Ben-Gurion University of the Negev

Industrial Engineering and Management

364-2-1651 - Machine Learning And Data Mining

Lecturer: Boaz Learner

TA: Stanislav Khoroshevsky

Project Topic: Fake News Detection

Authors:

Nicole Jiaqi Li (850052085)
Olalekan Olusegun Isola (850335050)
Trivikram Muralidharan (850307943)

Submission Date: 19/07/2021

ABSTRACT

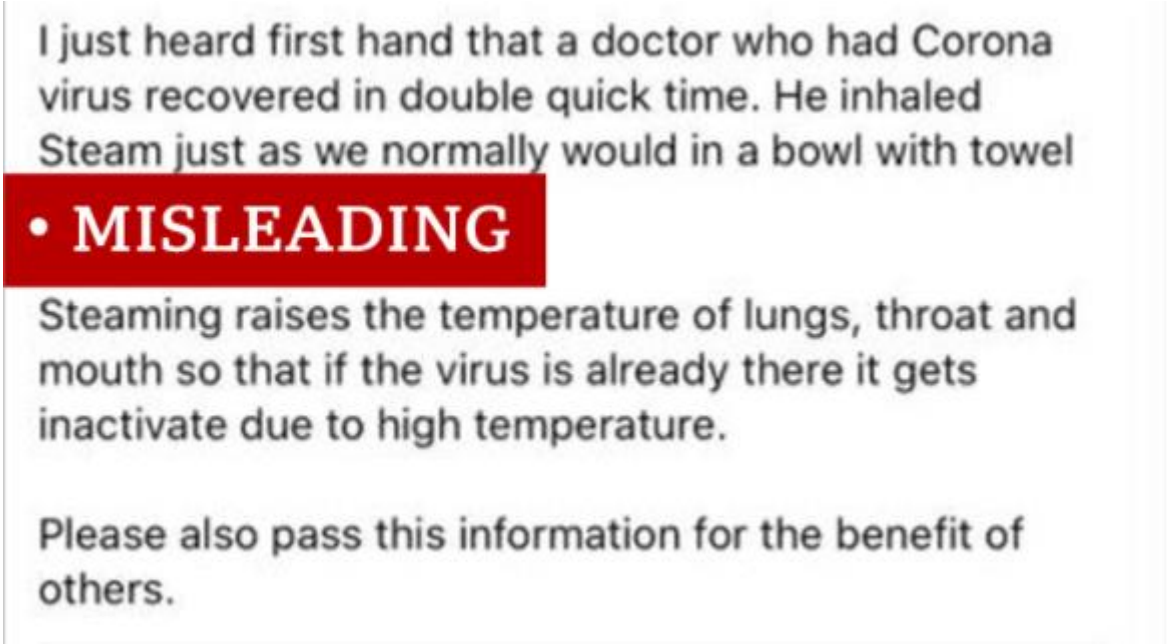
Fake news is a problem that is becoming commonplace around the world, causing the spread of misinformation, panic and distress among the general public. The increasing sophistication of fake news articles that are published in online mediums that have large readership, such as Twitter and Online News Media sites, makes it an urgent and important issue to tackle. This project deals with developing machine learning and deep learning (LSTM) models for the purpose of detecting fake news articles. The project evaluates and compares various text representation methods and their efficacy in a binary classification problem (fake news detection) and compares various machine learning and deep learning models. The best performing classifier is then subjected to more rigorous tests in order to understand the robustness of the model from the dual perspectives of class imbalance in the training set and the problem of mislabeled training instances. The project follows the CRISP-DM methodology and has also hosted a live demo version of the various detectors developed online.

Table of Contents

Table of Contents

1. Business Understanding

The phenomenon of widespread fake news began ever since the start of the digital age. What was once used as a strategic tactic in war times between countries (fake propaganda), is now being exploited by various malicious actors (whether they be individuals, organizations, or state sponsored actors) for either monetary gain, or other hidden agendas. The kind of damage done by fake news is manifold. For example, fake news in medicine has the chance to cause the public to either fear authentic medicine or to ingest improper ones leading to an overall degradation in their health. With a huge percentage of the population of the world having access to many news streams from the internet (via their mobile phones), misinformation tends to spread quickly and efficiently.



