Use of Artificial Neural Network to Identify Fake Profiles on social media

AISHWARYA1, BHAGYAVANTI BIRADAR[2], BIRADAR AISHWARYA RAVINDRA[3], AKSHATA PATIL[4], PROF. MASRATH BEGUM[5]
[1, 2, 3, 4, 5] *Guru Nanak Dev Engineering College, Bidar*

Abstract- In this project, 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.

I. INTRODUCTION

In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media . 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 . 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.

There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Facebook had a data breach which affected about 50 million users . Facebook provides a set of clearly defined provisions that explain what they do with the user's data . The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Facebook's built-in security features.

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.

The solution presented in this paper intends to focus on the dangers of a bot in the form of a fake profile on your social media. This solution would come in the form of an algorithm. The language that we chose to use is Python. The algorithm would be able to determine if a current friend request that a user gets online is an actual person or if it is a bot or it is a fake friend request fishing for information. Our algorithm would work with the help of the social media companies, as we would need a training dataset from them to train our model and later verify if the profiles are fake or not. The algorithm could even work as a traditional layer on the user's web browser as a browser plug-in.

II. RELATED WORK

Sybil rank was designed in late 2012, to efficiently identify fake profiles through a ranking graph-based system. The algorithm uses a seed selection method combined with early terminated random walks to propagate trust . Its computational cost is measured in O (nlogn). Profiles are ranked according to the number of interactions, tags, wall posts, and friends over time.

Profiles that have a high rank are considered to be real with fake profiles having a low rank in the system. Unfortunately, this technique was found to be mostly unreliable because it failed to take into account the possibility that real profiles can be ranked low and fake profiles can be ranked high.

Sarode and Mishra proposed a different approach which is a sequence of steps to detect fake profiles . They used the Facebook graph API tool to gain access to numerous profiles and wrote a script to extract the viewed information. Later on, this extracted information forms the attributes the classifier will use in their algorithm. First, the data is in JSON format, which is further parsed to a structured format (CSV) that is easier readable by machine learning techniques. These comma separated values will later make the classifier more efficient. The authors tried unsupervised and also supervised machine learning techniques. In this case, supervised machine learning techniques had a higher accuracy rate of almost 98%. For supervised machine learning, they split up the dataset into training and testing sets. They used 80% of the samples to train the classifier and the rest to test it.

After the algorithm runs, there is feedback provided to the profile, requiring it to submit identification to prove it is not a fake profile .

Profiles are processed on mass to extract features. Resilient Back Propagation algorithm in neural networks algorithm combined with support vector machines is used in the classification of fake profiles. Sybil Frame uses multi-stage level classification. Approaches include content-based and structure based. Content-based approach explores the dataset and extracts information used to calculate prior information about nodes and edges. Structure-based approach correlates nodes using Markov random field and loopy belief propagation which employs previous information. The content-based approach is used in the first stage of Sybil Frame and Structure-based approach is used in the second stage of Sybil Frame technique.

Clickstreams are analyzed, and Friend recommendations are examined in stage I. Vote Trust uses a voting based system that pulls user activities to find fake profiles using trust-based vote assignment and global votes total. It is considered as the first line of defense due to limitations which include real accounts that were already compromised being sold.

III. PROPOSED SYSTEM

In our solution, 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 to keep the results between the interval of 0.0 and 1.0. At the output end, this could easily be multiplied by 100 to give us the possible percentage that it is a malicious request. Our solution would be only one deep neural network, meaning it only has a single hidden layer.

Each input neuron would be a different, previously chosen feature of each profile converted into a numerical value (e.g., gender as a binary number, female 0 and male 1) and if needed, divided by an arbitrary number (e.g., age is always divided by 100) to minimize one feature having more influence on the result than the other.

The neurons represent nodes. Each node would be responsible for exactly one decision-making process. Each object has a weight and bias that in turn would help in the decision-making process. The output would be the possibility in the percentage that the friend request is not from a real person. Figure 1 shows the used neural network.

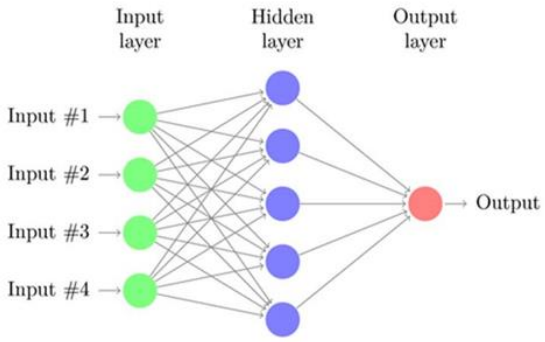

Figure 1: Neural Network.

