Determining Fake Facts in Social Media using various classification methods and Raspberry Pi

Monalisa Mallick and Arindum Roy
National Institute of Technology, Rourkela.

ABSTRACT

Fake facts have become a major problem in social media today. Whether it is via WhatsApp, Facebook or other social media sites, the propagation of fake facts is everywhere. With social media having become the go-to destination for news, the spread of fake facts and fake news is at an all time high. A lot of times, there are organizations who use social media to carry out their propaganda and spread their messages to the masses. In doing so, a large amount of fake facts are spread among the general public, and when these facts are religious or racial in nature they may lead to a lot of chaos and confusion in the society. In such a scenario, what our project aims to do is to create an Android app that can be used to determine whether a fact is true or fake. With a reputed data set, [1] , we will use an average of various classification algorithms to learn the distinction between fake facts and real facts, and this model is stored on a Raspberry Pi server which can be accessed via an android app so that for any article given to it as input it can classify it as either fake or real.

KEYWORDS

Fake facts, Classification, Social media, Android application, Perception, Trust.

I. INTRODUCTION

THE various sub-domains that have been used throughout the duration of this project include, various classification techniques like Logistic Regression, Support Vector Machines(SVM), Naive Bayes, and Random Forest classification[2]. These classification techniques were critical in determining the difference between the fake facts and the real facts. In today's scenario when social media has become the primary source of news for most of the people across the world, we believe that our product finds a broad spectrum of application in everyday use. The reasons for consumption of news on social media are mainly because the news is cheap and also because of just the sheer vastness of the news that is available. Everyday we receive a lot of messages, and a lot of fake facts are circulated in such messages. In such a scenario, it becomes difficult to check the authenticity of each message one by one which is why we have developed an android application that behaves like a chat-bot and does the work of verifying the fact for the user. We believe that this makes things a lot easier for the general community at large, by limiting the amount of fake news that people will believe. Moreover, people will also get to know of all possible types of fake facts that are spread in the society and the kinds of organizations that are responsible for spreading them.

II. RELATED WORKS

A. *Work done so far:*

THERE has been a lot of work by different bodies surrounding text analysis of fake news and facts and other related topics such as rumors or spam or sentiment analysis about the deceptive opinions regarding product reviews. For example, [3] proposes the use of profile alignment compatibility as an indicator of truthfulness in product reviews. It defines two types of compatibility between product profiles, and designed a methodology to tackle them by extracting aspects and associated descriptions from reviews. Over 20 years ago, Fogg and Tseng [4] described credibility as a perceived quality composed of multiple dimensions. Throughout the paper they have discussed users evaluating the credibility of computer products, a general phrase we use to describe many types of computer devices, systems, and

applications. Gupta [5] presented a semi-supervised ranking model for scoring tweets according to their credibility which is used in TweetCred, a real-time system that assigns a credibility score to tweets in a user's timeline. Core Linguistic analysis have also been combined with traditional classifiers and statistical analysis to label an article as true or false [6,7]. Ma Wong[8] proposed a novel approach to automatically identify rumors on microblogging websites. We develop a Dynamic Series-Time Structure model which explores the variation of various social context features over time. Linguistic and network-based approaches based paper[9] provides researchers with a map of the current landscape of veracity (or deception) assessment methods, their major classes and goals, all with the aim of proposing a hybrid approach to system design. In her recent blog McClure[10] discusses the five important characteristics to distinguish fake news from real news.

B. *Limitations:*

ALL the work highlighted here rely on hand-crafted features where they rely on the detection and extraction of local characteristics, domain knowledge, or considering a normal distributional assumption, offering a more general modeling of the data. The usage of mainly linguistic features to build a recurrent network was proposed by Ma et al [11]. It aims to expand the boundary of hand-crafted features. Some results are presented suggesting that template-based methods are significantly more accurate than simpler alternative methods. Sometimes lack of the labels can make it hard to evaluate the model on the particular task. Various methods tend to exploit supervised machine learning that are based on wide range of features, but they just consider the overall statistics. The variation of the information provided, over a time are overlooked. These trends have not gone unnoticed and over the last years, regular efforts have been made at developing and improving the methods so that they could take advantage of the available data for a longer time. To improve the efficiency in detection, it is important to study the changes of the overall as well as the individual properties along its life cycle.

