Fake News Detection Using Content-Based Features and Machine Learning

Okuhle Ngada
School of IT
Nelson Mandela University
Port Elizabeth, South Africa
okuhle.ngada@outlook.com

Bertram Haskins
School of IT
Nelson Mandela University
Port Elizabeth, South Africa
bertram.haskins@mandela.ac.za

2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE) | 978-1-6654-1974-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/CSDE50874.2020.9411638

***Abstract*—The problem of fake news is a complex problem and is accompanied with social and economic ramifications. Targeted individuals and entities may lose trustworthiness, credibility and ultimately, suffer from reputation damages to their brand. Economically, an individual or brand may see fluctuations in revenue streams. In addition, the complex nature of the human language makes the problem of fake news a complex problem to solve for currently available computational remedies. The fight against the spread of fake news is a multi-disciplinary effort that will require research, collaboration and rapid development of tools and paradigms aimed at understanding and combating false information dissemination. This study explores fake news detection techniques using machine learning technology. Using a feature set which captures article structure, readability, and the similarity between the title and body, we show such features can deliver promising results. In the experiment, we select 6 machine learning algorithms, namely, AdaBoost as AB, Decision Tree as DT, K-Nearest Neighbour as KNN, Random Forest as RF, Support Vector Machine as SVM and XGBoost as XGB. To quantify a classifier's performance, we use the confusion matrix model and other performance metrics. Given the structure of the experiment, we show the Support Vector Machine classifier provided the best overall results.**

***Index Terms*—Machine Learning, Natural Language Processing, Text analysis**

I. INTRODUCTION

The misinformation resulting from fake news can lead to many consequences, such as confusion and loss of trust in a firm. It is difficult for readers to differentiate fake news from real news, given the fact it is easy to create professional, profitable websites with the intent of distributing false information. Furthermore, fake news stories spread rapidly on social media due to a platform's lack of fact verification and the social media user's inability to identify fake stories before sharing. A user's opinion regarding an entity, be it a person, brand, or organisation may change after reading a fake story, which in turn, harms the entity at hand. On the 9th of January 2018, a website named AllAfricaNews fabricated a story that the South African President, Jacob Zuma, had resigned. According to [1], this resulted in a positive response to the Rand's value for moments until other news firms reported the story as fake news. Although many fake news initiatives exist, these have been criticised for showing bias, or favouring the originator's beliefs and ideals. Research in fake news detection is relatively new, despite fake news being around for many years.

The objective of this study is to determine the applicability of various machine learning techniques to the task of identifying fake news. To address this objective, the remainder of the study is structured as follows; Section II provides a theoretical grounding for the study. The experimental design is discussed in Section III and the results of the experiments are presented in Section IV. Section V provides an overview of what was learnt from the process of experimentation and the study is concluded in Section VI.

II. BACKGROUND

This section provides an overview of the issues related to fake news, as well as the role machine learning can play in its detection.

A. The Global Fake News Epidemic

According to [2], Facebook has 2.2 billion users, whilst Twitter has 330 million users. In a South African context, the country accounts for 16 million Facebook users, and 8 million Twitter users [3]. The figures mentioned highlight that social media presents a large user base, potentially prone to the dissemination of false online news stories. A recent study conducted by [4] shows news firms play a significant role in providing content on social media; 75% of the 9.7 million tweets sampled, had a link to at least one news firm. Furthermore, 42% of the most linked-to sites are news firms [4]. Another study carried out after the 2016 US Election indicates that 64% of American adults believe fake news causes confusion and 23% of Americans admitted to sharing fake news; intentionally or unintentionally [5]. Using the facts presented, it is evident that news organisations have influence and play a vital role in distributing information. Confusion amongst individuals is one of many effects emanating from the propagation of fake news. Technology has made it simpler to create professional and profitable fake news websites, which draw revenue from advertising [6]. In a fake news context, this results in many professionally built fake news websites, with the intent of spreading false information, and possibly generating revenue through various advertising networks. During the 2016 US Elections, Veils, a small city in Macedonia, rose to fame for the wrong reasons – teenagers in this city authored 100 pro-Trump fake news websites. Boris, one of many teenagers behind several fake news websites, earned \$16,000 dollars in revenue from advertising engines such as Google AdSense [7]. According to [8], two possible motives for authoring fake news could be: advertising revenue resulting from stories that go viral on social media and to favour ideals the author believes in.

