

USE ARTIFICIAL NEURAL NETWORKS TO IDENTIFY FAKE PROFILES X_0005

¹Omprakash, ²Y Gangadhar, ³Shobharani Kadhiri, ⁴V Bhargavi

¹²³Assistant Professor, Brilliant Institute of Engineering & Technology,
Abdullapurmet(V&M) Ranga Reddy Dist-501505

⁴UG Scholar, Department of CSE, Brilliant Institute of Engineering & Technology,
Abdullapurmet(V&M) Ranga Reddy Dist-501505

ABSTRACT

We use machine learning, namely an artificial neural network to determine what are the chances that Facebook friend request is authentic or not. We also outline the classes and libraries involved. Furthermore, we discuss the sigmoid function and how the weights are determined and used. Finally, we consider the parameters of the social network page which are utmost important in the provided solution. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse, is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media [1]. Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new photos, likes, dislikes, and tag other users in content posted, Facebook makes revenue via advertisements and data. More specifically, the average American user generates about \$26.76 per quarter [2]. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Facebook and Twitter. They can also target banks and other financial institutions.

INTRODUCTION

In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media [1]. Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new

photos, likes, dislikes, and tag other users in content posted, Facebook makes revenue via advertisements and data. More specifically, the average American user generates about \$26.76 per quarter [2]. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and

often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Facebook and Twitter. They can also target banks and other financial institutions.

II. EXISTING SYSTEM

1. Malicious users create fake profiles to phish login information from unsuspecting users. A fake profile will send friend requests to many users with public profiles. These counterfeit profiles bait unsuspecting users with pictures of people that are considered attractive. Once the user accepts the request, the owner of the phony profile will spam friend requests to anyone this user is a friend.

2. The fake profile's contents typically have links that lead to an external website where the damage happens. An unaware curious user clicking the bad link will damage their computer. The cost can be as simple as catching a virus to as bad as installing a rootkit turning the computer into a zombie. While Facebook has a rigorous screening to keep these fake accounts out, it only takes one fake profile to damage the computers of many.

Disadvantages

1. Analyzing of account data becomes more critical for humans
2. Detecting of fake account take time

III. PROPOSED SYSTEM

1. In our solution, we use machine learning, namely an artificial neural network to determine what the chances that a friend request is authentic are or not.

2. We utilize Microsoft Excel to store old and new fake data profiles. The algorithm

then stores the data in a data frame. This collection of data will be divided into a training set and a testing set. We would need a data set from the social media sites to train our model.

3. For the training set, the features that we use to determine a fake profile are Account age, Gender, User age, Link in the description, Number of messages sent out, Number of friend requests sent out, Entered location, Location by IP, Fake or Not. Each of these parameters is tested and assigned a value. For example, for the gender parameter if the profile can be determined to be a female or male a value of (1) is assigned to the training set for Gender. The same process is applied to other parameters.

We also use the country of origin as a factor

Advantages :-

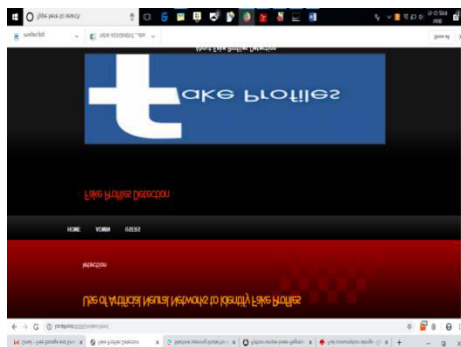
1. Analyzing large amount of data becomes very easy
2. No human power requires
3. Using ANN algorithm easy to identify the status of account weather the account fake or genuie.

To deploy and test the application on a Django server, start by setting up the Django environment and placing your application code in the appropriate directory. Launch the server using the `'python manage.py runserver'` command, making it accessible at `'http://localhost:8000/'`. Open your web browser and navigate to this URL to access the main page of the application. From there, click on the 'ADMIN' link to reach the admin login screen. Log in using `'admin'` for both the username and password to gain administrative access.

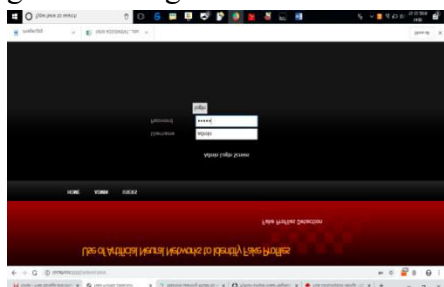
Once logged in, select the 'Generate ANN Train Model' link to initiate the training of the Artificial Neural Network (ANN) on your dataset. Monitor the server console to

track the training process and view details such as accuracy. Ensure that the ANN achieves a training accuracy of 98%, reflecting effective learning from the data. After training, click on the 'View ANN Train Dataset' link to review and scroll through the details of the training dataset. Log out from the admin panel and navigate to the 'User' section to test the predictive capabilities of the application. Enter sample test account details into the provided fields, using records such as `10, 1, 44, 0, 280, 1273, 0, 0`, `10, 0, 54, 0, 5237, 241, 0, 0`, `7, 0, 42, 1, 57, 631, 1, 1`, and `7, 1, 56, 1, 66, 623, 1, 1`. Submit these details and observe the results to see whether the ANN classifies each account as genuine or fake. This process verifies that the application functions correctly and that the ANN model is effective in its predictions.

