FAKE NEWS DETECTOR

Dissertation submitted to
Shri Ramdeobaba College of Engineering & Management, Nagpur
In partial fulfilment of requirement for the award of
degree of

Bachelor of Engineering

in

COMPUTER SCIENCE & ENGINEERING

By

Dnyaneshwari Vyas

Gauri Chandak

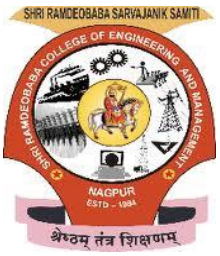
Aashray Kashyap

Ameya Yerpude

Atharva Deshmukh

Guided

Prof. D. Naidu



Computer Science & Engineering
Shri Ramdeobaba College of Engineering & Management, Nagpur 440 013
(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

April 2019

FAKE NEWS DETECTOR

Dissertation submitted to
Shri Ramdeobaba College of Engineering & Management, Nagpur
In partial fulfilment of requirement for the award of
degree of

Bachelor of Engineering

in

COMPUTER SCIENCE & ENGINEERING

By

Dnyaneshwari Vyas

Gauri Chandak

Aashray Kashyap

Ameya Yerpude

Atharva Deshmukh

Guided

Prof. Devishree Naidu

Computer Science & Engineering
Shri Ramdeobaba College of Engineering & Management, Nagpur 440 013
(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

April 2019

SHRI RAMDEOBABA COLLEGE OF ENGINEERING & MANAGEMENT, NAGPUR

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the report on "**Fake News Detector**" is a bonafide work of Dnyaneshwari Vyas, Gauri Chandak, Aashray Kashyap, Ameya Yerpude & Atharva Deshmukh submitted to the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur in partial fulfilment of the award of a Bachelor of Engineering, in Computer Science & Engineering has been carried out at the Department of Computer Science & Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur during the academic year 2018– 2019.

Date: 15th April 2019

Place: Nagpur

Prof. Devishree Naidu	Dr. Manoj B. Chandak
Project Guide	H.O.D
Department of Computer Science & Engineering	Department of Computer Science & Engineering

Dr. R. S. Pande

Principal

DECLARATION

I, hereby declare that the report titled “Fake News Detector” submitted herein, has been carried out in the Department of Computer Science & Engineering of Shri Ramdeobaba College of Engineering & Management, Nagpur. The work is original and has not been submitted earlier as a whole or part for the award of any degree at this or any other institution.

Date: 15th April 2019

Place: Nagpur

Dnyaneshwari Vyas (06)

Gauri Chandak (07)

Aashray Kashyap (31)

Ameya Yerpude (37)

Atharva Deshmukh (40)

Approval sheet

This report entitled "**Fake News Detector**" by Dnyaneshwari Vyas, Gauri Chandak, Aashray Kashyap, Ameya Yerpude & Atharva Deshmukh is approved for the degree of Bachelor of Engineering in Computer Science & Engineering from Shri Ramdeobaba College of Engineering & Management, Nagpur.

Name & signature of Supervisor(s)

Name & signature of External Examiner(s)

Name & signature of RRC Members

Name & signature of HOD

Date: 15th April 2019

Place: Nagpur

ACKNOWLEDGEMENTS

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation & support made it possible, whose constant guidance and encouragement crown all efforts with success.

We are very grateful to our project supervisor Prof. Devishree Naidu for the guidance, inspiration and constructive suggestions that helpful me in the preparation of this project. We also thank our HOD and overall supervisor Dr. M. B. Chandak for their wonderful and skilful guidance in assisting us with the necessary support to ensure that our project is a success. We also thank our parents and family at large for their moral and financial support in funding the project to ensure successful completion of the project.

ABSTRACT

In this project, we explore the application of Machine Learning techniques to identify whether a news source is reliable or unreliable. We use a corpus of labelled real and fake new articles to build a classifier that can make decisions about information based on the content from the corpus. We use a text classification approach, using four different classification models, and analyse the results. The best performing model was the LSTM implementation.