South Africa is no stranger to the effects of fake news. Political debates, allegations of state capture made in a report published by former public protector Thuli Madonsela [9] and leaked emails exposing corrupt dealings between the infamous Gupta Family and Jacob Zuma have changed views and ideals of many South Africans. Wasserman [9] found that the Gupta Family used services from UK-based PR firm, Bell Pottinger to spread false narrative that Jacob Zuma, and his supporters, are victims of 'white monopoly capital'. Several Twitter accounts were created to counter allegations made against Jacob Zuma and the Gupta Family [9]. After analysing the source of related tweets, it was found that the tweets originated from media houses owned by the Gupta Family, Africa News Network 7 (ANN7) and The New Age, and from a rival political party, Black First Land First (BLF) [9]. The effects related to this instance of false information could be seen through heightened racial tensions in the country, as well as distrust in the government. It is evident that high-profile organisations, such as Bell Pottinger, can be manipulated into spreading false narrative, only benefiting the originator. The consumer, namely, the news reader, is the one that pays the ultimate price with confusion and distrust being a few of the many consequences.

B. *Current Efforts in Combatting Fake News*

Presently, we have seen the rise of many fact-verification websites such as Snopes, PolitiFact and FactCheck. These fact-verification initiatives have teams of journalists and writers, who assess the credibility of various stories circulating on the internet, as well as claims made by politicians. [10] noted that despite the establishment of many fact-checking outlets, they vary in thoroughness. Many fact-checking websites exist; however, these websites have been criticised for bias [10]. Zimdar [11], affiliated to Merrimack College, compiled a list of fake news websites, split into 11 categories. Zimdar [11] also provides guidelines in analysing news sources and websites. [10] criticised the list, stating the labelling system used is only effective for individuals who share her political and ideological views. In a South African context, users of the MyBroadband Forums have resorted to compiling and updating a list of well-known fake news websites, where forum users suggest new fake news websites. Changes to the list of fake news websites are approved by a forum moderator [12].

C. *Machine Learning and Automated Fake News Detection*

Over the years, researchers in the fake news community have demonstrated how machine learning technology can assist in fake news identification. Today, various data sets have been compiled for the application of fake news identification. [13] were first in releasing a fake news data set, consisting of short statements. This was later succeeded by [14], who released a larger data set, consisting of short statements. Many other publicly available datasets, for application of fake news identification exist. [15] compared the performance of various machine learning models using a dataset, consisting of false and real news from the OpenSources [16] and articles from the SignalMedia dataset [17]. Both websites maintain a dataset, for application of automated fake news detection research [15]. [18], built ClaimBuster – an ongoing project which uses Natural Language Processing techniques and supervised learning to verify politically driven statements. The system uses a labelled dataset. Each item in the corpus is given one of three labels, namely, a Non-Factual Sentence (NFS), Unimportant Factual Sentence (UFS) and Check-Worthy Factual Sentence (CFS). The use of ClaimBuster can assist organisations in identifying false claims from competitors and assist professionals in verifying documents [18]. [19] conducted an experiment where several machine learning classification techniques were evaluated in the identification of fake news. One of many highlights were the high accuracy levels of various machine learning models tested [19].
With most machine learning frameworks, machine learning models provide several hyper-parameters which may affect the performance of a given classifier. Given the complexity of such parameters, there exist methodologies which assist in selecting optimal parameters for a given classifier. Hyperparameter selection is the process of selecting the best combination of parameters which yield optimal performance for a given classifier. The SciKit Learn library provides two hyper-parameter searching techniques, namely, GridSearchCV [20] and RandomizedSearchCV [21]. [22] use a defined set of hyper parameters, which include the number of estimators to use for the Random Forest, AdaBoost and bagging classifiers. [14] implements the grid search technique to optimise parameters for the Logistic Regression and Support Vector Machine models.

