stop words from the Natural Language Toolkit, we also get rid of punctuation that could get in the way of processing by replacing them with a space. In order to keep the original data along with the new pre-processed data, we name the pre-processed data with _token after title and text to mark as the columns that will be used in our models.

3.3 Text Analysis

In understanding our data, we visualize the top 20 word counts for both the title and text of the new articles regardless of the type. The figure below visualizes the most common words for the title variable. We see that words like 'Trump', 'VIDEO', 'says' are said the most throughout these articles. Something that stands out is that there are some repetitive words. Within text analysis, we treat the same words but with lower or capital letters as different words (i.e 'Trump' and 'TRUMP'). We decide to do so as the capital letters in the title or the text shows a form of exaggeration which could be a method for the model to determine whether the news is fake or not. The same idea is carried out for the most common words found in the text.

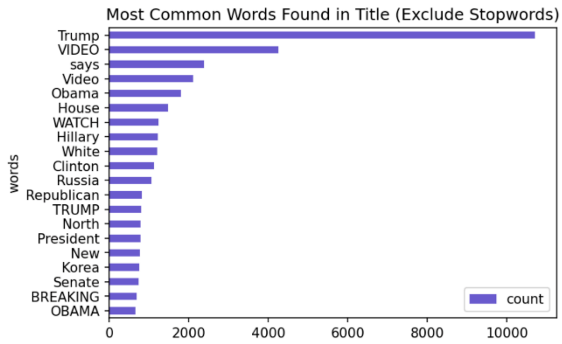


Figure 1: Most common words found in title

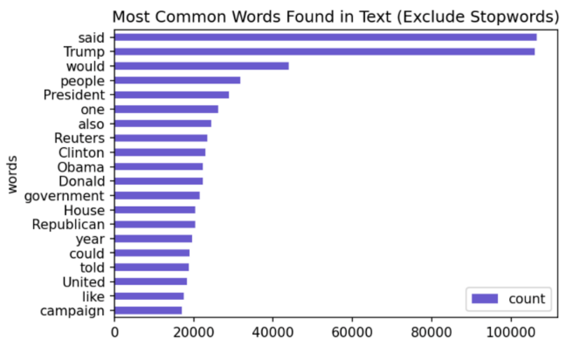
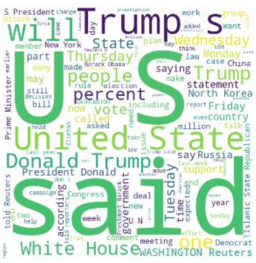


Figure 2: Most common words found in text

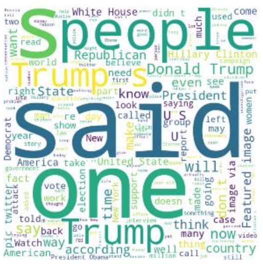
Since Figure 1 and Figure 2 above were text analysis done on the whole dataset of both fake and truthful news, the word clouds from Figure 3 portray the most used word differentiated by the type of news as well as by title and text. Similar to when the news articles were undifferentiated, the words that are most common are the words that stand out as the largest words within the clouds.

(a) Title of fake news

(b) Title of true news



(c) Text of fake news



(d) Text of true news

Figure 3: Word clouds generated from the dataset

3.4 Sentiment Analysis

Within Natural Language Processing, Sentiment Analysis plays an effective role in determining the type of opinion the body of text expresses. The types of opinion can be quantified into a positive or a negative polarity. Through a score given through a polarity analysis, we are able to assign a positive, negative or a neutral polarity to each body of text.

Using ‘SentimentIntensityAnalyzer’ to define a function and ‘polarity_scores’ function from the analyzer, we give our title and text tokens a polarity score. If the score is equal to or over 0.05, it would be considered *positive* polarity, if under or equal to - 0.05, *negative* polarity and any scores out of these ranges are considered *neutral*.

isTrue	0	1	0_percentage	1_percentage
title_sentiment				
negative	12012	8081	0.5257	0.3773
neutral	4889	7744	0.2140	0.3616
positive	5949	5591	0.2604	0.2611

Table 2: Counts and percentage of title sentiment for fake/true news

Table 2 shows the results for the title token. For the title of a news article carrying false news, about 53% of the sentences used infers a negative connotation. Only about 26% have a positive polarity. The reason being can be assumed that when titles are being written, it is easier to make up an unsupported title with a negative connotation as they can be a bit more opinionated.

