Fake Profile Detection Using Deep Learning

Yadnika Birari, Abhishek Chaudhuri, Prof.Madhura Vyawahare, Sanjana Darne

Pillai College Of Engineering Panvel, Maharashtra
Pillai College Of Engineering Panvel, Maharashtra
Pillai College Of Engineering Panvel, Maharashtra
Pillai College Of Engineering Panvel, Maharashtra

Submitted: 01-06-2021 Revised: 14-06-2021 Accepted: 16-06-2021

ABSTRACT—These days each and every person has access to the internet, this means that most of the internet users in today's date will be unable to differentiate between what's safe and what's threatening to them. As the number of internet users are increasing day by day the users of OSN (online social networks) are also increasing which is directly proportional to the increase in all kinds of fake and malicious attacks on the users of these online social networks. On top of that the open nature of these online social networks have made them vulnerable to various attacks including the sybil attacks. As the online social platforms are growing more and more popular the identity clone attacks that aim at creating fake identities for malicious purposes are also growing directly proportional to it. In this system we will make the use of deep learning to check if the twitter id provided to the system is fake or genuine. And for doing that we will make the use of RNN LSTM in deep learning and the string comparators for the comparison of the two different strings.

Keywords: OSN(online social networks), sybil attacks, fake identities, twitter, string comparators, Deep Learning.

I. INTRODUCTION

In this era the online social networks are considered to be the most popular platforms on the internet. It plays a major role for the users of the internet to perform their everyday actions such as news reading, content sharing, messages posting, product reviews and event discussions etc. The massive amounts of personal data of the users coupled with the open nature of these online social networks have made these online social networks vulnerable to various attacks including the sybil attacks. As the online social networks are becoming increasingly popular the identity clone attacks that aim at creating fake identities for malicious purposes are also becoming a growing concern these days. There are multiple types of spammers that coexist in the online social networks.

We are here proposing a system that can be used to detect the fake profiles present on the online social network (Twitter). This system will use deep learning to generate a base tweet and then using the string comparators it will compare the different tweets and in the end we will have the results if the Twitter id is genuine or it's fake. Thus this system will be of great use for the people as well as the host of the social media service.

II. LITERATURE SURVEY

The authors Sarah Khaled, Hoda M. O. Mokhtar came up with solving the problems on fake profile detection in social media platforms. This particular approach of identifying fake social media profiles was classified into the approaches aimed at analysis of the individual account and the approaches capturing the activities spanning in a large sample of accounts. The classification of these profiles based on their features made the use of several machine learning algorithms. [1]. Binghui Wang, Le Zhang proposed a system called SybilBlind which is a structure-based framework that is used to detect sybils in the social media platforms without a manually labeled training set. The evaluation is that the SybilBlind both theoretically and empirically, as well as compared it with Sybil detection methods that we adapt to detect Sybils when no manually labeled training sets are available. Their empirical results demonstrated the superiority of SybilBlind over the adapted methods. [2]. Mohammadreza M, Mohammad Eb. has proposed a model that makes the use of a resampling approach. The resampling approach means changing the distribution of training sampling sets by making the required changes to the data that is by processing the data. Balancing the datasets is one of the approaches used towards improving the class efficiency. This particular system also showed the use of principal component analysis. The basic idea of principal component analysis (PCA) is one of the multivariate classical methods and perhaps the most ancient and most popular one. Mostly all these machine learning methods train the classifiers using the machine learning algorithms. Attribute similarity, network friend similarity and IP address analysis are some of the social network attributes on which the classifiers are based[3]. On the same line we have developed a system to give better performance for detecting fake profiles.

III. ALGORITHMS

In order to identify the fake profiles, the Recurrent Neural Networks along with its various algorithms have been implemented to compare the strings and their values have been calculated in mean, algorithms used are: the levenshtein distance, dice's coefficient and Long short term memory (LSTM).