In terms of factors that could suggest an article being fake news, several pointers have been noted by various authors. [23] noted two indicators that could be used in discerning an article for being the truth or falsified. The authors describe headline/body dissonance, where the title and body of a news article do not correlate. The title could present itself as a factual article, when the body is built, using unverified facts [23]. The authors also add, questioning headlines give an impression that an article is truthful when that may not be the case [23]. [24] hypothesise that authors of false reviews typically write content that contain contradictions or the content written fails to mention some aspects which other truthful articles mention.

III. Experimental design

The aim of the experiment described in this study is to validate the effectiveness of automated fake news detection, using natural language processing techniques and machine learning. The problem of automated fake news detection is not simple, given the complexity of human dialect. We select the dataset authored by [25], and 6 machine learning algorithms for the task of fake news detection. In the experiment, articles fall into one of two classes; namely fake or reliable. Following basic pre-processing techniques, we extract features which describe the contents of articles and structure of articles. Finally, we use the Confusion Matrix model, together with other performance metrics which draw on the results obtained from the confusion matrix; namely, accuracy, F1-Score, recall, precision, and Receiver Operating Characteristics (ROC) score. To validate the credibility of the results in table III, we use 10-fold cross validation on each of the selected algorithms against the [25] dataset. We use the python SciKit-Learn Library [26] for the machine learning, and the Python NLTK package [27] for the pre-processing.

A. Data sets and Tooling

For the experiment, we use the fake and real news dataset authored by [25]. The Kaggle hosted dataset consists of two datasets; one fake news dataset consisting of 23 481 fake news articles, and a real news dataset containing 21 417 real news articles. The datasets contain articles published between the years 2015 and 2018. To train the Doc2Vec model, a subset of the FakeNewsCorpus, authored by [16] is selected. Based on the Word2Vec model, [28] introduce the Doc2Vec model, which entails two implementations, namely, the Paragraph Vector – Distributed Memory (PV-DM) and Paragraph Vector – Distributed Bag of Words (PV-DBOW). In the Doc2Vec architecture, each paragraph is assigned a unique vector and each word in the paragraph is assigned a vector. The paragraph vector and word vectors are combined to predict the next word in a given context [28]. Due to limited computational power, the first 12 000 online news articles of each news article category contained in [16] dataset are selected. The resultant news dataset contains 120 000 news articles. Rows which contain missing data, specifically, missing article bodies, were dropped. The fake and real news datasets are merged and randomly shuffled before a machine learning algorithm is trained on the data. Figure 1 represents a word cloud for fake news articles, whilst Figure 2 represents a word cloud for real news articles as found in the dataset. The tag cloud selects the top 3000 words from the fake and real news vocabularies. SciKit's TfIdfVectorizer [29] was employed in computing a TF-IDF score to measure the relevance of words in both fake and real news datasets.

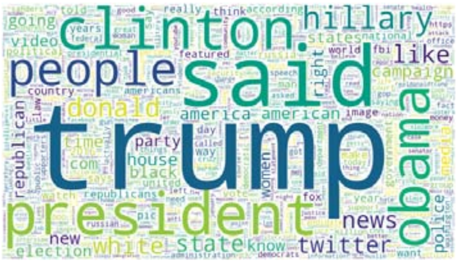

Fig. 1. Word cloud for the top 3000 words contained in [25] fake news dataset, using the Scikit Learn's TfIdfVectorizer [29] model to measure word relevance.

