Which machine learning paradigm for fake news detection?

Dimitrios Katsaros
KIOS Center of Excellence &
University of Cyprus
Nicosia, Cyprus
katsaros.dimitrios@ucy.ac.cy

George Stavropoulos
University of Thessaly
Volos, Greece
gstavropoulos@e-ce.uth.gr

Dimitrios Papakostas
University of Thessaly
Volos, Greece
jim.papakostas@gmail.com

ABSTRACT

Fake news detection/classification is gradually becoming of paramount importance to out society in order to avoid the so-called reality vertigo, and protect in particular the less educated persons. Various machine learning techniques have been proposed to address this issue. This article presents a comprehensive performance evaluation of eight machine learning algorithms for fake news detection/classification.

CCS CONCEPTS

• **General and reference → Evaluation**; • **Human-centered computing → Collaborative and social computing design and evaluation methods**; **Social network analysis**.

KEYWORDS

Fake news, machine learning

ACM Reference Format:
Dimitrios Katsaros, George Stavropoulos, and Dimitrios Papakostas. 2019. Which machine learning paradigm for fake news detection?. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19), October 14–17, 2019, Thessaloniki, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3350546.3352552

1 INTRODUCTION

Nowadays online information grows at unprecedented rates, and gradually more and more people consult online media, e.g., the Web, Online Social Networks (OSN) such as Facebook and Twitter, for satisfying their information needs. However, not all information/knowledge producers are trustworthy, and the problem of fake news – fabricated stories presented as if they were originating from legitimate sources with an intention to deceive – and their spreading is getting more and more severe. It is speculated that by 2020, people in developed countries will encounter more fake than real news. This phenomenon is termed *reality vertigo*[1].

This problem emerged as a major issue particularly during the 2016 US Presidential election, and it is even believed that fake news affected the final outcome. Unfortunately this is not an isolated event; a study [10] shows that false medical information gets more views, likes, comments than true medical information. Even worse, fake news are not only (more) popular, but they are spreading at a faster pace [16] than real news. So, countermeasures against fake news started to develop rapidly.

1.1 Motivations and contributions

The need for detecting fake news – or classifying a news item as fake, true, or suspicious – is of paramount importance if we wish to avoid reality vertigo and protect our society, especially the less educated persons of our society. Machine learning has been proven very effective in combating spam email, which is one type of misinformation; so, algorithms belonging to this category of techniques were among the very first whose efficacy has been investigated. The following machine learning paradigms have been examined in the context of fake news detection:

- Regression
  - L1 regularized logistic regression
- Support Vector Machines (SVM)
  - C-support vector classification
- Bayesian methods
  - Gaussian naive Bayes
  - Multinomial naive Bayes
- Decision tree-based methods
  - Decision trees
  - Random forests
- Neural networks
  - Multi-layer perceptron (MLP)
  - Convolutional neural networks (CNNs)

However, their relative performance is unknown, and so is their generic behavior when tested against diverse datasets. The aim of this article is to answer these two broad questions. In this context the present article makes the following contributions:

- It contrasts the effectiveness and efficiency of the competitors for several diverse datasets, and various performance measures.
- It contrasts the speed of the competitors for these datasets.
- It introduces a public Web-based application to test the competitors against any real URL for possible fake news.

The rest of the article is organized as follows: Section 2 presents briefly the related work. Section 3 introduces the algorithms that will be evaluated. Section 4 describes the evaluation environment, i.e., competitors, datasets, performance measures, and on, and section 5 presents the actual evaluation of the competing algorithms. Finally, Section 6 concludes the article.

[1] https://www.nature.com/news/astronomers-explore-uses-for-ai-generated-images-1.21398

WI '19, October 14–17, 2019, Thessaloniki, Greece

ACM ISBN 978-1-4503-6934-3/19/10…$15.00
https://doi.org/10.1145/3350546.3352552

2 RELATED WORK

Machine learning and data mining algorithms have been considered as a very significant arsenal in the battle against fake news. Several supervised models have been proposed. For instance, a ranking model based on SVM and Pseudo-Relevance Feedback for tweet credibility has been developed in [4]. A credible news classifier based on regression was proposed in [5]. SVM on content-based features was utilized in [6] in order to detect fake, satirical and real news items. A comprehensive survey of data mining algorithms employed for fake news detection is contained in article [12].