III. Motivation and Objectives

A. *Motivation :*

1) Almost all the people have become predominantly dependent on social media as their primary source of information. But propagation of false facts have also made their way through it. To deal with this, there is a requirement of a proper fact checker to distinguish legitimate and the fraudulent facts.

2) The implementation of a model would help us check the authenticity of each message, one by one and return the correct output. This can avoid any undesired situation that can give rise to any political, administrative or even serious problems like communal conflicts.

B. *Objective:*

1) Sometimes the search results fail to display the correct information, which will be taken care of by our model that retrieves output after a proper understanding and processing of the facts.

2) Being able to distinguish between the real and fake facts easily and instantaneously, we tend to develop the product that can be proved beneficial to the entire society.

IV. Methodology

THE product that has been developed by us, has three major stages under which it can be categorized. First and foremost, the most important step was the implementation of the machine learning algorithms on the data set, to train a model, so that it could be used to detect fake facts.

In the second part of the project, we had to set up a server with the Raspberry Pi, with the trained model in the back-end of the server. The server had to setup as such, so that any device which was on the same network as the server could send a server request to the Raspberry Pi with the fact to be checked as the

argument, and get an output as a result.
In the final stage of the project, we created an Android app with a simple yet elegant to use user interface, which could verify the fake facts that the user inquired about, directly by sending requests to the Raspberry Pi. In the following paragraphs, we will go into details about each stage of the project and how they were implemented.

A. Training the machine learning algorithm to distinguish between fake and real facts:

THE data set we have used to design our machine learning algorithm is the LIAR data set [1], that contains different data files for the purposes of training, testing and validation. The original data set had a total of thirteen different variables, but for the purpose of simplicity during the course of this product development, we have considered only two variables, namely the statement that is to be verified and the True/False label. In the data pre-processing part of the project, certain pre-processing functions like tokenization were carried out on the data. Also data quality checks were carried out to ensure that null or missing values of data dont contribute to confusing the machine learning algorithm. In the python file Classifier.py we have used several classification techniques like Logistic Regression, Linear SVM (Support Vector Machines), Naive Bayes and Random Forest classification techniques from the python library scikit-learn []. After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification. We have performed parameter tuning on these algorithms and chosen best performing parameters for these classifier. Finally, selected model was used for fake news detection with the probability of truth, being calculated as well. The final model was saved in a file called finalmodel.sav which was called by a python file called prediction.py, which asked the user for a fact as input at the command line and displays whether the given fact is true or false along with the truth probability of the given statement.