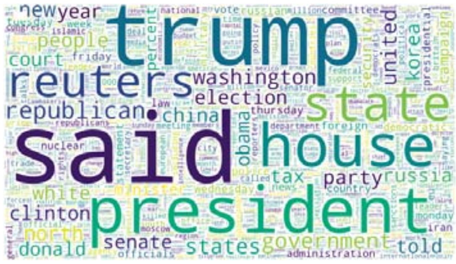

Fig. 2. Word cloud for the top 3000 words contained in [25] real news dataset, using SciKit Learn's TfIdfVectorizer [29] to measure word relevance

B. Pre-processing

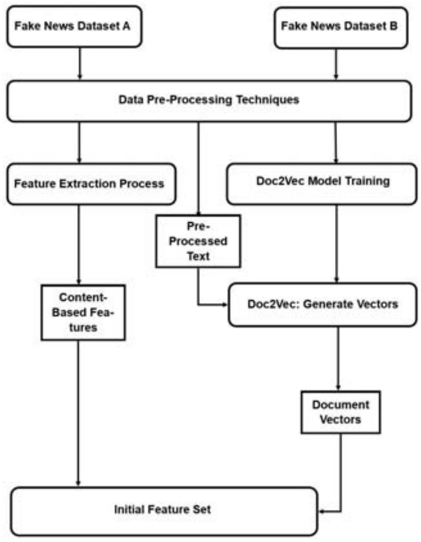

Fig. 3. Data cleaning and feature extraction flow employed in the study

Figure 3 illustrates the data pre-processing strategy undertaken in this study. [25] is represented as Fake news dataset A, and [16] is represented as Fake News Dataset B. The FakeNewsCorpus [16] dataset is selected to train the Doc2Vec model. We employ stop word removal and special character removal as part of the data pre-processing techniques for datasets [25] and [16]. Stop word removal is the process of removing common words that do not add value to the sentence.

C. Feature Extraction

1) Content Based Features: Following pre-processing techniques, we use the work of [30] as a reference for possible features that best describe the contents of an article. We consider total word counts on the article title and body, total count of punctuation marks in the article title and body, a count on parts of speech tags (POS) present in articles and average sentence lengths in article contents as features. Other features considered include the Type-Token Ratio, which considers the total unique words in a document versus the total number of words in a document. The Type Token ratio is calculated using the following equation [31]:

$$tt\_ratio = \frac{Nt}{Nw} \tag{1}$$

Where:

Nt: Number of unique words (terms) per document
Nw: Number of words per document

We also compute the cosine similarity, where an aggregated cosine similarity score between a given article title and every sentence contained in the article body is calculated. The decision to include cosine similarity as a metric is supported by the work of [30], who include the metric for calculating headline and article body similarity. If A and B are vectors, the cosine similarity calculation can be expressed as [32]:

$$cos\_sim = sim(A, B) = \frac{A \cdot B}{||A||||B||'} \tag{2}$$

In addition, we include three text readability metrics as features which examine the readability of articles by examining the education level required to comprehend the text. We select the Gunning Fog Index [33], Automated Readability Index [33], and Flesh Kincaid Grade Level Index [33] as readability metrics. The decision to include such metrics in the feature set are based on the work of [34], who explore fake news detection by using metrics which describe the readability of text. We use the py-readability-metrics [35] package to compute readability metrics. Table I illustrates all content-based features selected for the study.

2) Doc2vec - Document Vectors: To transform raw text into numeric values, we include document vectors for articles contained in the dataset using the Doc2Vec model. We train a Doc2vec model using a subset of online news articles collected by [16]. The parameter configuration used for the model is as follows: the window size parameter is set to 8, and the selected training model is Paragraph Vector-Distributed Bag of Words (PV-DBOW). The decision to use such configuration is inspired by [28] sentiment analysis experiment. In the experiment, the selected IMDB dataset consists of 100 000 movie reviews. The authors configure the window-size parameter, and vector dimensions parameter to 8 and 400 respectively [28].