A different line of research was taken by [1, 7] where the actual content was analyzed and news items were represented as multidimensional tensors. This is in contrast to aforementioned works which are based on feature extraction.

Some works investigated the issue of fake news detection following a credibility diffusion-based approach. These works [3] construct complex networks of heterogeneous entities (persons, tweets, events, message, etc) and study the paths of fake news propagation in order to find out non-credible sources of information, and thus infer fake news.

There are academic efforts to develop online services which will study how misinformation spreads and competes in online social networks. For instance Hoaxy[2] [10] is such a service for Twitter; it as actually a platform for the study of diffusion of misinformation in Twitter.

Less related areas are those concerning rumor classification, trust discovery, clickbait detection, spammer and bot detection, as well as related online services e.g., Botometer which checks Twitter accounts and assigns them a score based on how likely they are to be a bot. However, there are significant differences among that areas and fake news detection as explained in [12], and thus we do not consider them here. Finally, there are algorithms for detecting fake images online [2], but these are beyond the scope of this article.

3 INVESTIGATED ALGORITHMS

The investigated algorithms are the following: L1 Regularized Logistic Regression, C-Support Vector Classification, Gaussian Naive Bayes, Multinomial Naive Bayes, Decision Trees, Random Forests, Multi-Layer Perceptron, and Convolutional Neural Networks. The version of the first seven algorithms is that provided by scikit-learn [14], whereas for the last one we developed our own code according to [8].

4 EVALUATION ENVIRONMENT AND SETTINGS

4.1 Execution environment

Our tests were executed in two different servers, the first one was used for training the CNNs on a Tesla K20x GPU, and the second one for the rest of the algorithms. This is due to the fact that CNN training is a highly CPU-intensive task. The following table has the detailed specifications of the machines used in our experiments.

Table 1: Servers specifications.

	Server 1	Server 2
CPU Architecture	Haswell	Ivy Bridge
Model No.	Xeon E5-2695V3	Xeon E5-2620V2
# of Cores	14	6
Core Frequency	2.30 GHz	2.10 GHz
Main Memory	128 GB	128 GB
GPU	Nvidia Tesla K20x	None

4.2 Datasets

The datasets are described in Table 2.

Table 2: Datasets used in the evaluation.

dataset name	Dataset properties		
	size	*property*	*source*
"Liar, liar pants on fire": A new benchmark dataset for fake news detection	Training set size of 10269 articles	Two labels for the truthfulness ratings (real/fake) were used instead of the original six	[17]
The Signal Media One-Million News Articles Dataset	1 million articles	13000 articles were selected at random and marked as real news	Signalmedia[3]
Getting Real about Fake News	13000 articles	All 13000 articles were marked as fake news	Kaggle[4]

Before using any of our datasets, firstly we subjected them to some refinements like stop-word, punctuation and non-letters removal and finally we used the Porter2 English Stemmer algorithm for stemming, due to its improvements over the widely used Porter stemmer [15]. This was done in order to avoid noise in our data and make classification faster and more efficient.

Using the datasets from Table 2, we created three input datasets (experiments) on which we evaluated the algorithms. For the first experiment we used the Wang's training dataset [17] which contains various statements from PolitiFact[5], a Pulitzer Prize-winning Website. From this dataset we used only the headline of each news story and two labels for the truthfulness ratings (real/fake).

Using the two remaining datasets, we created two new datasets which contained a mix of true/fake headlines and a mix of true/fake body texts respectively. For the newly created datasets we chose to keep a balance between the true and fake news using the same number for them from the original datasets. The headlines dataset finally contained 25000 news stories titles that were selected at random from both original datasets and about the body text dataset,

[2] https://hoaxy.iuni.iu.edu/

[3] http://research.signalmedia.co/newsir16/signal-dataset.html

[4] https://www.kaggle.com/mrisdal/fake-news

[5] http://www.politifact.com/

using the fact that the average length of stories from five of the top sites that were shared on social media on December 2016 was between 200–1000 words[6], we collected 10000 body texts of a length between 150–4000 words. Here we present the results obtained from the first two datasets, which we call as Dataset1 and Dataset2.