I just heard first hand that a doctor who had Corona virus recovered in double quick time. He inhaled Steam just as we normally would in a bowl with towel

• MISLEADING

Steaming raises the temperature of lungs, throat and mouth so that if the virus is already there it gets inactivate due to high temperature.

Please also pass this information for the benefit of others.

Figure 1. An example fake news article during the recent pandemic

With the advent of deep fakes and increasingly sophisticated text generation models that use deep learning, fake news articles have become even more convincing and believable, and it is the need of the hour to combat such news articles with corresponding innovations in fake news detection.

The lack of adequate instantaneous, and unbiased news verification systems presents a gap in the technological era that needs to be filled as soon as possible.

As a part of this project, we not only evaluate various text representation methods and models for the purpose of generating a classifier for fake news detection, but we also host a demo using the models that we created on the internet where anyone can test their text-based news articles for authenticity.

2. Data Understanding

The dataset used in the project is a Kaggle dataset (Source : https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset) of Fake and True News articles that were collected from 31st March 2015 to 19th February 2018.

Attributes of the dataset are as follows :

Classwise Count :

- Fake – 23460
- True - 21417

Features available:

- Title – Title of the news article
- Text – Textual content of the news article
- Subject – General subject of the news article
- Date – Date of publication of the news article

3. Data Preparation

For the purpose of making the data compatible to the experiments that we wish to conduct, we have taken the following measures:

3.1. Removal of Missing Data:

Entries in the data the contained empty values in the critical feature "text" were dropped (since they do not add much value to the data collection)

3.2. Lowercasing the textual data:

All data was converted to lowercase (in order to handle smoothen case conflicts)

3.3. Stop word removal:

For each entry, all the common stop words like "of", "on", "a", "the" were removed and filtered from the text. This is done to allow classifiers to more easily identify what words tend to contribute towards making a news article true or fake. Having stop words as a part of the data tends to increase the FPR (since random occurrences of these common words as a side-effect of the English language might be mistaken for actual indicators).

3.4. Removal of non-alphabetical character words:

For each entry, all the textual data (in "title" and "text" features) were filtered for words comprised of only alphabetical characters and the rest of the words were discarded. This is done in order to aid in the classification task (since words that do not consist entirely of alphabetical characters is considered to be noise)

3.5. Text Representation Methods :

Since one of our experiments is about comparing various text representation methods for the purpose of determining which methods is most suitable for the task of fake news detection, we have made use of three different text representation methods : TF-IDF, Bag of Words and Word2Vec.

3.5.1. TF-IDF: This method [1] is used for the purpose of allocating an "importance measure" for each word in the corpus of text data. It is the product between a count of the number of occurrences of a particular word in a given document (term frequency), with the log of the relative occurrence of the word across all documents in the corpus (inverse document frequency).

TF-IDF

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \log\left(\frac{\mathrm{N}}{\mathrm{df}_i}\right)$$

Figure 2. TF-IDF formula

3.5.2. Bag of Words: This method [2] is used for the purpose of allocating a representation of a an entry in the corpus using the unique words present across the entire corpus.

3.5.3. Word2Vec: This method introduced by Mikolov et al. [3] is used for the purpose of converting a given entry into a vectorized form of fixed size. These vectors are generated in such a way that words which have similar meaning are given values close to each other (located close to each other on the vector space)

4. Modeling

The experiments that we wish to conduct are two-fold:

- Select the best classifier and the best text representation method for the given dataset
- Evaluate the best performing classifier (hero) on its robustness towards incorrect trainset labelling and class imbalance issues.

Figure 3 illustrates the experimental design that we follow.