D. Initial Feature Set

Using the fake and real news data sets authored by [25], Table I represents 53 features extracted from the corpora; such features describe sentence composition and structure. The vectors generated by the Doc2Vec model are added to the feature set, thus resulting in each sample containing a feature set of 453 vectors. From table I, py-readability-metrics [35] calculates features text_ari, text_gf, and text_fkg using equations 3, 4, and 5:

$$text\_ari = 4.71 \cdot LPW + 0.5 \cdot Ws - 21.43 \tag{3}$$

$$text\_gf = 0.4 \cdot (Ws + 100 \cdot Ps) \tag{4}$$

$$text\_fkg = (0.38 \cdot Aws + 11.8 \cdot As) - 15.59 \tag{5}$$

Where:

LPW: Number of letters per word
Ws: Number of words per sentence
Ps: Poly syllables per word
Aws: Average words per sentence
As: Average syllables per word

TABLE I
INITAL FEATURE SET

Calculated Features	Name
text_ari	Automated Readability Index
text_gf	Gunning Fog Index
text_fkg	Flesch Kincaid Grade Level Index
tt_ratio	Type-Token ratio for an article's body
cos_sim	Cosine Similarity score for an article's title and body
Accumulative Features	
Count of Unique Words, Average Sentence Length, Title Word Count, Article Body Word Count, POS Tag for 'Adjective', POS Tag For 'Noun', POS Tag For ',', POS Tag for 'Proper Noun', POS Tag For 'Cardinal Digit', POS Tag For 'Verb - Present Participle', POS Tag For '.', POS Tag For 'Determiner', POS Tag For 'Verb - Past Participle', POS Tag for '(', POS Tag For 'Preposition', POS Tag for ')', POS Tag for 'Averb', POS Tag for 'Verb - Single, POS Tag for 'Personal Pronoun', POS Tag for ':', POS Tag for 'Adjective', POS Tag for 'Verb', POS Tag for Possessive pronoun; POS Tag for Modal; POS Tag for ' " ', POS Tag for '3rd person verb', POS tag for 'Proper Noun', POS Tag for 'interjection', POS Tag for 'Adverb', POS tag for 'Proper Noun', POS Tag for 'coordinating conjuction', POS Tag for 'Pronoun', POS tag for 'Possessive pronoun', POS tag for 'Wh-determiner', POS Tag for 'Existential', POS Tag for 'Adjective', POS Tag for 'Adverb - Comparitive', POS Tag for 'Particle', POS Tag for 'Symbol', POS Tag for 'to', POS Tag for 'Foreign word', POS Tag for 'Adverb - superlative', POS Tag for 'Possessive wh-pronoun', POS Tag for 'Predeterminer', POS tag for 'List item marker'	

E. Machine Learning Algorithms

We select 6 machine learning algorithms as illustrated in table II. K-Nearest Neighbour is denoted as KNN, Support Vector Machine as SVM, Random Forest as RF, Decision Tree as DT XGBoost as XGB, and AdaBoost as AB. The feature matrix generated from processing the mentioned dataset is fed into the mentioned machine learning algorithms. We split the dataset into train-test split of 80% and 20% of articles respectively.

1) Machine Learning Classifier Configuration: The selected machine learning configuration is highlighted in II. Using the GridSearchCV hyper parameter selection process, we are able to select optimal configuration for the 6 machine learning algorithms selected in this experiment. The choice of parameters to experiment with were based on parameter selections other authors in the fake news research community have employed. Table II illustrates parameter sets used to determine the best classifier configurations. Due to the high computational cost associated with finding the best set of hyper parameters, grid search is performed over a 3-fold cross validation configuration.

IV. Results

The confusion matrix provides metrics which can be used in calculating a given model's performance. Such metrics are accuracy, precision, recall and F-measure. The test dataset, (20% of articles), is used. Data normalisation and dimensionality reduction techniques were not considered. Table

TABLE II
PARAMETERS EXPLORED IN THE HYPER PARAMETER SELECTION PROCESS

Classifier	Configuration	Best Config.	Accuracy
KNN	n_neighbors: [2, 3, 4, 5, 6, 7, 8, 9, 10], 'algorithm': ['ball_tree', 'kd_tree', 'brute']	n_neighbors: 5, algorithm: ball_tree	90.8%
SVM	'kernel': ['linear', 'rbf'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000],	kernel: linear C: 0.1	99.4%
RF	'n_estimators': [2, 5, 7, 10], 'max_depth': [0, 1, 3, 5, 7]	'n_estimators:': 10, 'max_depth': 7	97.1%
XGB	'n_estimators': [7, 10, 50, 100], 'max_depth': [1, 3, 5, 7]	'n_estimators:': 100, 'max_depth': 5	98.8%
AB	'n_estimators': [20, 50, 100, 150], 'algorithm': ['SAMME', 'SAMME.R']	'algorithm': 'SAMME.R', 'n_estimators': 150	99.2%