The model focuses on identifying fake news sources, based on multiple articles originating from a source. Once a source is labelled as a producer of fake news, we can predict with high confidence that any future articles from that source will also be fake news. Focusing on sources widens our article misclassification tolerance, because we then have multiple data points coming from each source.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

FAKE NEWS DETECTOR

1.1 INTRODUCTION

1.1.1 Background

Internet and social media made the access to the news information much easier and comfortable. Often Internet users can follow the events of their interest in online mode, and spread of the mobile devices makes this process even easier.

But with great possibilities come great challenges. Mass media have a huge influence on the society, and as it often happens, there is someone who wants to take advantage of this fact. Sometimes to achieve some goals mass-media may manipulate the information in different ways. This leads to producing of the news articles that are not completely true or even completely false. There even exist lots of websites that produce fake news almost exclusively. They deliberately publish hoaxes, propaganda and disinformation purporting to be real news – often using social media to drive web traffic and amplify their effect. The main goal of fake news websites is to affect the public opinion on certain matters (mostly political). Examples of such websites may be found in Ukraine, United States of America, Germany, China and lots of other countries. Thus, fake news is a global issue as well as a global challenge.

Many scientists believe that fake news issue may be addressed by means of machine learning and artificial intelligence. There is a reason for that: recently artificial intelligence algorithms have started to work much better on lots of classification problems (image recognition, voice detection and so on) because hardware is cheaper and bigger datasets are available.

1.1.2 Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.1.3 Problem Definition

With the rise of social media and the internet in general, we see that information is easily accessible and is relied upon by millions as credible sources of information on just about everything. Not all information is reliable though, many people use these mediums to propagate fake news for their personal gains or to cause damage to others, like before elections there is a lot buzz on the internet, which tries to sway people's opinion for political gains.

1.1.4 Objectives

The objectives of this project include:

- To help users in differentiating between real and fake news.
- To help make users informed decisions.

Chapter 2

REVIEW OF LITERATURE

2.1 DEFINITION OF FAKE NEWS

The credibility of news is defined by many words such as trustworthiness, accuracy, fairness, objectivity etc. Machine learning approach can be used on various platforms to check the credibility of a news source.

Fake news or junk news or pseudo-news is a type of yellow journalism or propaganda that consists of deliberate disinformation or hoaxes spread via traditional print and broadcast news media or online social media. The false information is often caused by reporters paying sources for stories, an unethical practice called checkbook journalism. Digital news has brought back and increased the usage of fake news, or yellow journalism. The news is then often reverberated as misinformation in social media but occasionally finds its way to the mainstream media as well.

For example, an alleged Facebook post that was offensive to a particular religion led to large scale violence, arson and communal tension in Basirhat town in West Bengal in 2017. Social media post by a 17-year-old boy lead to violence and rioting and ask whether political parties are contributing to escalating tensions instead of helping curb the tension.

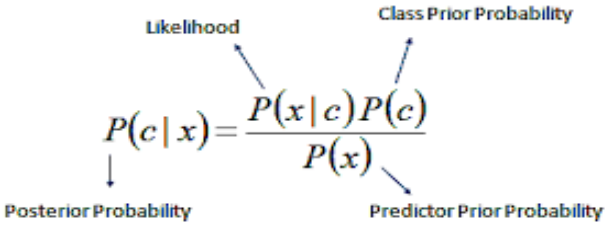
2.2 MACHINE LEARNING METHODS

This project uses four methods to classify a news source as reliable or unreliable. They are

- Naïve Bayes Classification
- Support Vector Machine (SVM)
- Neural Network
- Long Short Term Memory (LSTM).

2.2.1 Naïve Bayes Classifier

It is a well-known classification method, based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

2.2.2 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

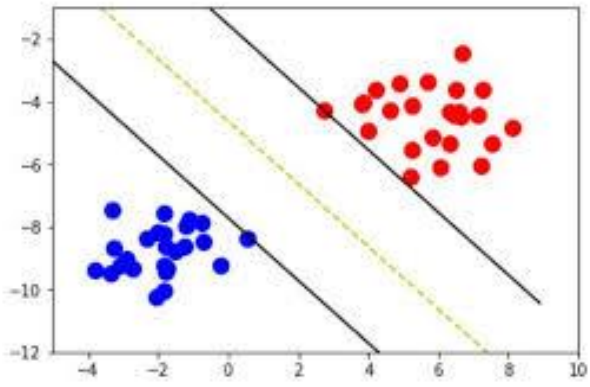


Figure 1: Support Vector Machine Classification

2.2.3 Neural Network

A neural network usually involves a large number of processors operating in parallel and arranged in tiers. The first tier receives the raw input information -- analogous

to optic nerves in human processing. Each successive tier receives the output from the tier preceding it, rather than from the raw input -- in the same way neurons further from the nerve receive signals from those closer to it. The last tier produces the output of the system.