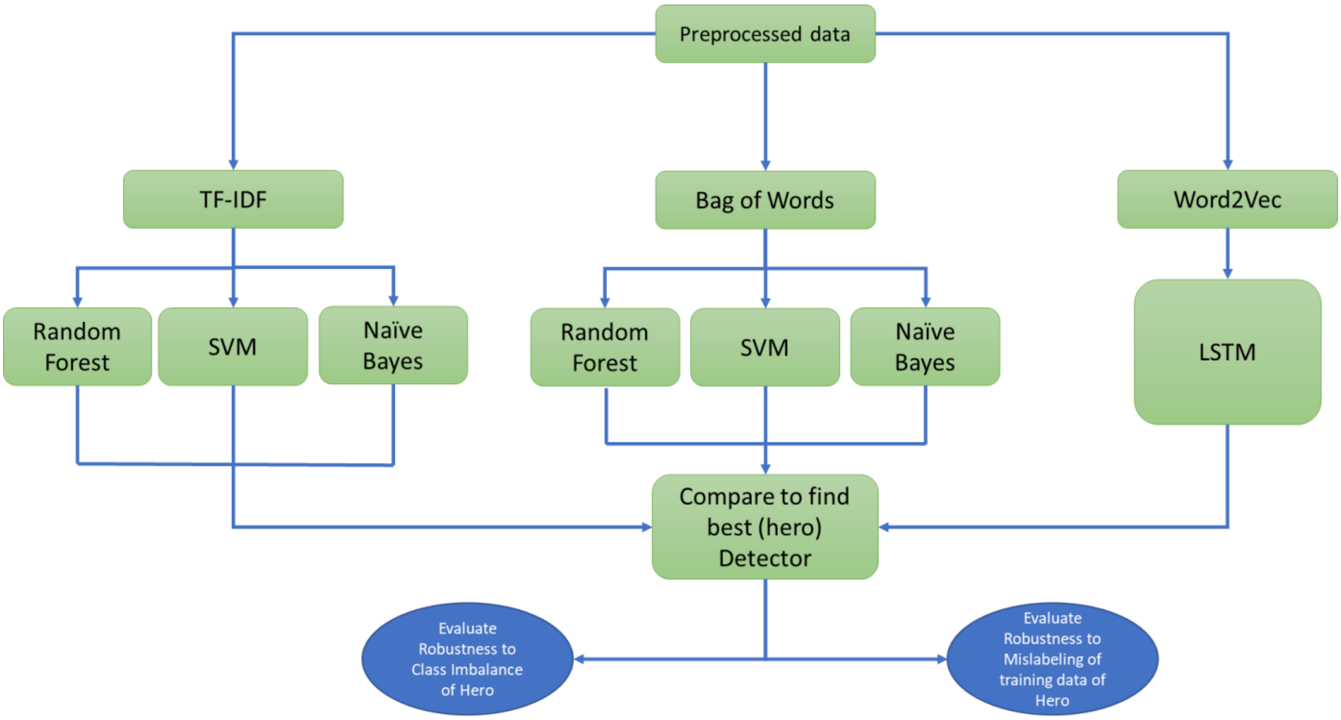

Preprocessed data
TF-IDF
Bag of Words
Word2Vec
Random Forest
SVM
Naïve Bayes
Random Forest
SVM
Naïve Bayes
LSTM
Compare to find best (hero) Detector
Evaluate Robustness to Class Imbalance of Hero
Evaluate Robustness to Mislabeling of training data of Hero

Figure 3. Experimental Design

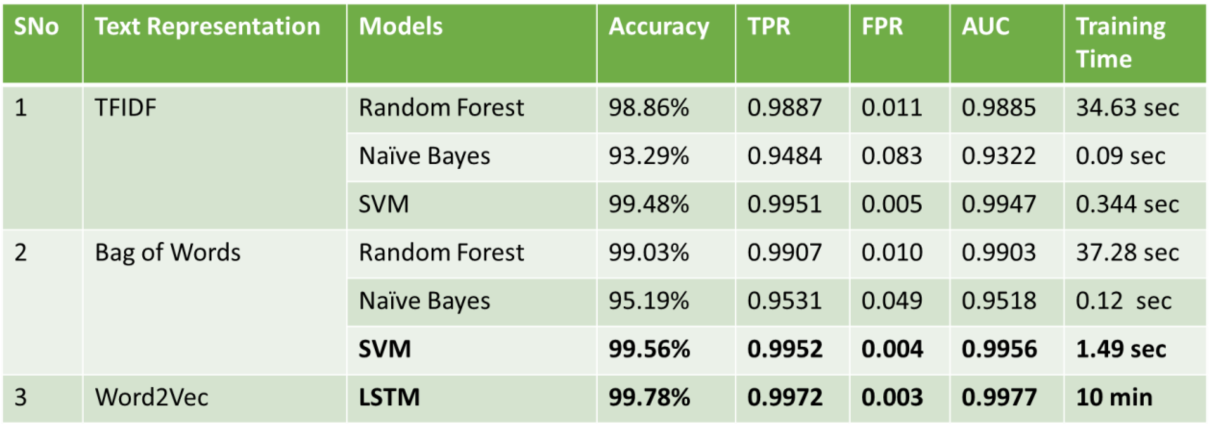
As can be seen in Figure 3, we utilize three different text representation methods and four different classifiers.