As for titles of trusted news articles, only 38% carries a negative connotation with neutral being at 36%, only showing a 2% difference. Compared to the lowest 21% neutral polarity for the false titles, it shows an increase as proven news tends to be more neutral than false news as it shouldn’t be too overly leaned to one specific side since it’s basically a summarization of what that article will hold.

isTrue	0	1	0_percentage	1_percentage
text_sentiment				
negative	11847	9276	0.5185	0.4331
neutral	465	474	0.0204	0.0221
positive	10538	11666	0.4612	0.5447

Table 3: Counts and percentage of text sentiment for fake/true news

Table 3 shows the results for the text token. The results we discover from these sentiments are quite interesting. Unlike the title token, the text token shows a transparent difference between the untruthful and truthful news articles.

For both types of news, the neutral remains quite similar to each other since the text is a longer body of articles that should hold true facts and what we can infer from them leading to real news text to have positive sentiments prevalent. The graph also shows that when true, there are more than half of positive connotations included as newspaper companies would want to carry more positive sentiments in their paper as portrayals of pessimistic thoughts would keep readers away.

However, for untruthful news, because it has been written to evoke the reader with irritating and provocative thoughts, about 52% of their articles would be made up of negative sentiments.

4 Methodology

In this section, we introduce and discuss various models we believe are best suited in order to accomplish our goal of achieving high accuracy in predicting fake news. In particular, we build classifiers with different sets of features using our dataset, namely, title sentiments, text sentiments, vectorized titles, and vectorized texts.

4.1 Sentiments

We first build a logit classifier using the sentiment feature created from the titles. This sentiment feature consists of three categories of *positive, neutral,* and *negative*, and *negative* is used as a reference in this model. Please see table 4 for the odds ratio estimates from this model.

	index	OR	OR%	2.5%	97.5%	p.values	
1	title_sentiment[T.neutral]	2.364	136.4%	2.246	2.487	< .001	***
2	title_sentiment[T.positive]	1.394	39.4%	1.324	1.468	< .001	***

Table 4: Odds ratio table for title sentiment

These estimates are odds ratios; the odds for news with a negative title sentiment to be real news is set to be 1.0, and the effects of news having neutral or positive title sentiment are estimated as the following: 2.364, 1.394. This means that if the news title's sentiment is neutral, the odds of such news being true is higher by a factor of 2.364. Likewise, if the news title's sentiment is positive, the odds of such news being true is higher by a factor of 1.394. This in other words indicates that the news with negative title's sentiment has a lowest probability of being true. Also, the p-values for each of the odds ratios are smaller than 0.001, which indicates the statistical significance of these odds ratios. However, the adjusted Pseudo R-squared from this model is 0.023, which is relatively low.

The second model we run is also a logistic classifier, but now with the sentiment feature created from the texts. This feature also consists of three categories of *positive*, *neural*, and *negative* with *negative* being the baseline.

	index	OR	OR%	2.5%	97.5%	p.values	
1	text_sentiment[T.neutral]	1.287	28.7%	1.112	1.489	< .001	***
2	text_sentiment[T.positive]	1.427	42.7%	1.368	1.489	< .001	***

Table 5: Odds ratio table for text sentiment

The results of the odds ratios from Table 5 are similar in this model in that news with negative text's sentiment has a lowest probability of being true. However, as opposed to the result from using the title's sentiment, the positive text sentiment has a stronger effect (i.e., 1.427) in raising the probability of news being true than the neutral text sentiment (i.e., 1.287). The p-values are significantly low for both of the odds ratios estimates; however, the adjusted pseudo R-squared from this model is significantly low as it is 0.005.