Recurrent neural networks:

As humans cannot understand the meaning of a word or sentence without its previous information. Every sentence is to be linked with the previous sentence in order to understand the full text. Similarly, the traditional neural networks have been facing issues and are unable to act like this. This is where the Recurrent Neural Networks helps and addresses the value. This type of network is with loops which allows the data to persist.

Long Short Term Memory(LSTM):

LSTM networks are a type of neural network which is capable of learning order dependencies in sequence of prediction problems. This usually helps in big problem domains such as machine translations, speech recognitions and many more complex issues. LTSMs is the complex area of deep learning. LSTMs are designed to avoid the occuring long-term dependency problems and remembering information is in their behavior and it does have to struggle to learn.

The Levenshtein distance:

It is a string metric for measuring the differences between strings having two sequences. The insertion, deletion or substitution between the two words is calculated using levenshtein distance which is required to change one word into the other. It may also refer for editing distance even if it also denotes a large cluster of distance metrics. It is generally related to pairing wise string alignments. Ituses:

- a single distance vector instead of using a matrix.
- a loop unrolling on the loop on its outer side.
- by removing common prefixes andpostfixes.
- minimizing the comparisons. Dice's coefficient:

It measures how similar two sets are with each other. Here, it can be used to check the similarities between two strings in terms of the number of similar bigrams that is the pair of adjacent letters in a string.

IV. PROPOSED SYSTEM

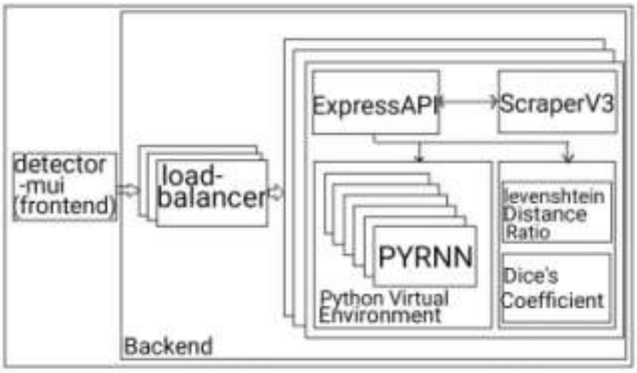


Fig.1. Flow of the project

The above mentioned system has three phases depending on the following:-

Phase1:- Scraping the data

In this section the system scraps the Twitter data using the Twitter metadata api. This is the first phase of the system where the user of the system puts in the Twitter id that has to be analysed and then the system starts scrapping the Twitter id for all possible tweets made by that id in the past few months.

Phase2:- Generating the base tweet

In this section the system creates a child process and the whole text generation process takes place in a Python virtual environment. The text generation that is the generation of the base tweet is done using deep learning. Here we make the use of LSTM RNN and generate a base tweet which has all the possible characteristics of the tweets that we have scrapped in the phase 1 of the system. After generating the base tweet we hop on to the 3rd phase of the system.

Phase 3:- Comparing the strings

In this section the system compares the two different strings in two different ways. First of all we get the base tweet that was generated using deep learning and then compare it with a randomly picked tweet from the dataset of tweets scrapped in phase one. Now the comparison is done using the string comparators and in this system we have made use of two string comparators which are the Levenshtein distance ratio and the Dice's coefficient. At first we perform the Levenshtein distance ratio and find out the random result and the average result then similarly we perform the Dice's coefficient and find out the random result and the average result and in the end to make the result more accurate we take the average of all the four results.