We have used a Naïve Bayes, Random Forest and SVM with the TF-IDF and BoW representation methods, and an LSTM [4] on the Word2Vec Representation method. In the following section, we expand upon the results that we obtained for each of these experiments.

5. Evaluation

5.1 Experiment 1 : Comparing text representation methods and classifiers

As can be seen in Figure 4, the SVM performed the best for both TF-IDF and BoW text representation methods and the LSTM with Word2Vec performed the best overall.



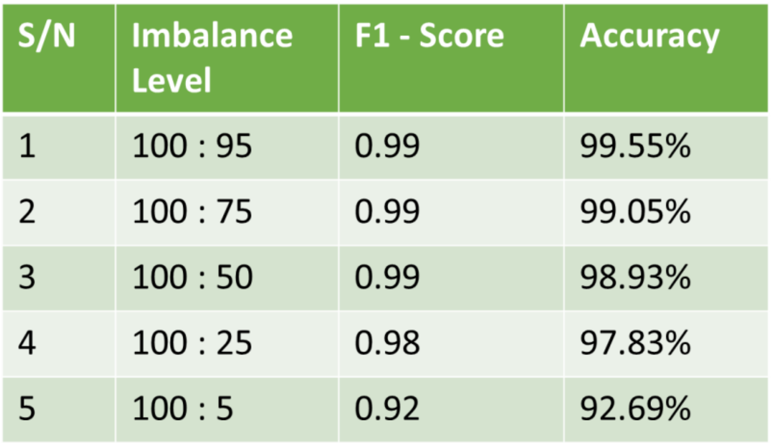
SNo	Text Representation	Models	Accuracy	TPR	FPR	AUC	Training Time
1	TFIDF	Random Forest	98.86%	0.9887	0.011	0.9885	34.63 sec
		Naïve Bayes	93.29%	0.9484	0.083	0.9322	0.09 sec
		SVM	99.48%	0.9951	0.005	0.9947	0.344 sec
2	Bag of Words	Random Forest	99.03%	0.9907	0.010	0.9903	37.28 sec
		Naïve Bayes	95.19%	0.9531	0.049	0.9518	0.12 sec
		SVM	**99.56%**	**0.9952**	**0.004**	**0.9956**	**1.49 sec**
3	Word2Vec	**LSTM**	**99.78%**	**0.9972**	**0.003**	**0.9977**	**10 min**

Figure 4. Table of Results – Experiment 1.

Now, the best performing detector from Experiment 1 (LSTM with Word2Vec) is used for the purpose of running experiment 2.

5.2. Experiment 2 : Model Robustness to Data Imbalance and Incorrect Labelling

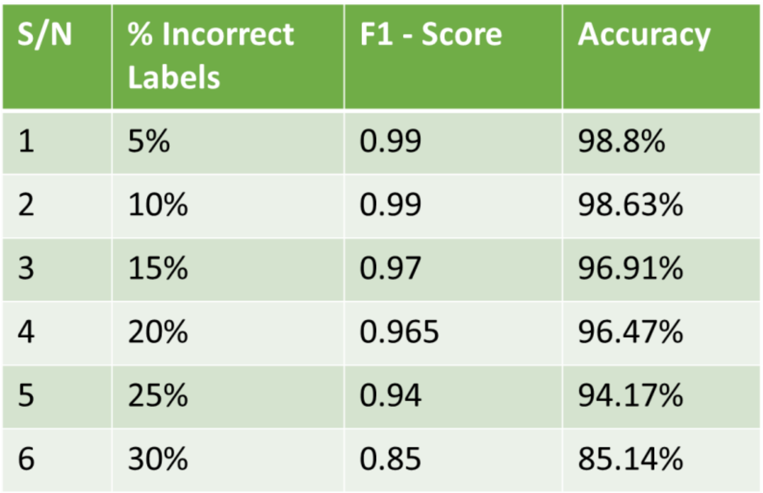
As can be seen in Figure 5, the LSTM has poor accuracy (due to excessively high FPR) in case 5 (100:5), but performs relatively well under all other circumstances. It is therefore correct to assume that the model is robust to data imbalance.