The next model we build is a logit classifier using the sentiment features constructed from both titles and texts. The negative sentiment is used as the baseline for both the title sentiment variable and text sentiment variable, and the estimated odds ratios are as the followings in Table 6:

	index	OR	OR%	2.5%	97.5%	p.values	
1	title_sentiment[T.neutral]	2.237	123.7%	2.123	2.357	< .001	***
2	title_sentiment[T.positive]	1.296	29.6%	1.228	1.367	< .001	***
3	text_sentiment[T.neutral]	1.056	5.6%	0.910	1.226	0.471	
4	text_sentiment[T.positive]	1.246	24.6%	1.191	1.303	< .001	***

Table 6: Odds ratio table for title/text sentiments

The result indicates that the title sentiments (i.e., neutral and positive) have greater effect in raising the odds of being real news than text title sentiment, since the odds ratios estimates for the title sentiments have higher values. However, the p-value of the odds ratio for the neutral text sentiment is 0.471, which indicates that this odds ratio is not statistically significant. Moreover, contrary to the expectation of adding more variables to a model, this model only achieves the adjusted R-squared of 0.025. Since the text and title sentiments do not jointly possess high predicting power and contain a statistically insignificant odds ratio estimate, we decide not to use these features for further model constructions.

4.2 Title

The next set of the models are built using the title feature. Before we dive into the construction of classifiers, we first transform the title feature into a vector form so that it can be used in the model construction. In this process, we create a 100-dimensional Word2Vec embedding model with the skip-gram method. We then use the embedding weight vector to get embedding vectors for the titles. In the logistic regression, we utilize this vectorized form of the title feature and train the model using 5-fold cross validation. In order to check for the risk of overfitting, we make predictions on both the training dataset and test dataset. From here and on, we will be using accuracy as a metric to gauge the performance of the models as our data is balanced. The accuracy of the model on the training set is 0.9982 while the one on the test set is 0.9979. In other words, the model has a fairly similar performance on both of the data sets, which indicates that the model is not overfitted.

4.3 Text

While we observed a high predicting power of the title feature, we also want to experiment how the text feature in our dataset might contribute to our model. Thus, we perform the same procedures of word embedding on the text feature and use this vectorized form of the text feature in the model construction. We specifically want to look at the pure performance of this feature, so we construct a logistic classifier that contains this feature as the only explanatory variable. Similar to the previous model with the title feature, we train a logistic classifier using the 5-fold cross validation and make predictions on both the training set and test set. As a result, the accuracy of this model on the training set is 0.9908 while the one on the test set is 0.9895. While the model does perform better on the training set, the difference is negligible; the model did not experience the overfitting issue.

4.4 Title and Text

4.4.1 Logistic Classifier CV Classifier

Since our goal is to achieve a model with the highest accuracy in predicting fake news, there is no question in incorporating both of the title and text features into the model based on the high accuracy they yielded in the previous steps. In order to create a representation of these two features combined, we concatenate the vector representations from the title and text features and create 200-dimension vectors. We then build a logistic classifier and train the classifier using the 5-fold cross validation. The result on the accuracy on the training set and test set are respectively 1.0 and 0.9997. These values prove that there is no overfitting issue and that this combined feature has a significant predicting power.

4.4.2 Decision Tree Classifier

Although the accuracy we achieved using the logistic classifier was quite satisfactory, we also want to try out different types of models for comparison. Since we have not used any tree - based model in this project, we will try and see how the decision tree classifier would perform. Also, we continue to use the same combined feature of title and text for this model as the competency of the feature was well demonstrated in the previous steps. Please see the chart below for its performance:

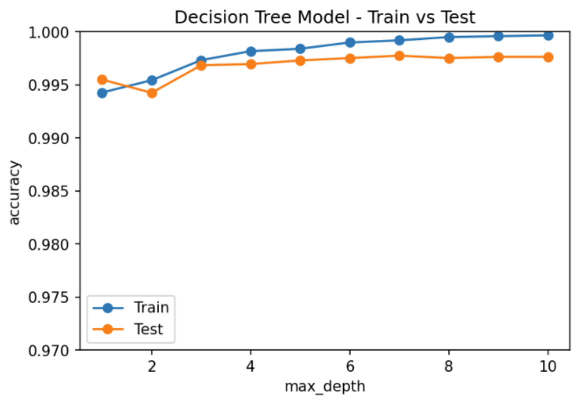


Figure 4: Decision tree model using different depth

The result from Figure 4 indicates that the classifier performs fairly well for all levels of depths, while we do observe the divergence between the performance of the model on the training set and the test set. While the divergence generally implies overfitting, the largest difference we see is merely 0.003. Therefore, we choose to pick the best model based on the highest accuracy on the test set, which is the model with depth 8. This model has the accuracy measure of 0.9975.