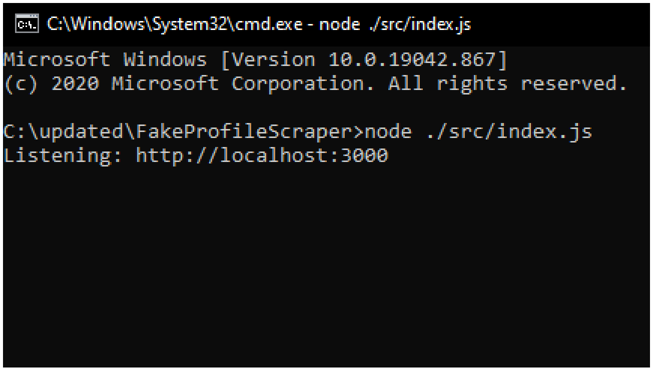
V. IMPLEMENTATION DETAILS AND RESULTS ANALYSIS

For implementation we have used ‘svelte application’ to show the result in our application based website.

1. Back-end.

Command to start Back-end by scraping the data from Twitter api in real time by using a scraping algorithm for comparing the tweets.

Fig.2. Back-end

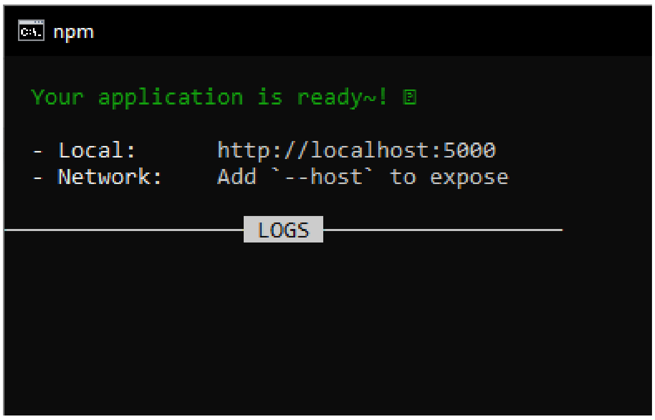


```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\updated\FakeProfileScraper>node ./src/index.js
Listening: http://localhost:3000
```

2. Front-end.

Command to start Front-end and use a generated localhost id to start the svelte application.



```
npm

Your application is ready~! 

- Local:      http://localhost:5000
- Network:    Add `--host` to expose

LOGS
```

Fig.3. Front-end

3. The home page.
Svelte application is used to show the results of our application based website which has textbox to enter the twitter username.

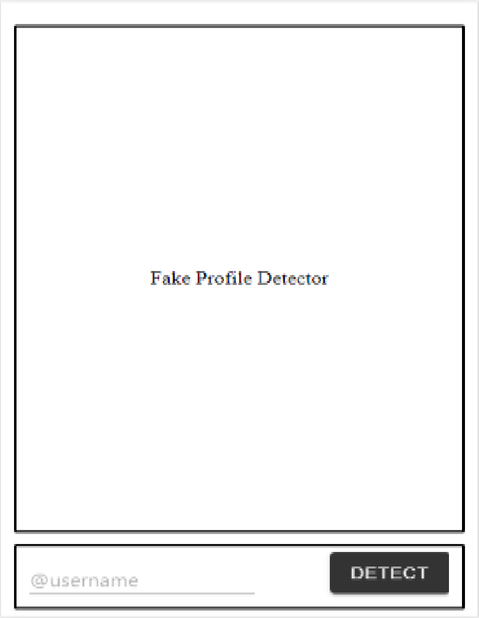
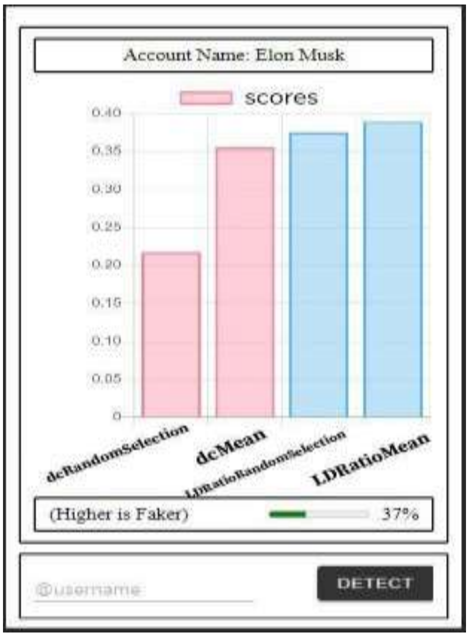