4.3 Performance measures

Since we consider the fake news detection problem as a binary classification task, we evaluated the competitors in terms of the following commonly used measures, namely F1-measure and accuracy [11].

Moreover, we consider the execution time as another significant quantity to measure; it is comprised by the time to complete two tasks, namely training and classification. So, we measured the following two quantities:

- *Training time*, which indicates the total time (in seconds) needed for training the model.
- *Classification time*, which indicates the total time (in seconds) needed for providing the classification decision.

5 PERFORMANCE EVALUATION

5.1 Text-to-vector transformation

First of all we needed to transform the text into some numeric or vector representation. This numeric representation should depict significant characteristics of the text. There are many such techniques, for example, occurrence, term-frequency, TF-IDF, word co-occurrence matrix, word2vec and GloVe. In our tests, we used the following two techniques:

- *Word Embeddings*. A word embedding is a parameterized function mapping words of some language to high-dimensional vectors $W : words \longrightarrow R^n$. In our tests two different techniques were used:
  - *Pre-trained Word Vectors*. We use the publicly available Glove vectors [13] trained on 6 billion tokens of Wikipedia 2014 + Gigaword 5. The vectors have dimensionality of 50, 100, and 300.[7]
  - *Trained Word Vectors Based on our datasets*. We use word2vec from genism library to train our own vectors based on the selected datasets. The vectors have dimensionality of 50, 100, 300 and were trained using the continuous bag-of-words model. In order to get a single vector representation within each headline/article we averaged the corresponding word vectors.
- *Term Frequency-Inverse Document Frequency (TF-IDF)*. TF-IDF weighting scheme is the combination of two terms, the Term Frequency (TF) and Inverse Document Frequency (IDF). We defined TF-IDF as follows:

$$tf_{t,d} = \frac{\text{number of times term } t \text{ appears in a document}}{\text{total number of terms in the document}}$$

$$idf_t = \log \frac{\text{total number of documents}}{\text{number of documents with term } t \text{ in it}}$$

So, the final TF-IDF weight of the term t is given by the following product:

$$tf - idf_{t.d} = tf_{t,d} \times idf_t.$$

So, each competitor has seven variants, i.e., three variants due to the three different dimensions of the pre-training, three variants due to the three different dimensions of the training based on our datasets, and one variant based on TF-IDF. So our first step is to discover which of the six former variants is the best one for each competitor.

5.2 How many dimensions are necessary?

We ask the following two questions: *How many dimensions are preferable for our algorithms?*, and *Is it training based on the examined dataset or on benchmark datasets a better solution?*

We present the average accuracy of the six variants of each algorithm in Figures 1–2. Deviation is small, so average is quite a good measure for all algorithms with the exception of *DT* for Dataset1.

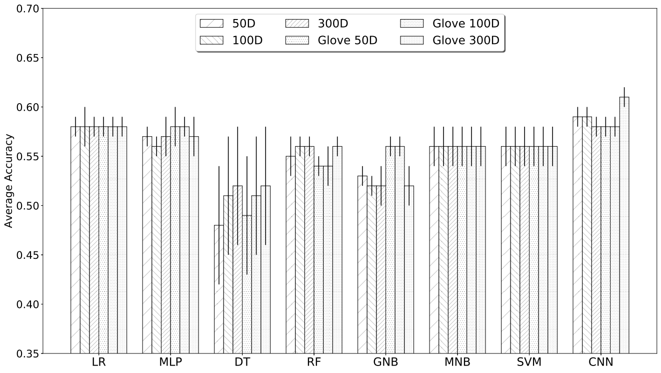

Figure 1: Average accuracies on Dataset1.