4.4.3 Neural Network

Another model we experiment with is the neural network model. In particular, we choose to use a multilayer perceptron classifier with the activation function of *tanh* and *adam* as the solver. To achieve the best performance, we experiment the models with different combinations of nodes and layers of the following: (1,), (2,), (3,), (1, 1), (2, 2), (3, 3). Please see below for the performance results.

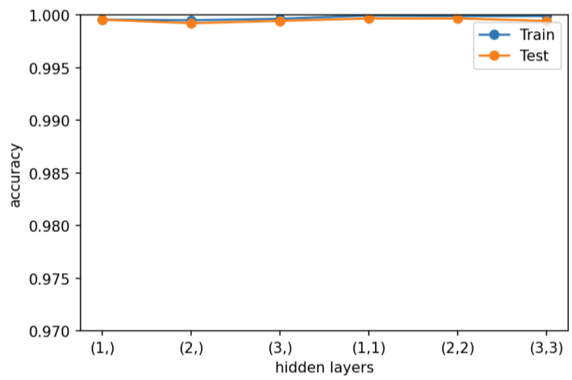


Figure 5: Neural Network Model with different hidden layers

As you can observe from Figure 5, there is no to little difference in the performance of the models on the training set and test set, meaning that the overfitting is not an issue in any of the models.

Based on this result, we choose to pick the best combination of the nodes and layers solely on the accuracy. Among the models of different layers and nodes, the one that leads to the highest accuracy is the one with 2 layers and 2 nodes. Therefore, we re-fit the model using this combination of nodes and layers (i.e., (2,2)), and the model leads us to the accuracy of 0.9997.

4.4.4 Logistic Classifier (CBOW)

The models previously trained are all based on the skip-gram word embedding model. However, we also want to check if the performance differs by using another word embedding technique. Here we introduce the Continuous Bag of Words Model (CBOW), which is a technique of predicting the middle word by combining the distributed representations of context.

Similar to the process done previously, we create a 100-dimensional Word2vec embedding model with the CBOW training algorithm for title and text each, and then get embedding vectors for both title and text. Here we choose only the logistic regression CV as our training method and fitted on title and text, since it has the highest accuracy using the skip-gram method and we want to compare with it using a different word2vec technique. The result on the accuracy on the training set and test set are respectively 0.9998 and 0.9995. These values prove that there is no overfitting issue and it also has a strong predicting power.

5 Model Comparison

After creating several models based on the ISOT dataset and testing the performance of each model, we introduce another dataset from Kaggle to test if our model also works well on a different dataset. The Kaggle dataset contains 6335 rows with a balanced data of 3164 fake news and 3171 true news; the column includes both title and text, which will be our two features used for testing, and "label" that stores "FAKE" and "REAL" as string values. We perform exploratory data analysis, where we explore some empty values in the text column. Dropping these problematic rows, we get a total of 6298 rows that will later on be included in the performance testing. Similar to the process done previously in modeling, we also do pre-processing and create a word2vec embedding model for title and text each, so that we can get the vectorized title and text.

Here we consider the two best models to test on the Kaggle dataset. The first model is the logistic regression cv classifier fitted on both title and text features, using the skip-gram embedding method. We get an accuracy of 0.9374, which is not bad, but also not as good as the ISOT dataset. For the second model, we use the neural network MLP model instead, and we get a slightly lower accuracy of 0.9346.

Table 7 shows a general comparison among all models and test results. In conclusion, the best model will be the skip-gram embedding method using a logistic regression cv classifier fitted on both title and text, since the accuracy is the highest in both ISOT and Kaggle dataset. The neural network MLP model can be another option that we consider in order to get a high accuracy on identifying fake news.

	Dataset	Embedding Method	Classifier	Features	Accuracy
0	ISOT	skip-gram	Logistic Regression CV	title	0.997854
1	ISOT	skip-gram	Logistic Regression CV	text	0.989496
2	ISOT	skip-gram	Logistic Regression CV	title,text	0.999661
3	ISOT	skip-gram	Decision Tree	title,text	0.997515
4	ISOT	skip-gram	Neural Network MLP	title,text	0.999661
5	ISOT	CBOW	Logistic Regression CV	title,text	0.999548
6	Kaggle	skip-gram	Logistic Regression CV	title,text	0.937440
7	Kaggle	skip-gram	Neural Network MLP	title,text	0.934582

Table 7: Model Comparison

6 Conclusion and Future Work

In this paper, we build machine-learning and deep-learning models upon the 2 separate ISOT datasets with over 20,000 rows each with articles from 2016 to 2017, and then compare the performance using another dataset from Kaggle which was released in around 2019 with over 6,000 rows of articles. Among all models, the logistic regression CV classifier fitted on title and text with the skip-gram embedding method creates the highest accuracy of 99.97% on the ISOT dataset and 93.74% on the Kaggle dataset.