III summarises results obtained with each classifier and the selected hyper-parameter configurations. Table IV summarises the confusion matrix results obtained at each classifier. The confusion matrix provides 4 metrics that can assess a given classifier's performance: True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). Finally, we include results obtained through the use of a 10-fold cross validation technique on the selected machine learning algorithms in Table V. In table V, we denote AdaBoost as AB, Decision Tree as DT, K-Nearest Neighbour as KNN, Random Forest as RN, Support Vector Machine as SVM, and XGBoost as XGB.

TABLE III
PRECISION, ACCURACY, RECALL, F-MEASURE AND ROC PERFORMANCE RESULTS

Classifier	Precision	Accuracy	Recall	F-Measure	ROC
SVM	99.3%	99.4%	99.4%	99.4%	99.3%
DT	95.0%	94.9%	95.3%	95.2%	94.8%
RF	99.4%	98.9%	98.4%	98.9%	99.9%
KNN	96.4%	91.2%	86.4%	91.2%	91.5%
XGB	99.1%	98.8%	98.6%	98.8%	98.8%
AB	99.3%	99.3%	99.4%	99.3%	99.3%

TABLE IV
CONFUSION MATRIX RESULTS BY CLASSIFIER

Classifier	TN	FP	FN	TP
SVM	4248 (47.3%)	29 (0.3%)	27 (0.3%)	4676 (52.1%)
KNN	4115 (45.8%)	151 (1.7%)	639 (7.1%)	4075 (45.4%)
RF	4268 (47.5%)	41 (0.5%)	49 (0.5%)	4622 (51.5%)
DT	4028 (44.9%)	233 (2.6%)	224 (2.5%)	4495 (50.1%)
AB	4300 (47.9%)	30 (0.3%)	36 (0.4%)	4614 (51.4%)
XGB	4269 (47.5%)	44 (0.5%)	66 (0.7%)	4601 (51.2%)

TABLE V
10-FOLD CROSS VALIDATION RESULTS EXPRESSED AS A PERCENTAGE

	1	2	3	4	5	6	7	8	9	10
SVM	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.2	99.6
DT	95.3	95.1	95.3	95.5	95.6	95.2	94,9	94.4	95.2	94.7
RF	98.8	98.9	98.5	99.0	98.9	98.8	98,8	98.8	98.7	98.9
KNN	91.1	91.3	91.4	90.6	92.0	91.4	91,9	91.6	91.1	91.1
AB	99.0	99.2	99.3	99.1	98.9	99.1	99,2	99.1	99.3	99.2
XGB	99.0	98.8	98.9	98.9	98.5	99.1	98.7	98.9	99.0	98.7

V. DISCUSSION

The results presented in section III show promising results are attainable using features that describe the contents and structure of the articles together with document vectors to represent each document. We chose the doc2vec model to represent each document based on observations noted by several authors regarding the use of simple, traditional methods of representing documents (bag of words, TF-IDF, etc); such methods do not retain the meaning and relationship of words. Considering precision, the RF classifier obtained the best result, at 99,4%. When accuracy is considered, SVM achieved the best result at 99,4%. For recall, SVM achieved the best result at 99,4%. For F-Measure, SVM achieved the best result at 99,4%. For the Receiving Operating Characteristics (ROC), the RF classifier achieved the best result at 99.9%. When aggregating all the metrics in Table III, we find the SVM classifier achieves the best result at 99,36%. Upon examination of results reported in Table IV, we find the SVM classifier yields minimal error on falsely classifying articles, at 0.3% (or 27 articles) for FN and 0.3% (or 29 articles) for FP. In addition, the SVM classifier yields the highest, correct classification of fake and real articles at 47.3% (or 4676) of articles as True Positive. The AdaBoost Classifier (AB) attained the best true negative classification result at 4300 51,4% (or 4300) of articles. Finally, in table V, we show the results obtained at each fold, for each classifier, are similar to all other results. Furthermore, the results obtained in the k-fold cross validation process are not vastly different to the results obtained in Table III, thus adding credibility to results exhibited in Table III.