.

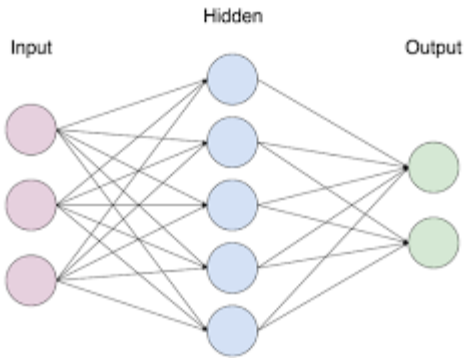


Figure 2: Neural Network

2.2.4 Long Short-Term Memory (LSTM)

These are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

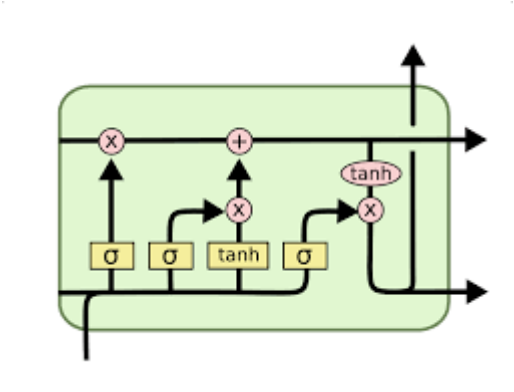


Figure 3: Long Short Term Memory

Chapter 3

SYSTEM METHODOLOGY

3.1 METHODOLOGY

In this chapter, the methods of data collection and the organization of the new system are discussed. It includes methods which were used in order to achieve the objectives of the project, particular requirements for implementation of the project.

3.2 MODULES OF THE SYSTEM

3.2.1 MODULE 1: ANALYSING DATASETS

The existing data is very large and needs to be cleaned to get specific outputs for visualisation purposes. The data available gives us all the information required for processing. What needs to be done is decide on attributes or parameters that will give the results required for a particular query. This query is then created to give the output for the data visualisation.

3.2.1.1 Flow of Control

What we implement here is the processing and obtaining of specific sub sets of data from the given data to display a desired output. First, the parameters are chosen to be used in the query (for e.g. query is to display the text and label in each row and process it to pass it model for training). The resulting data is passed onto the pre-processing module for further pre-processing.

3.2.2 MODULE 2: DATA PRE-PROCESSING

For every dataset the data may not always be in correct form, there can be some discrepancies in the data and machine learning model directly doesn't use text as input. So, to remove this discrepancies and to provide models with the correct information for training the following modules are performed on the data obtained.

3.2.2.1 Cleaning the data

The dataset obtained is to be processed as the dataset might contains non-useful information is removed from the dataset. This cleaning of data includes removal of stop words , deleting special characters, punctuations and converting the text to lowercase.

3.2.2.2 Feature Extraction

The embeddings used for the majority of our modelling are generated using the Doc2Vec model. The goal is to produce a vector representation of each article. Before applying Doc2Vec, we perform some basic pre-processing of the data. By cleaning the data , it produces a comma-separated list of words, which can be input into the Doc2Vec algorithm to produce an 300-length embedding vector for each article. Doc2Vec is a model developed in 2014 based on the existing Word2Vec model, which generates vector representations for words. Word2Vec represents documents by combining the vectors of the individual words, but in doing so it loses all word order information. Doc2Vec expands on Word2Vec by adding a "document vector" to the output representation, which contains some information about the document as a whole, and allows the model to learn some information about word order. Preservation of word order information makes Doc2Vec useful for our application, as we are aiming to detect subtle differences between text documents.