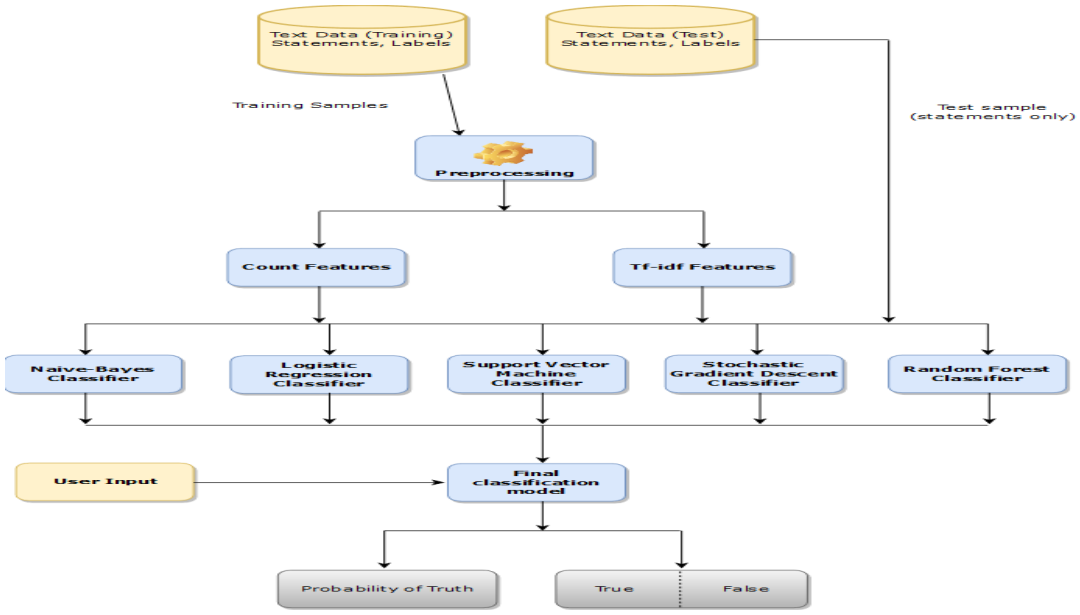

Fig. 1. **Process flow Diagram**

B. Setting up the server on the Raspberry Pi with the trained model in the back-end:

IN the second part of the project we have taken a Raspberry Pi Model 3B , as the server for our product. The finalmodel.sav file along with the prediction.py file was kept in the backend of the server that was setup using Apache on Raspberry Pi. A PHP file was made which accepted any URL requests to the server and passed the arguments received with the GET request, to the python file prediction.py. It was also responsible for echoing the output of the python file to the user.

C. Developing the android application:

IN the third part of the product, an android app was made using Android Studio. The backend of the app was written using Java and the front end of the app was written using XML(eXtensible Markup Language). A library called Volley was used to handle server requests made by the android app to the server that is connected on the same network as the phone, i.e. the Raspberry Pi server. Once the request is sent, the result that is obtained from the server is displayed on the app screen for the user to see.

```
// Instantiate the RequestQueue.
        RequestQueue queue = Volley.newRequestQueue( context: this);
        String url =("http://192.168.43.189/pdlab/main.php?username="+mTextView.getText());

// Request a string response from the provided URL.
        StringRequest stringRequest = new StringRequest( com.android.volley.Request.Method.GET, url,
                (response) → {
                        Intent myintent=new Intent( packageContext: LoginActivity.this, PopUp.class).putExtra( name: "mnls", response.substring(0)
                        startActivity(myintent);
                    //   mTextView.setText("Response is: "+ response.substring(0));

                }, (error) → {
                Intent myintent=new Intent( packageContext: LoginActivity.this, PopUp.class).putExtra( name: "mnls", value: "That didn't work!"+erro
                startActivity(myintent);
               // mTextView.setText("That didn't work!"+error.toString());
        });

// Add the request to the RequestQueue.
        queue.add(stringRequest);


    }
```

Fig. 2. **Volley request made by the Android application**

V. RESULTS

In the first part of our project we trained the model finalmodel.sav and accessed it using the python file prediction.py. The output that was obtained on executing the python file at the terminal, was as follows:

```
asliroy@roys-predator: ~/Fake_News_Detection
asliroy@roys-predator:~/Fake_News_Detection$ python3 -W ignore prediction.py
Please enter the news text you want to verify: Akshay Kumar is a terrorist.
You entered: Akshay Kumar is a terrorist.
The given statement is  False
The truth probability score is  0.469187612795
asliroy@roys-predator:~/Fake_News_Detection$ python3 -W ignore prediction.py
Please enter the news text you want to verify: Donald Trump is the President of
the United States of America.
You entered: Donald Trump is the President of the United States of America.
The given statement is  True
The truth probability score is  0.646425857778
asliroy@roys-predator:~/Fake_News_Detection$ python3 -W ignore prediction.py
Please enter the news text you want to verify: Barack Obama is a Republican.
You entered: Barack Obama is a Republican.
The given statement is  False
The truth probability score is  0.238516805488
asliroy@roys-predator:~/Fake_News_Detection$ python3 -W ignore prediction.py
Please enter the news text you want to verify: The World Trade Centre attacks to
ok place in 2001.
You entered: The World Trade Centre attacks took place in 2001.
The given statement is  True
The truth probability score is  0.531010010402
asliroy@roys-predator:~/Fake_News_Detection$
```

Fig. 3. **Output of the Python file as executed at the terminal**

Thus it is clear, that the model has a good distinction between which news is fake and which true.