Deploy this application on DJANGO server and then run in browser enter URL as <http://localhost:8000/index.html> to get below screen

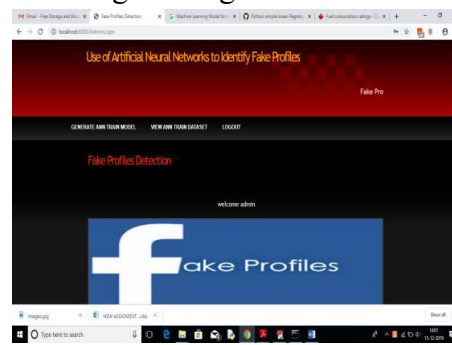


In above screen click on 'ADMIN' link to get below login screen

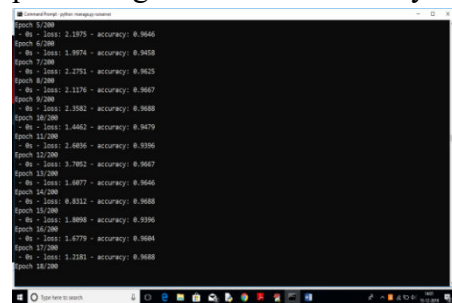


In above screen enter admin and admin as username and password to login as admin.

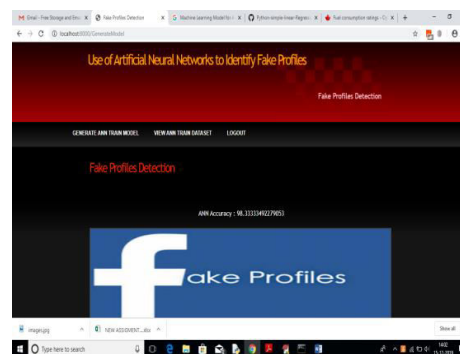
After login will get below screen



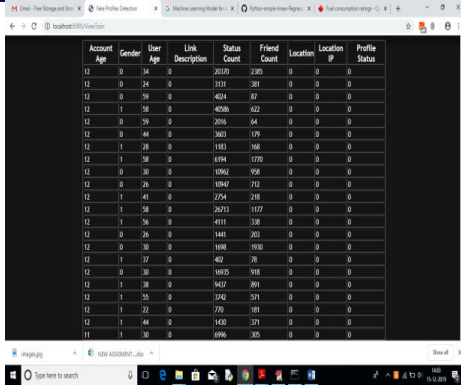
In above screen click on 'Generate ANN Train Model' to generate training model on dataset. After clicking on that link you can see server console to check ANN processing details with accuracy



In above black console we can see all ANN details.

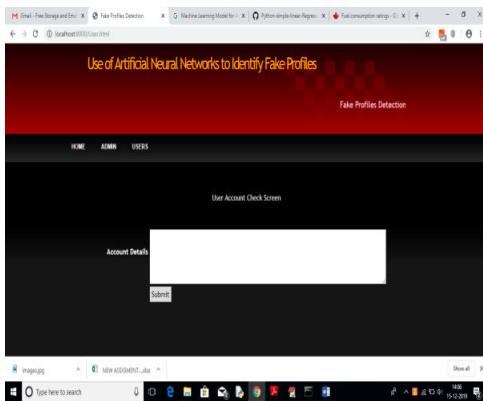


In above screen we can see ANN got 98% accuracy to train all Facebook profile. Now click on 'View Ann Train Dataset' link to view all dataset details