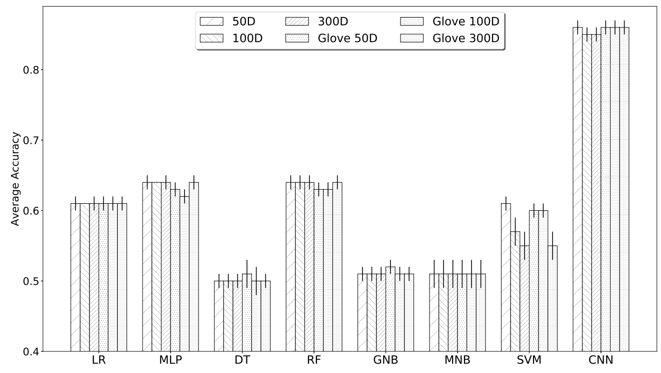

Figure 2: Average accuracies on Dataset2.

We present the average F1-measure of the six variants of each algorithm in Figures 3–4.

[6] https://www.newswhip.com/2017/01/long-shared-stories-social-media/

[7] https://nlp.stanford.edu/projects/glove/

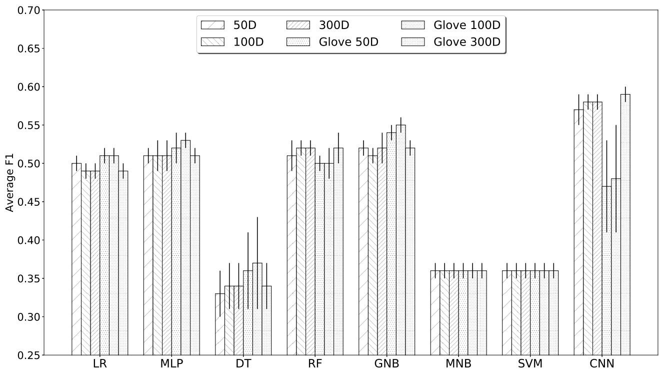

Figure 3: Average F1-measure on Dataset1.

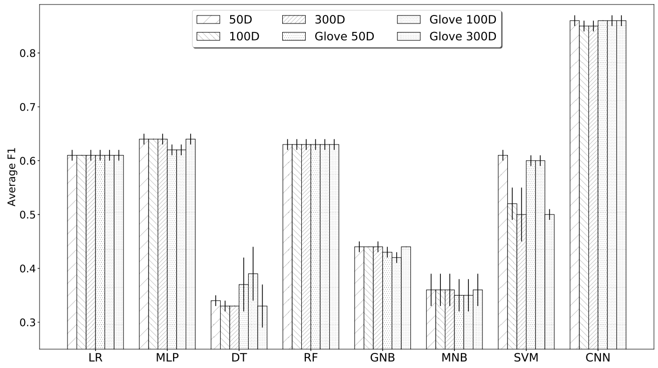

Figure 4: Average F1-measure on Dataset2.

It is expected that no choice on the number of dimensions and/or training on any kind of data can generate a variant of an algorithm that will be the champion one; such problems and the associated algorithms are highly dependent on data distributions. In Table 3 we present the variant of each algorithm that showed the best performance.

Table 3: Champion variant of each algorithm with respect to the number of dimensions and type of training.

Algorithm	*Dataset1*	*Dataset2*	*Dataset3*
LT	100D Glove	100D	50D
MLP	100D Glove	100D	50D
DT	100D Glove	100D Glove	50D Glove
RF	100D	300D Glove	50D
GNB	100D Glove	300D Glove	50D Glove
MNB	any variant	300D Glove	50D
SVM	any variant	50D	50D Glove
CNN	300D Glove	100D Glove	300D Glove

We can draw two quite evident conclusions from Table 3. The first observations is that a small or moderate number of dimensions is preferable because they do not create overfitted models. Secondly, pretraining based on benchmark datasets can be quite effective, meaning that such kind of pretraining is able to create models beating those generated on the specific data that are the target of investigation; this is a quite encouraging result.