S/N	Imbalance Level	F1 - Score	Accuracy
1	100 : 95	0.99	99.55%
2	100 : 75	0.99	99.05%
3	100 : 50	0.99	98.93%
4	100 : 25	0.98	97.83%
5	100 : 5	0.92	92.69%

Figure 5. LSTM Robustness to class imbalance

Incorrect labelling is a problem that plagues many data science projects today. In order to simulate this, we have purposefully mislabelled some instances of the training set in order to observe how robust the LSTM model is to incorrect labelling in the training set. The results of this experiment are illustrated in Figure 6. and the LSTM is show to be relatively robust to incorrect labelling and seems to fail significantly only between the 25% - 30% incorrect labels distribution (which is a generous assumption of mislabelling that is rare to occur in real life). Given this result, it is fair to state that the LSTM is robust to the presence of mislabelled instances of data in the training dataset.



S/N	% Incorrect Labels	F1 - Score	Accuracy
1	5%	0.99	98.8%
2	10%	0.99	98.63%
3	15%	0.97	96.91%
4	20%	0.965	96.47%
5	25%	0.94	94.17%
6	30%	0.85	85.14%

Figure 6. LSTM Robustness to incorrect labelling in training set

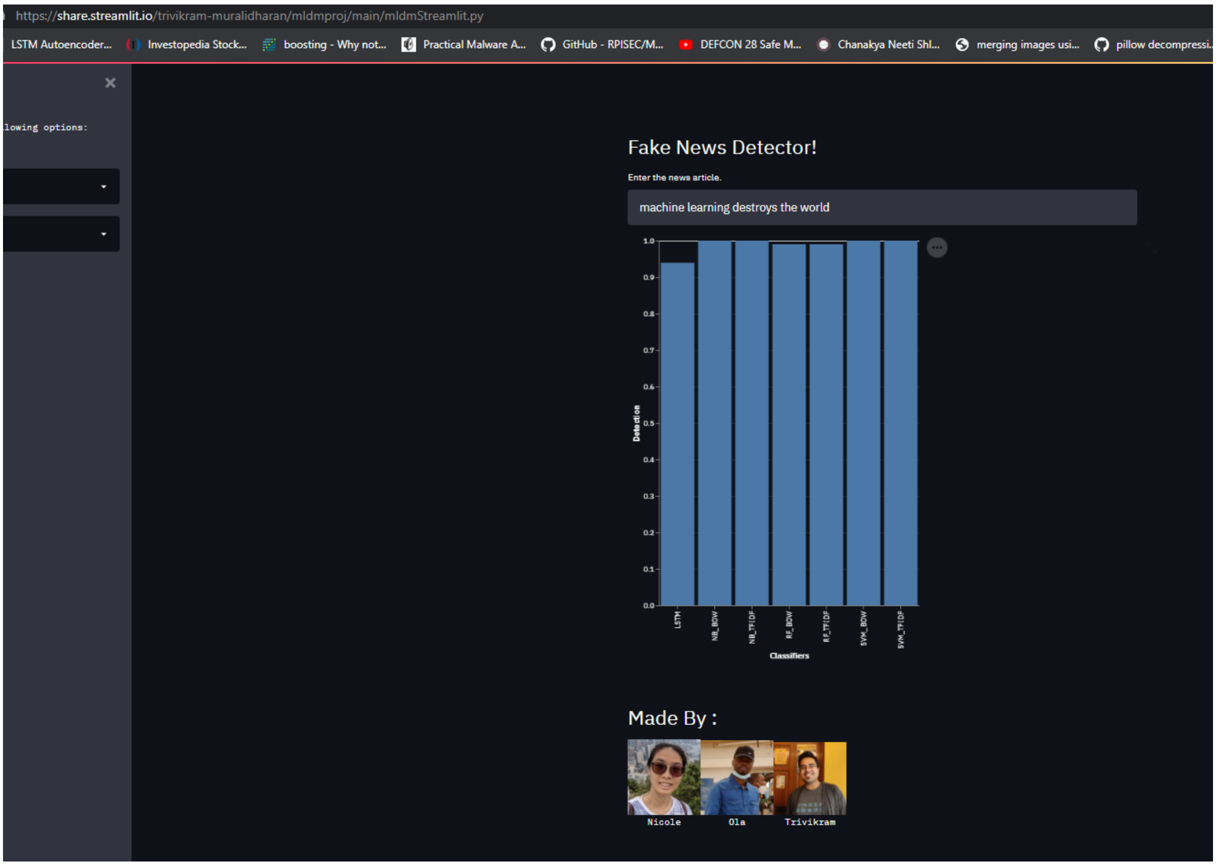
6. Deployment

For the purpose of deployment (and live testing of the models that we built in the project), we decided to build a Streamlit [5] application that would allow us to share the application with people on the internet to test if news articles they read are true or fake. Please visit the following website to test it out yourself : https://share.streamlit.io/trivikram-muralidharan/mldmproj/main/mldmStreamlit.py (note : there is a good chance that our compute resource quota would have ended by the time you test the website. In case you wish to see the demo during your evaluation of the project, please drop us a message and we will refresh and rehost it on the same URL. The code for this hosted project is included in the code we submitted. "Scripts_for_hosting")

The application is essential a single page app where any user can enter textual content that they wish to verify. Each of the models that we have trained in our experiments have been saved (along with their tokenizers / vectorizers for generating corresponding input representation for new test data received as input on the website) and are called to predict the "fakeness" of the textual input that is received on the website.

The predictions of each of these models for the given text input are displayed on the website as a bar graph.

Figure 7 is a screenshot from our hosted application on Streamlit.



https://share.streamlit.io/trivikram-muralidharan/mldmproj/main/mldmStreamlit.py

lowing options:

Fake News Detector!