3.2.3 MODULE 3: APPLYING MODEL

The pre-processed data is then passed to the models for trainings.

3.2.3.1 Naïve-Bayes

This is one of the simplest approaches to classification, in which a probabilistic approach is used, with the assumption that all features are conditionally independent given the class label. As with the other models, we used the Doc2Vec embeddings described above. The Naive Bayes Rule is based on the Bayes' theorem

$$P(c|x) = P(x|c)P(c)/P(x)$$

Parameter estimation for naive Bayes models uses the method of maximum likelihood. The advantage here is that it requires only a small amount of training data to estimate the parameters.

3.2.3.2 Support Vector Machine

We use the Radial Basis Function kernel in our project. The reason we use this kernel is that two Doc2Vec feature vectors will be close to each other if their corresponding documents are similar, so the distance computed by the kernel function should still represent the original distance. Since the Radial Basis Function is

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)$$

It correctly represents the relationship we desire and it is a common kernel for SVM. The main idea of the SVM is to separate different classes of data by the widest "street". This goal can be represented as the optimization problem

$$\arg\max_{w,b}\left\{\frac{1}{||w||}\min_{n}\left[t_n(w^T\phi(x_n)+b)\right]\right\}$$
$$s.t. \quad t_n(w^T\phi(x_n)+b) \geq 1, \qquad n = 1, \ldots N$$

Then we use the Lagrangian function to get rid of the constraints.

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{n=1}^{N} a_n\left\{t_n(w^T\phi(x_n)+b) - 1\right\}$$

where an ≥ 0, n = 1, . . . N.

Finally we solve this optimization problem using the convex optimization tools provided by Python package CVXOPT.

3.2.3.3 Neural Network:

We implemented two feed-forward neural network models, one using Tensorflow and one using Keras. Neural networks are commonly used in modern NLP applications, in contrast to older approaches which primarily focused on linear models such as SVM's and logistic regression. Our neural network implementations use three hidden layers. In the Tensorflow implementation, all layers had 300 neurons each, and in the Keras implementation we used layers of size 256, 256, and 80, interspersed with dropout layers to avoid overfitting. For our activation function, we chose the Rectified Linear Unit

(ReLU), which has been found to perform well in NLP applications. This has a fixed-size input x ∈ $R^{1\times300}$

$$h_1 = \text{ReLU}(W_1 x + b_1)$$
$$h_2 = \text{ReLU}(W_2 h_1 + b_2)$$
$$y = \text{Logits}(W_3 h_2 + b_3)$$

3.2.3.4 LSTM(Long Short Term Memory)

The Long-Short Term Memory (LSTM) unit was proposed by Hochreiter and Schmidhuber[8]. It is good at classifying serialized objects because it will selectively memorize the previous input and 3 use that, together with the current input, to make prediction. The news content (text) in our problem is inherently serialized. The order of the words carries the important information of the sentence. So the LSTM model suits for our problem. Since the order of the words is important for the LSTM unit, we cannot use the Doc2Vec for preprocessing because it will transfer the entire document into one vector and lose the order information. To prevent that, we use the word embedding instead. We first clean the text data by removing all characters which are not letters nor numbers. Then we count the frequency of each word appeared in our training dataset to find 5000 most common words and give each one an unique integer ID. For example, the most common word will have ID 0, and the second most common one will have 1, etc. After that we replace each common word with its assigned ID and delete all uncommon words. Notice that the 5000 most common words cover the most of the text, as shown in Figure 1, so we only lose little information but transfer the string to a list of integers. Since the LSTM unit requires a fixed input vector length, we truncate the list longer than 500 numbers because more than half of the news is longer than 500 words as shown in Figure 2. Then for those list shorter than 500 words, we pad 0's at the beginning of the list. We also delete the data with only a few words since they don't carry enough information for training. By doing this, we transfer the original text string to a fixed length integer vector while preserving the words order information. Finally we use word embedding to transfer each word ID to a 32-dimension vector. The word embedding will train each word vector based on word similarity. If two words frequently appear together in the text, they are thought to be more similar and the distance of their corresponding vectors is small. The pre-processing transfers each news in raw text into a fixed size matrix. Then we feed the processed training data into the LSTM unit to train the model. The LSTM is still a neural

network. But different from the fully connected neural network, it has cycle in the neuron connections. So the previous state (or memory) of the LSTM unit c_t will play a role in new prediction h_t.