5.3 Method of choice to generate vector representations

Based on the identified "champion" variant of each algorithm from the previous section, we ask the following question: *Is is preferable to use a TF-IDF scheme or word embeddings to generate vector representations of textual information?* The answer to this question is illustrated in Figures 5–8. The first three plots compare the performance of the champion word embedding variant against the TF-IDF variant of each algorithm from the perspective of average accuracy; whereas the other three plots contain the results from the perspective of average F1-measure.

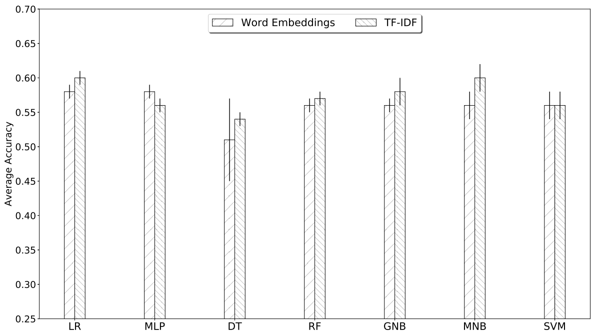

Figure 5: Average accuracies on Dataset1.

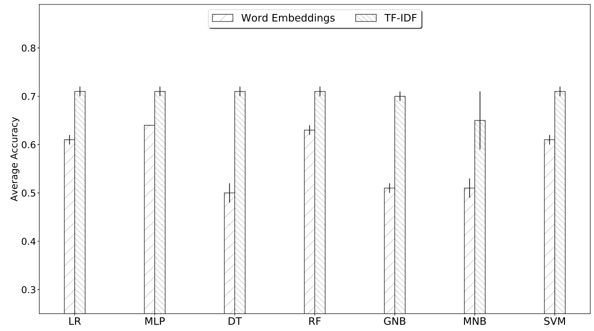

Figure 6: Average accuracies on Dataset2.

The results show clearly that the TF-IDF representation is a better alternative for the great majority of cases and algorithms. In particular, this representation achieves a 10% better performance in almost cases, in some cases this gap widens to reach a 30%. The only exception is for SVM in the case of Dataset3.

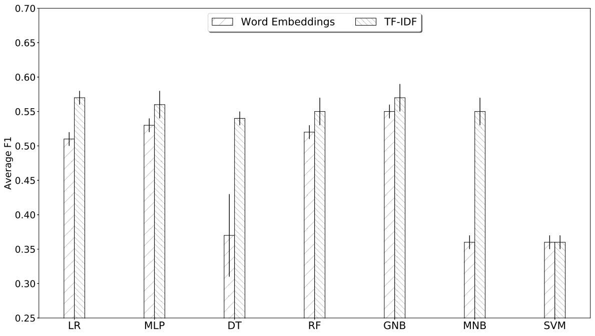

Figure 7: Average F1-measure on Dataset1.

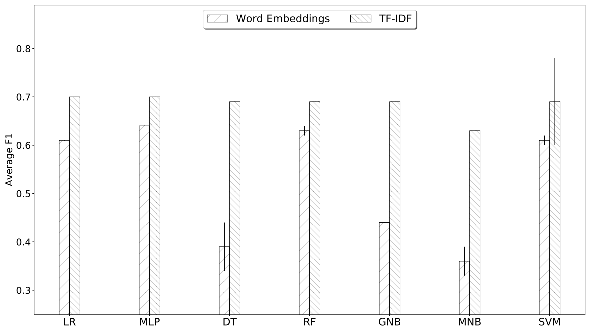

Figure 8: Average F1-measure on Dataset2.

5.4 Execution time

As far as the execution time is concerned, Table 4 shows the execution time – training and classification time – of all variants of the algorithms for Dataset1. In general, SVM and the neural network-based algorithms are the most time-consuming during the training phase, which is expected.

Table 4: Training/classification times (in seconds) for Dataset1.