Enter the news article.

machine learning destroys the world

Made By :

Nicole Ola Trivikram

Figure 7. A screenshot from the hosted demo

7. Summary

For the purpose of this project, we utilized the CRISP-DM methodology for data mining starting from the Business Understanding to the Deployment phase.

From the experiments that were conducted, it was seen that the LSTM model works best for the dataset that was used.

The experiments that tested the LSTM also showed that the LSTM was robust to the class imbalance problem and towards human error in labelling of training data.

There are several well cited research works [6,7] that have tried to deal with the problem of fake news detection. While they have suggested features that are useful and have also shown that there are gaps in the literature for addressing several issues in the domain (such as fake video detection or fake audio detection), the experiments conducted were always done under the most favorable of circumstances. The problem of class imbalance and mislabeled data have conveniently not been discussed in most of these articles.

As a concluding remark, we hope for this project to function as a reminder to all readers that the real life deployment of machine learning models always presents challenges from an external validity perspecitve. More often than not, it is discovered by the machine learning developer that the models he trained, (and more importantly the data that was used to train the model) are far from sufficient to deal with the myriad edge cases that exist in the wild. It would be prudent for research to begin evaluating ensemble models that are capable of handling greater variance in the data.

REFERENCES

[1] Juan Enrique Ramos. 2003. Using TF-IDF to Determine Word Relevance in Document Queries. (2003).

[2] Bag-of-words model - Wikipedia. Retrieved July 19, 2021 from https://en.wikipedia.org/wiki/Bag-of-words_model

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.* (January 2013). Retrieved July 19, 2021 from https://arxiv.org/abs/1301.3781v3

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (November 1997), 1735–1780. DOI:https://doi.org/10.1162/NECO.1997.9.8.1735

[5] Streamlit. Retrieved July 19, 2021 from https://streamlit.io/

[6] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic Detection of Fake News. *CEUR Workshop Proc.* 2789, (August 2017), 168–179. Retrieved July 19, 2021 from https://arxiv.org/abs/1708.07104v1

[7] ShuKai, SlivaAmy, WangSuhang, TangJiliang, and LiuHuan. 2017. Fake News Detection on Social Media. *ACM SIGKDD Explor. Newsl.* 19, 1 (September 2017), 22–36. DOI:https://doi.org/10.1145/3137597.3137600