Fig.4. The home page

4. The result of a real account user.

Entering the username of a real account and the result calculated using the Random selection and mean of the Dice's Coefficient and Levenshtein Distance in percentage. From Figure 5 we can understand that based on the score we can conclude that profile is a real profile. As we can see in the FIgure 5 shows the low percentage when we pass the id "@elonmusk" it shows 37% after calculations.

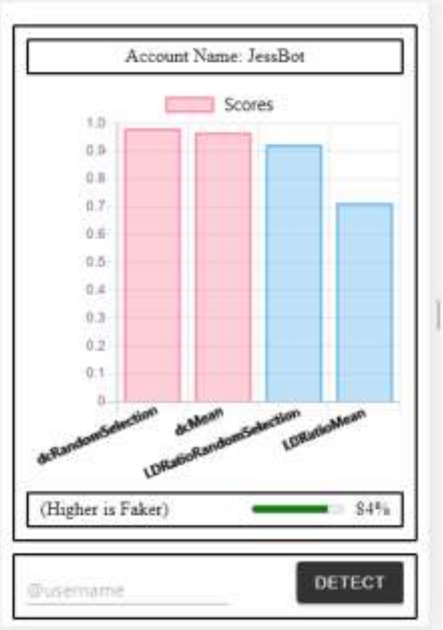

Account Name: Elon Musk

(Higher is Faker) 37%

Fig.5. Result of Real Account User

5. The result of a bot account user.

Entering the username of a bot account and the result calculated using the Random selection and mean of the Dice's Coefficient and Levenshtein Distance in percentage. From Figure 6 we can understand that based on the score we can conclude that the user is bot. As we can see in the FIgure 6 shows the low percentage when we pass the id "@bot_of_jess" it shows 84% after calculations.


Account Name: JessBot

(Higher is Faker) 84%

Fig.6. Result of Bot Account User.

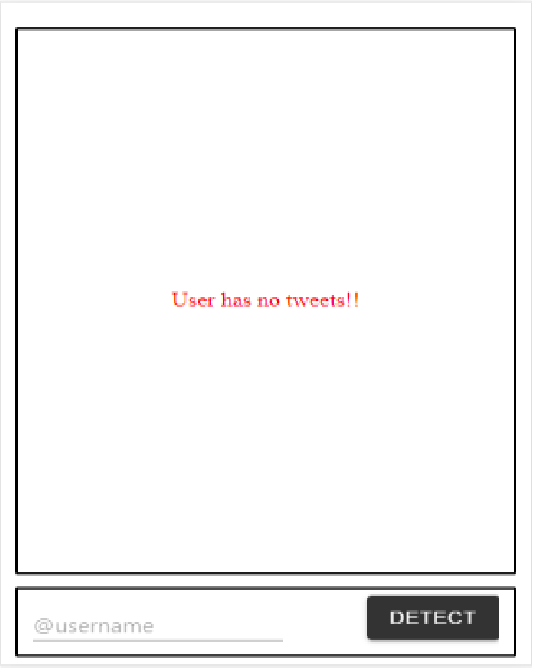
6. Result of a user with no tweets.

If the user has not tweeted anything, then the model is capable of displaying the result as 'user has no tweets'. As identifying real, fake or bot accounts is very dependent on tweets posted by users. Identifying this attribute plays a vital role in fake profile detection. Figure 7 shows the result when we pass the id "@YadnikaB" without any tweet.

7. Result of a user if the user account is suspended.

Displaying the result as ‘user has been suspended’ if the user account is suspended by twitter then it is also identified by our model. As we can see in the figure 8 for user id “@DarneSanjana7” it displays the user is currently suspended.

Fig.8. Result of Suspended User Account



User has no tweets!!