				Glove Vectors			
Model	50D	100D	300D	50D	100D	300D	TF/IDF
LR	0.69-0.01	0.97-0.01	0.58-0.01	5.75-0.01	7.36-0.0	3.13-0.01	0.04-0.0
MLP	8.37-0.0	7.4-0.0	11.45-0.0	8.12-0.0	6.45-0.0	10.74-0.0	8.46-0.0
DT	1.1-0.0	02.01-0.0	6.39-0.0	1.1-0.0	1.76-0.0	5.44-0.0	0.58-0.0
RF	1.02-0.01	1.39-0.01	2.31-0.01	0.96-0.01	1.33-0.01	2.26-0.01	0.84-0.01
GNB	0.01-0.0	0.01-0.0	0.03-0.01	0.01-0.0	0.01-0.0	0.03-0.01	0.05-0.01
MNB	0.01-0.0	0.01-0.0	0.03-0.0	0.01-0.0	0.01-0.0	0.02-0.0	0.00-0.00
SVM	14.44-01.08	19.08-1.68	54.86-4.81	13.04-01.09	19.09-1.72	53.44-4.78	10.39-0.91
CNN	9.88-0.24	12.28-0.27	16.99-0.27	12.11-0.29	14.72-0.28	17.15-0.29	

6 CONCLUSIONS

The fast spreading of fake news and the impact they are having on our society, along with the inscalability of manually detecting them, have created a surge of research and development in machine learning algorithms to battle them. In this article, we evaluated representatives from eight well-known families of algorithms, namely regression, support vector classification, multi-layer perceptron, gaussian and multinomial naive Bayes, random forests, decision trees and convolutional neural networks against three publicly available datasets. We tested the efficiency and training speed of these algorithms. We concluded that a space with a hundred dimensions is of adequate dimensionality to capture the needed text features and get high accuracy of detection. Moreover, we established that the TF-IDF method for generating vectors from the text is a better alternative relative to word embeddings, and finally that pretraining based on benchmark datasets is able to reap performance benefits similar to that when training is performed based on the data under study. As far as the champion algorithm is concerned, we have shown that convolutional neural networks is the best performing algorithm with the downside of requiring significantly higher training time.

REFERENCES

[1] G.B. Guacho, S. Abdali, N. Shah and E. Papalexakis, "Semi-supervised content-based detection of misinformation via tensor embeddings," Technical Report. Available at: https://arxiv.org/abs/1804.09088, 2018.

[2] A. Gupta, H. Lamba, P. Kumaraguru and A. Joshi, "Faking Sandy: Characterizing and identifying fake images on Twitter during Hurricane Sandy," *Proceedings of the ACM International Conference on World Wide Web (WWW)*, pp. 729–736, 2010.

[3] M. Gupta, P. Zhao and J. Han, "Evaluating event credibility on Twitter," *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pp. 153–164, 2012.

[4] A. Gupta and P. Kumaraguru, "Credibility ranking of tweets during high impact events," *Proceedings of the Workshop on Privacy and Security in Online Social Media*, 2012.

[5] M. Hardalov, I. Koychev and P. Nakov, "In search of credible news," *Proceedings of the Artificial Intelligence: Methodology, Systems and Applications*, pp. 172–180, 2016.

[6] 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," Technical Report. Available at http://arxiv.org/abs/1703.09398

[7] S. Hosseinimotlagh, E. Papalexakis, "Unsupervised content-based identification of fake news articles with tensor decomposition ensembles," *Proceedings of the Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*, 2018.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[9] D.P. Kingma, J.L. Ba, "ADAM: A method for stochastic optimization", *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[10] X. Liu, B. Zhang, A. Susarla, R. Padman, "Go to YouTube and See me tomorrow: The role of social media in managing chronic conditions." Available at https://ssrn.com/abstract=3061149.

[11] C.D. Manning, P. Ragnavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[12] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations*, vol. 19, iss. 1, pp. 22–36, 2017.

[13] J. Pennington, R. Socher, and C.D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] M.F. Porter, "An algorithm for suffix stripping", *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[16] S. Vosoughi, D. Roy, S. Aral, "The spread of true and false news online," *Science*, vol. 359, iss. 6380, pp. 1146–1151, 2018.

[17] W.Y. Wang, ""Liar, liar pants on fire": A new benchmark dataset for fake news detection," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 422–426, 2017.