$$
\begin{aligned}
h_t &= o_t \cdot \tanh(c_t) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c_t} \\
\tilde{c_t} &= \tanh(x_t W_c + h_{t-1} U_c + b_c) \\
o_t &= \sigma(x_t W_o + h_{t-1} U_o + b_o) \\
i_t &= \sigma(x_t W_i + h_{t-1} U_i + b_i) \\
f_t &= \sigma(x_t W_f + h_{t-1} U_f + b_f)
\end{aligned}
$$

Chapter 4

SYSTEM IMPLEMENTATION TECHNOLOGIES

4.1 SOFTWARE

1. Python

An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

2. Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python.

3. Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

4. **Scikit-Learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

Scikit Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Gaussian Naive Bayes (GaussianNB), can perform online updates to model parameters via partial_fit method.

Scikit SVM

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

Scikit.svm.svc - The implementation is based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples

Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

LSTM

The Long Short-Term Memory network or LSTM network is a type of recurrent neural network used in deep learning because very large architectures can be successfully trained.

Chapter 5

SYSTEM DESIGN

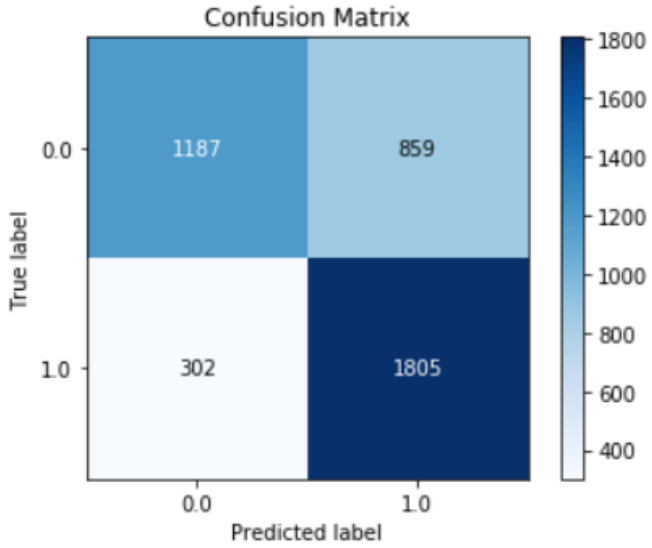


Figure 4: Confusion matrix for Naïve-Bayes

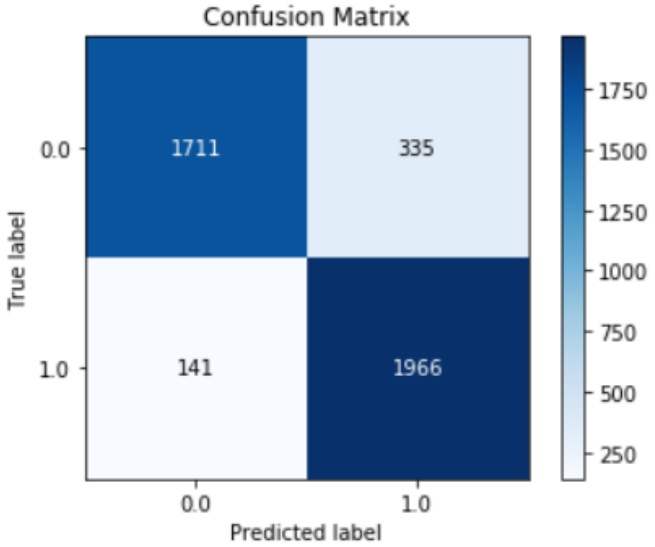


Figure 5: Confusion Matrix for SVM

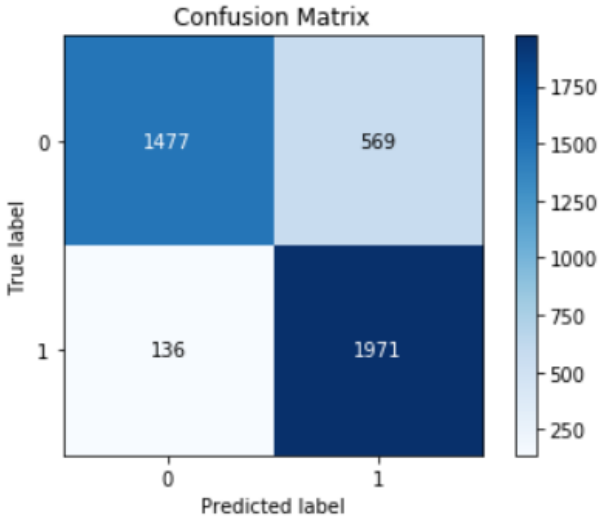


Figure 6: Confusion Matrix for NN-Tensorflow

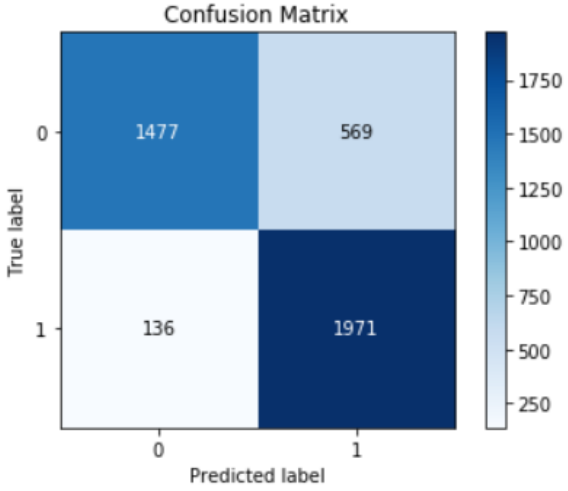


Figure 7: Confusion Matrix for NN-Keras

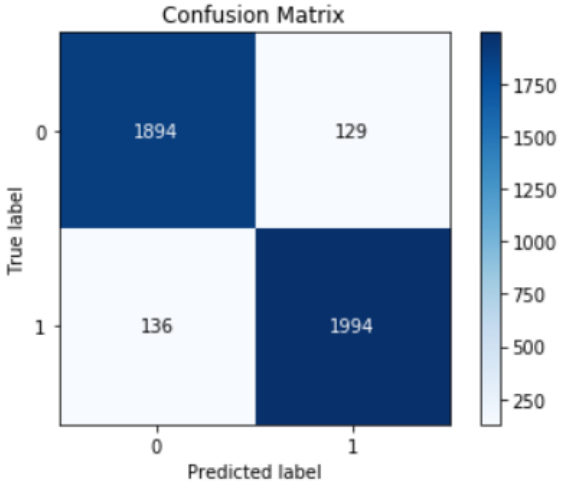


Figure 8: Confusion Matrix for LSTM