VI. CONCLUSION AND FUTURE WORK

The results obtained show fake news detection, using features that best describe the structure of article bodies can yield promising results. Furthermore, this work demonstrates that it is possible to build tools which could assist fact-checkers, or online news readers in flagging or highlighting potentially fake news articles. Although other research related to fake news detection include the use of features that describe the users, articles and the user interactions on a given article, this work shows that considering just the article contents can yield promising results.

Future work related to the study include the use of deep learning approaches for fake news detection. Several researchers in this field have reported promising results using deep learning techniques which, in most cases, yield better performance results compared to traditional machine learning techniques.

REFERENCES

[1] L. Kawa and C. Goko. (2018) How fake news and elon musk sent the rand haywire. Retrieved: 2018, July 13. [Online]. Available: https://www.moneyweb.co.za/news/markets/how-fake-news-and-elon-musk-sent-the-rand-haywire/

[2] J. Clement. (2018, Aug) Twitter: monthly active users worldwide. Retrieved: 2018, September 25. [Online]. Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[3] A. Goldstuck and O. Patricios. (2017, Sep) Inside the sa social media landscape report 2018 - ornico. Retrieved: 2018, March 23. [Online]. Available: https://website.ornico.co.za/2017/09/21/sa-social-media-2018/

[4] G. Stockling, M. Barthel, and E. Grieco. (2018, Jan) Sources shared on twitter: A case study on immigration. Retrieved: 2018, September 22. [Online]. Available: https://www.journalism.org/2018/01/29/sources-shared-on-twitter-a-case-study-on-immigration/

[5] J. Anderson and L. Rainie. (2017, Oct) The future of truth and misinformation online. Retrieved: 2018, March 25. [Online]. Available: http://www.pewinternet.org/2017/10/19/the-future-of-truth-and-misinformation-online/

[6] P. Fernandez, "The technology behind fake news," *Library Hi Tech News*, 2017.

[7] S. Subramanian. (2017, May) The macedonian teens who mastered fake news. Retrieved: 2018, September 26. [Online]. Available: https://www.wired.com/2017/02/veles-macedonia-fake-news/

[8] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, May 2017. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211

[9] H. Wasserman, "Fake news from africa: Panics, politics and paradigms," *Journalism*, vol. 21, no. 1, pp. 3–16, 2020, retrieved: 2018, September 22. [Online]. Available: https://doi.org/10.1177/1464884917746861

[10] O. Batchelor, "Getting out the truth: The role of libraries in the fight against fake news," *Reference Services Review*, vol. 45, 06 2017.

[11] M. Zimdar. (2016) False, misleading, clickbait-y, and satirical "news" sources. Retrieved: 2018, September 22. [Online]. Available: https://docs.google.com/document/d/10eA5-mCZLSS4MQY5QGb5ewC3VAL6pLkT53V_81ZyitM/preview

[12] MyBroadband. (2017, Mar) List of known fake news sites in south africa (and beyond). Retrieved: 2019, April 27. [Online]. Available: https://mybroadband.co.za/forum/threads/list-of-known-fake-news-sites-in-south-africa-and-beyond.879854/

[13] A. Vlachos and S. Riedel, "Fact Checking: Task definition and dataset construction," pp. 18–22, 2014, retrieved: 2018, July 2018. [Online]. Available: http://aclweb.org/anthology/W14-2508

[14] W. Y. Wang, ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," 2017, retrieved: 2018, June 25. [Online]. Available: http://arxiv.org/abs/1705.00648