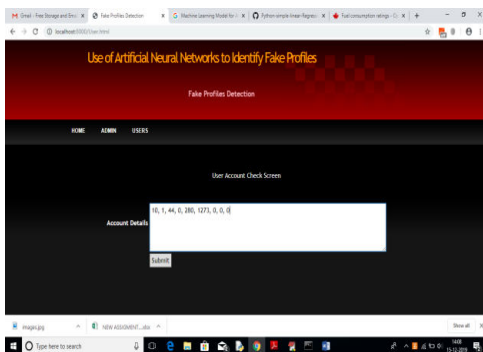


Account	Age	Gender	User	Link	Status	Friend	Location	Location	Profile
Age				Description	Count	Count	#		Status
12	0	34	0		2030	7305	0	0	0
12	0	24	0		3131	381	0	0	0
12	0	59	0		4024	87	0	0	0
12	1	50	0		4056	622	0	0	0
12	0	59	0		2929	64	0	0	0
12	0	44	0		3602	179	0	0	0
12	1	28	0		1183	168	0	0	0
12	1	58	0		6394	1790	0	0	0
12	0	20	0		1902	150	0	0	0
12	0	26	0		10947	712	0	0	0
12	1	41	0		2754	218	0	0	0
12	1	58	0		2673	1177	0	0	0
12	1	50	0		4111	330	0	0	0
12	0	26	0		1445	200	0	0	0
12	0	30	0		1698	1930	0	0	0
12	1	37	0		402	78	0	0	0
12	0	30	0		1695	918	0	0	0
12	1	28	0		1432	491	0	0	0
12	1	55	0		1242	271	0	0	0
12	1	22	0		770	181	0	0	0
12	1	44	0		1430	371	0	0	0
11	1	30	0		696	305	0	0	0

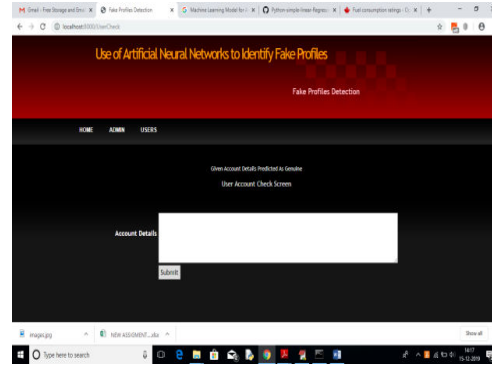
In above screen we can see all train data and scroll down to view all records. Now ANN train model is ready and you can logout and click on ‘User’ link to get below screen.



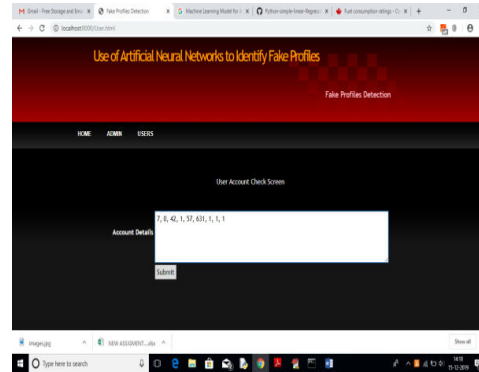
In above screen enter some test account details to get prediction/identification from ANN. You can use below records to check
 10, 1, 44, 0, 280, 1273, 0, 0
 10, 0, 54, 0, 5237, 241, 0, 0
 7, 0, 42, 1, 57, 631, 1, 1
 7, 1, 56, 1, 66, 623, 1, 1



For above input will get below result



In above screen we can see the result predicted as genuine account



For above account details we get below result

IV.CONCLUSION

we use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic or not. Each equation at each neuron (node) is put through a Sigmoid function. We use a training data set by Facebook or other social networks. This would allow the presented deep learning algorithm to learn the patterns of bot behavior by back propagation, minimizing the final cost function and adjusting each neuron's weight and bias.

V. REFERENCES

1. Alowibdi, J. S., Alrubaian, M., Al-Khalifa, H. S., & Ngai, E. C. H. (2016). Detecting spammer profiles in social networks using linguistic features. *Journal of Universal Computer Science*, 22(4), 512-531. <https://doi.org/10.3217/jucs-022-04-0512>
2. Figueroa, A., & Neumann, G. (2014). Identifying fake profiles in social networks using the long tail distribution. *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, 459-468. <https://doi.org/10.1145/2661829.2662061>
3. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2015). Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems*, 80, 56-71. <https://doi.org/10.1016/j.dss.2015.09.003>
4. Varol, O., Ferrara, E., Davis, C., Menczer, F., & Flammini, A. (2017). Online human-bot interactions: Detection, estimation, and characterization. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1), 280-289. <https://ojs.aaai.org/index.php/ICWSM/article/view/14878>
5. Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Prentice Hall.
6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
7. Facebook AI Research. (2020). *Deep learning for detecting fake accounts on social media platforms*. Facebook Research. <https://research.fb.com/publications/deep-learning-for-detecting-fake-accounts>
8. Nguyen, N. P., Yan, G., Thai, M. T., & Eidenbenz, S. (2011). Detecting and characterizing social spam campaigns. In *Proceedings of the 10th Annual Conference on Emerging Networking Experiments and Technologies* (pp. 1-12). ACM. <https://doi.org/10.1145/2079296.2079307>