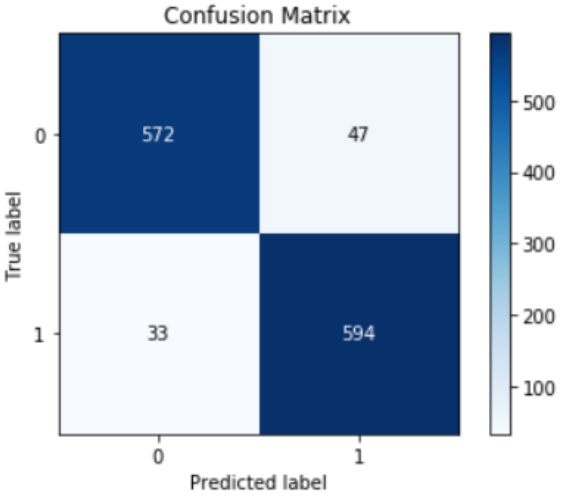


Figure 9: Confusion Matrix for Cross Validation

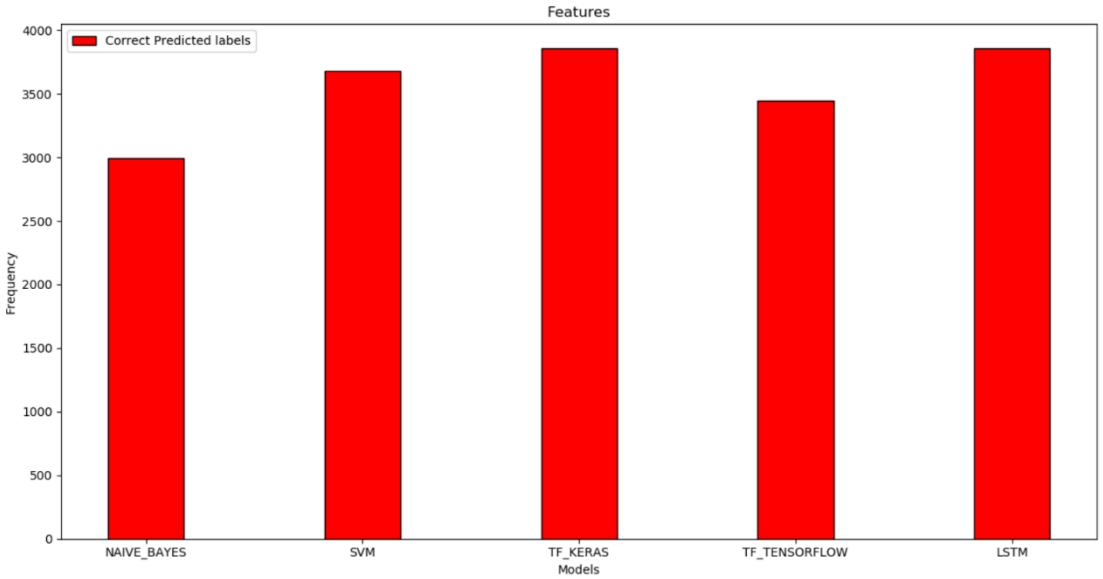


Figure 10: Bar Graph for correct predicted sources

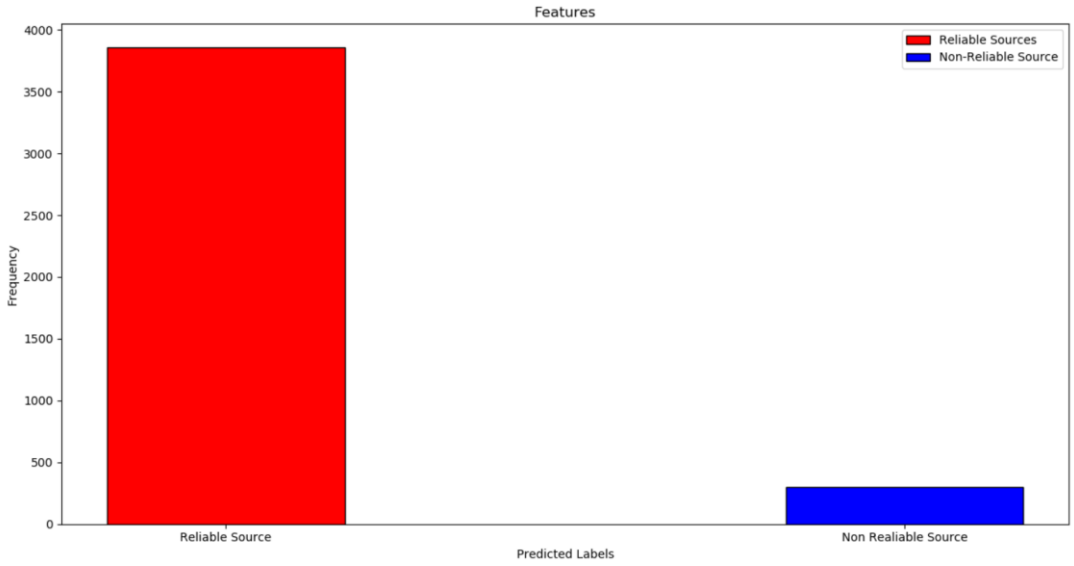


Figure 11: Predicted labels by models

Chapter 7

CONCLUSIONS AND SCOPE

7.1 CONCLUSION

With the increasing fame of social media, more and more people munch through news from social media instead of conventional news media. However, social media has also been used to spread fake news, which has strong harmful impacts on society. This project is an attempt to compare well known machine learning models in context of news classification. We learnt how to prepare a custom cross validation model using these models This project serves to help other programmers to help program a model for efficient fake news detection.