VII.CONCLUSION

In this paper, we presented a Fake Profile Detection System using Deep Learning; our project detects the fake or bot twitter profiles by using deep learning algorithms. This project can help the social media platforms hosts as well as the social media users to be protected from all the fake profile related threats.The future scope of the system is to make it more reliable and to include more characteristics to determine the genuineness of the profile.

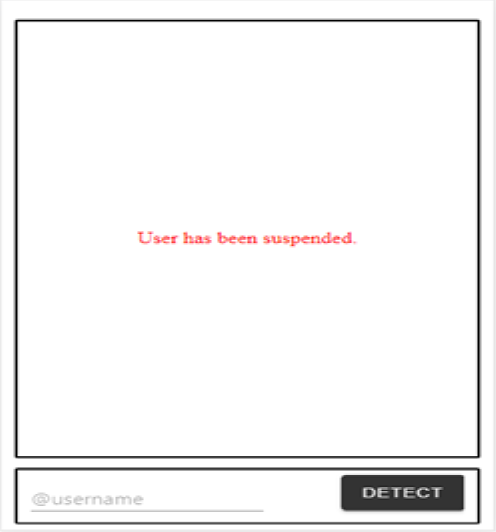
FUTURE SCOPE

There are many features that can be included in this project such as:

II. The future work concentrates on replacing more easy algorithms to detect fake user accounts such as replacing LSTM-Rnn with Transformers.

III. To apply the algorithms on different social media platforms to identify fake users.

IV. With proper accuracy of the result, this project aims to help cyber security branches.



User has been suspended.

Fig.7. Result of User with no tweets

REFERENCES

[1] Yasyn Elyusu, Zakaria Elyusu, and M'hamed Ait Kbir, "Social Networks Fake Profiles Detection Using Machine Learning Algorithms," Faculty of Sciences and Technologies, Tangier, Morocco,In book: Innovations in Smart Cities Applications Edition 3 (pp.30-40), 10.1007/978-3-030-37629-1_3(2019).

[2] Sarah, M. O. Mokhtar,Neamat El-Tazi, "Detecting Fake Accounts on Social Media "Faculty of Computers and Information, Cairo University,Cairo Egypt 2018 IEEE International Conference on Big Data (Big Data), 10.1109/BigData.2018.8621913(2018).

[3] Binghui Wang, Le Zhang, and Neil Zhenqiang Gong ECE , "SybilBlind: Detecting Fake Users in Online Social Networks without Manual Labels", Department, Iowa State University(2018).

[4] Mohammadreza Mohammadrezaei, Mohammad Ebrahim Shiri ,and Amir Masoud Rahmani, "Identifying Fake Accounts on Social Networks Based on Graph Analysis and Classification Algorithms",Computer Science, University of Human Development, Sulaymaniyah, Iraq, Content published prior to 2017 is hosted on the Wiley Online Library, DOI: 10.1155/2037(2017).

[5] Dr. Sanjeev Dhawan, Ekta, "Implications of Various Fake Profile Detection Techniques in Social Networks", UIET, Kurukshetra University, 136119, Kurukshetra, Haryana,India, February 2016IOSR Journal of Computer Engineering 02(02):49-55, 10.9790/0661-15010020249-55 (2016).

[6] Shalinda Adikari and Kaushik Dutta, "Identifying Fake Profiles in LinkedIn", PACIS Proceedings, AISeL, PACIS 2014 Proceedings. 278, https://aisel.aisnet.org/pacis2014/278/ ,(2014).

[7] Dr. S. Kannan, Vairaprakash Gurusamy, "Preprocessing Techniques for Text Mining", UEIS, New Delhi, India, 2019 JETIR May 2019, Volume 6, Issue 5 , www.jetir.org (ISSN-2349-5162) ,(2015).

[8] Thomas, Kurt, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. "Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse." Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13), pp. 195-210. (2013).