[15] L. Dyson and A. Golab, "Evaluating machine learning algorithms for fake news detection," pp. 1 –15, 2017, retrieved: 2020, September 27. [Online]. Available: https://github.com/aldengolab/fake-news-detection/blob/master/fakenews_finalpaper.pdf

[16] M. Szpakowski. (2018, Feb) several27/fakenewscorpus: A dataset of millions of news articles scraped from a curated list of data sources. Retrieved: 2019, July 20. [Online]. Available: https://github.com/several27/FakeNewsCorpus

[17] D. Corney, D. Albakour, M. Martinez, and S. Moussa, "What do a million news articles look like?" in *Proceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016.*, 2016, pp. 42–47. [Online]. Available: http://ceur-ws.org/Vol-1568/paper8.pdf

[18] N. Hassan, F. Arslan, C. Li, and M. Tremayne, "Toward automated fact-checking," *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*, pp. 1803 – 1812, 2017, retrieved: 2018, July 31. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3097983.3098131

[19] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," *ISDDC: International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, vol. 10618, December 2017, retrieved: 2018, June 01. [Online]. Available: http://link.springer.com/10.1007/978-3-319-69155-8

[20] SciKit-Learn. (2020) sklearn.model_selection.gridsearchcv — scikit-learn 0.23.2 documentation. Retrieved: 2020, September 25. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[21] ——. (2020) sklearn.model_selection.randomizedsearchcv — scikit-learn 0.23.2 documentation. Retrieved: 2020, September 25. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

[22] R. Moraes Silva, R. Santos, T. Almeida, and T. Pardo, "Towards automatically filtering fake news in portuguese," *Expert Systems with Applications*, pp. 1 – 18, January 2020, retrieved: 2020, February 22.

[23] Y. Chen, N. J. Conroy, and V. L. Rubin, "News in an online world: The need for an "automatic crap detector"," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1 – 4, 2015, retrieved: 2018, July 01.

[24] V. W. Feng and G. Hirst, "Detecting deceptive opinions with profile compatibility," *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 338–346, October 2013, retrieved: 2018, August 28. [Online]. Available: http://www.aclweb.org/anthology/I13-1039

[25] C. Bisaillon. (2020, Mar) Fake and real news dataset. Retrieved: 2020, May 20. [Online]. Available: https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[26] SciKit-Learn. (2020) scikit-learn: machine learning in python — scikit-learn 0.23.2 documentation. Retrieved: 2020, September 25. [Online]. Available: https://scikit-learn.org/stable/

[27] (2020, Apr) Natural language toolkit - nltk 3.5 documentation. Retrieved: 2020, September 25. [Online]. Available: https://www.nltk.org/

[28] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1188–1196, retrieved: 2020, July 20. [Online]. Available: http://proceedings.mlr.press/v32/le14.html

[29] SciKit-Learn. sklearn.feature_extraction.text.tfidfvectorizer scikit learn 0.23.2 documentation. Retrieved: 2020, September 25. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[30] R. Masood and A. Aker, "The fake news challenge: Stance detection using traditional machine learning approaches," *IC3K 2018 - Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, vol. 3, pp. 128–135, 2018, retrieved: 2020, February 17.

[31] L. Shen, "lexicalrichness - pypi," Aug 2018. [Online]. Available: https://pypi.org/project/lexicalrichness/

[32] J. Han, M. Kamber, and J. Pei, *Getting to know your data*. Elsevier/Morgan Kaufmann, 2012, p. 77–78.

[33] D. Bilal and L.-M. Huang, "Readability and word complexity of serps snippets and web pages on children's search queries: Google vs bing," *Aslib Journal of Information Management*, vol. 71, pp. 241 – 259, 03 2019, retrieved: 2020, May 25.

[34] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," *arXiv preprint arXiv:1703.09398*, 2017, retrieved: 2020, January 29.

[35] C. DiMAscio. (2020, Aug) py-readability-metrics pypi. Retrieved: 2020, September 26. [Online]. Available: https://pypi.org/project/py-readability-metrics/