7.2 SCOPE OF PROJECT

A complete, production-quality classifier will incorporate many different features beyond the vectors corresponding to the words in the text. For fake news detection, we can add as features the source of the news, including any associated URLs, the topic (e.g., science, politics, sports, etc.), publishing medium (blog, print, social media), country or geographic region of origin, publication year, as well as linguistic features not exploited in this exercise use of capitalization, fraction of words that are proper nouns (using gazetteers), and others. Besides, we can also aggregate the well-performed classifiers to achieve better accuracy. For example, using bootstrap aggregating for the Neural Network, LSTM and SVM models to get better prediction result. An ambitious work would be to search the news on the Internet and compare the search results with the original news. Since the search result is usually reliable, this method should be more accurate, but also involves natural language understanding because the search results will not be exactly the same as the original news. So we will need to compare the meaning of two contents and decide whether they mean the same thing.

REFERENCES

[1] Datasets, *Kaggle,* https://www.kaggle.com/c/fake-news/data, February, 2018

[2] Allcott, H., and Gentzkow, M., *Social Media and Fake News in the 2016 Election,* https://web.stanford.edu/œgentzkow/research/fakenews.pdf, January, 2017.

[3]Christopher, M. Bishop, *Pattern Recognition and Machine Learning,* http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf, April, 2016.

[4]Goldberg, Y., *A Primer on Neural Network Models for Natural Language Processing,* https://arxiv.org/pdf/1510.00726.pdf, October, 2015.

[5] Hochreiter, S., Jrgen, S., *Long short-term memory,* http://www.bioinf.jku.at/publications/older/2604.pdf, October, 1997.

[6] Adel, H., Vu, N. T., & Schultz, T. (2013). Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 206–211, Sofia, Bulgaria. Association for Computational Linguistics.

[7] Ando, R., & Zhang, T. (2005a). A High-Performance Semi-Supervised Learning Method for Text Chunking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05),* pp. 1–9, Ann Arbor, Michigan. Association for Computational Linguistics.\

[8]Auli, M., Galley, M., Quirk, C., & Zweig, G. (2013). Joint Language and Translation Modeling with Recurrent Neural Networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1044–1054, Seattle, Washington, USA. Association for Computational Linguistics.

[9]Auli, M., & Gao, J. (2014). Decoder Integration and Expected BLEU Training for Recurrent Neural Network Language Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 136–142, Baltimore, Maryland. Association for Computational Linguistics.

[10]Ballesteros, M., Dyer, C., & Smith, N. A. (2015). Improved Transition-based Parsing by Modeling Characters instead of Words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 349–359, Lisbon, Portugal. Association for Computational Linguistics.

[11]Bansal, M., Gimpel, K., & Livescu, K. (2014). Tailoring Continuous Word Representations for Dependency Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 809–815, Baltimore, Maryland. Association for Computational Linguistics.

[12]Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2015). Automatic differentiation in machine learning: a survey. arXiv:1502.05767 [cs].

[13]Bengio, Y. (2012). *Practical recommendations for gradient-based training of deep architectures*. arXiv:1206.5533 [cs].

[14]Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, *3*, 1137–1155.

[15] Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In *Proceedings of 1992 IEEE International Symposium on Circuits and Systems*, pages 2777-2780.

[16] Doya, K. and Yoshizawa, S. (1989). Adaptive neural oscillator using continuous-time backpropagation learning. *Neural Networks*, 2:375-385.

[17] Elman, J. L. (1988). Finding structure in time. Technical Report CRL Technical Report 8801, Center for Research in Language, University of California, San Diego.

[18]Fahlman, S. E. (1991). *The recurrent cascade-correlation learning algorithm*. In Lippmann, R. P., Moody, J. E., and Touretzky, D. S., editors, Advances in Neural Information Processing Systems 3, pages 190-196. San Mateo, CA: Morgan Kaufma