In the second part of the product development we kept the the python file at the backend of an apache server and tried to call it using a GET request. The output of this segment was as follows:

Fig. 4. **GET request made via the PHP file to prediction.py**

In the third and final part of the project, we made the android app which sends a request to the server, that is on the same network as the phone through which the app is executing. The results were as follows:

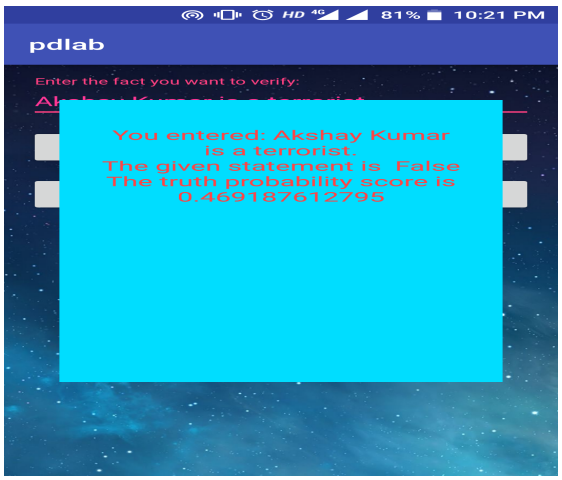

Fig. 5. **Output displayed on the android application screen**

VI. Conclusion

In our paper, and throughout the duration of our project, we have considered the importance of right to truthfulness of individual and have developed a model to retrieve the correctness of a particular fact along with its probability. The model can still be further modified by adding more and more features to be considered from the data set that we have. There could be extensions for future work. For instance, we have used two variables from the original data set in our method to distinguish between real and fake facts. The other variables can be added in order to add some more complexity and enhance the features. Any false positives that have occurred during the implementation of our project could be avoided thus giving us a more accurate result.

The classification techniques at some points of time, showed some examples of false positives, where statements which aren't true, were also labelled as true. If the algorithm is fine tuned even more, and more and more features are taken into consideration while preparing the model, then this problem will also go away.

In the future, such a product can be used to detect falsities in social media and also in specialized scenarios, wherever a proper data set is available.

References

[1] Wang and W. Yang, *Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection* [1705.00648], 2017.

[2] Singh, Vikram Midha, Neha, *A Survey on Classification Techniques in Data Mining.* International Journal of Computer Science Management Studies 22315268, 2016.

[3] Vanessa Wei Feng and Graeme Hirst, *Detecting Deceptive Opinions with Profile Compatibility.* In IJCNL,. 338346, 2013.

[4] B. J. Fogg and H. Tseng. *The elements of computer credibility.* In CHI 99: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 8087, New York, NY, USA, 1999.

[5] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier, *Tweetcred: Real-time credibility assessment of content on twitter.* In International Conference on Social Informatics. Springer, 228243, 2014.

[6] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete, *Information credibility on twitter.* In Proceedings of the 20th international conference on World wide web. ACM, 675684, 2011.

[7] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung, *Rumor Detection over Varying Time Windows.* PLOS ONE 12, 1 (2017), e0168344, 2017.

[8] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong, *Detect rumors using time series of social context information on microblogging websites.* In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 17511754, 2015.

[9] Niall J Conroy, Victoria L Rubin, and Yimin Chen, *Automatic deception detection: Methods for finding fake news.* Proceedings of the Association for Information Science and Technology, 52(1):14, 2015.

[10] Laura McClure,*How to tell fake news from real news.* blog.ed.ted.com/2017/01/12/how-to-tell-fake-news-from-real-news/ , 2017.

[11] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha, *Detecting rumors from microblogs with recurrent neural networks.* In Proceedings of IJCAI, 2016.