However, the performance from the Kaggle dataset generally has a significant lower accuracy than that from the ISOT dataset. One possible reason is that the news gathered from Kaggle dataset were two to three years away from the articles gathered on ISOT dataset; therefore, the contents and the trends of the news may be different between the years, leading to a possible higher estimation error based on an outdated model.

Future work can focus on collecting more data including both fake and true news, and at the same time refreshing the outdated news from a few years ago to keep our training dataset updated so that we'll be able to identify fake news with high accuracy. What's more, it'll even be better if we can create an real-time application or an interface for users to detect any fake news before reading a news article on social media. In the end, even if we cannot stop the fake news from showing up on social media, we can indeed prevent readers from getting access to it.

References

[1] Ahmed H, Traore I, Saad S. January/February 2018. *"Detecting opinion spams and fake news using text classification"*, Journal of Security and Privacy, Volume 1, Issue 1, WileyHassan Amin. *Fake News. Balanced dataset for fake news analysis*. [Online]. https://www.kaggle.com/hassanamin/textdb3/version/1.

[2] Ahmed H, Traore I, Saad S. 2017. *Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques.* In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in

[3] David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In Advances in neural information processing systems. 1953–1961.http://papers.nips.cc/paper/4396-iterative-learning-for-reliable-crowdsourcing-systems

[4] Del Vicario, M., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., ... & Quattrociocchi, W. (2016). The spreading of misinformation online. Proceedings of the National Academy of Sciences, 113(3), 554-559

[5] GeeksforGeeks. 2020. Removing stop words with NLTK in Python. [Online]

https://www.geeksforgeeks.org/removing-stop-words-nltk-python/

[6] Hassan Amin. Fake News. Balanced dataset for fake news analysis. [Online].

https://www.kaggle.com/hassanamin/textdb3/version/1.

[7] J.Cement. 2020. Number of social media users 2025. *Statista*.

[8] Lazer, D, Baum, MA, Benkler, Y. 2018. The science of fake news. Science 359(6380): 1094–1096.

[9] Lazer D, Baum M, Grinberg N, et al. 2017. *Combating fake news: An agenda for research and action.* Harvard Kennedy School, Shorenstein Center on Media, Politics and Public Policy

[10] Parth Patwa, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, Tanmoy Chakraborty. 2020. *Fighting an Infodemic: COVID-19 Fake News Dataset*. arXiv:2011.03327.

[11] Prez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. 2018. Automatic Detection of Fake News. In Proceedings of the International Conference on Computational Linguistics, (p. 33913401)

[12] Qiang Liu, Jian Peng, and Alexander T. Ihler. 2012. Variational inference for crowdsourcing. In Advances in neural information processing systems. 692–700.

http://papers.nips.cc/paper/4627-variational-inference-for-crowdsourcing

[13] Sarkar, D. (n.d.). *Emotion and Sentiment Analysis: A Practitioner's Guide to NLP*. KDnuggets. [Online]

https://www.kdnuggets.com/2018/08/emotion-sentiment-analysis-practitioners-guide-nlp-5.html#:%7E:text=The%20key%20aspect%20of%20sentiment,sign%20of%20the%20polarity%20score

[14] Tacchini, E., Ballarin, G., Vedova, M. L. D., Moret, S., & de Alfaro, L. 2017. Some Like It Hoax: Automated Fake News Detection in Social Networks. arXiv preprint arXiv:1704.07506.

[15] Torabi Asr, F., & Taboada, M. (2019). Big Data and quality data for fake news and misinformation detection. Big Data & Society.

https://doi.org/10.1177/2053951719843310

[16] University of Victoria. Fake News Detection Datasets. [Online].

https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php

[17] Wang, W. Y. 2017. " 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 (p. 422426).