We would need a training data set which would be provided by Facebook or other social network sites or

just web scraping given that we find enough fake profiles. This would allow our deep learning algorithm to learn the patterns of bot behavior by backpropagation, minimizing the final cost function and adjusting each neuron's weight and bias, changing the equations. In this paper, we outline the classes and libraries involved.

IV. IMPLEMENTATION

Our deep learning algorithm is written in the Python language. The used libraries are:

- Pandas
- NumPy
- MATLAB •
- theano
- scikit-learn
- Keras
- TensorFlow

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.

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. The top country with highest bot activity is China with the United States coming in as a strong second [8].

As stated before, there are different layers to the algorithm. For example, there are 128 nodes in the hidden layer. There is also an input layer and an output layer. With a single hidden layer is used, it is called a one deep machine learning algorithm. These layers are intended to mimic a neural network. In our case, it is named an artificial neural network or ANN [9]. This system can be used in AI programs to solve problems. It is often used in face recognition, pattern recognition and even in training virtual assistants (Siri, Alexa, etc.). This type of model is used to behave like the human brain. Different nodes would represent specific parameters; for example, there may be a node for the Age parameter and another one for the Gender parameter and so forth. Based on the inputs provided an output (decision) is produced. The inputs are directed to the hidden layer.

The data is to read using the readCSV() method. This is read into the training set. We use the dropna() method to clean up the data set. Using the correct parameters and formatting the data correctly is one of the most important things when building an ANN. We then convert the parameters such as Account Age, Gender, etc. to a numerical form somewhat like an enumerated data type. We convert the Account Age into weeks. We then compare the IP address parameter with the actual IP address of the profile. If it is a match, then a value of (1) is assigned to locationMatch. If it does not match, then the value of (0) is assigned to locationMatch. The same process is repeated for Gender.

We used the match() method to compare the link in the description with the training set's link in the description. A Boolean value of true or false is assigned to the url_found variable. If it is true, the value (1) is assigned to the Link in the description parameter. If it is false, the value (0) is assigned.

We then determine the Number of messages sent out parameter by dividing the number of messages sent by the age of the account. We then determine the Number of friend requests sent out parameter by dividing the Number of friend requests sent out by the age of the account. We use the Iloc() method to adjust the columns in the data set. We use this method for our input set and our outputs set.

Next, we split the input set and the output set in half. This will leave us with four different sets. The first input set is assigned to the input train. This variable contains the second half of the input set. The second input set is assigned to the input test. This variable contains the first half of the input set. The first output set is assigned to the output train. This variable

contains the second half of the output set. The second output set is assigned to the output test. This variable contains the first half of the output set.

We use the standardscaler() class [10]. This allows us to convert the data into a uniform distribution form so that it has a mean value of (0) and a standard deviation . We use fitTransform() method on the input train parameter and the transform() method on the input_set parameter. This converts the data to the intended distribution.

The next step is to use Sequential from the Keras library to create the model. We use the add() method of the Sequential class to activate the Sigmoid function and again to generate the output layer. Afterward, we compile the model using Stochastic Gradient Descent as an optimizer. The neural network is then fitted to our training set using the fit() method.

We have to test the accuracy of the input_test variable which contains one half of the input set. We do this using the predict() method. The result is then converted to a percentage. This result is stored in the output_pred variable, which is in turn stored in our database under a new column.

The input_test and the output_test variables are then passed as parameters to the evaluate() method. The result of the evaluate() method is assigned to score. The score is used to determine whether or not the profile is real or fake.

Libraries

Through the use of different libraries, we can easily make machine learning and data mining possible. The most common language in use for artificial intelligence today is Python, due to its popularity, compactness and the various ready-to-use libraries that are perfect for mathematical models.

One such library is Pandas, which is an excellent tool for data analysis . It can be used in a wide range of fields including academic and commercial domains such as finance, economics, statistics, analytics, etc. It can easily access and modify different datasets and optimize them for later use. Correct formatting of the datasets and finding the optimal vital features that are used later are crucial. Another library worth mentioning is NumPy. NumPy can be used for scientific computing and used primarily for multi-dimensional matrix multiplication as we are dealing with a large amount of numbers that are very dependent on each other.

A similar tool, MATLAB is often used to plot and visualize mathematical models for analysis or as Numpy, for matrix multiplications. The Theano library [is also ideal for evaluating mathematical expressions involving multidimensional arrays. For plotting and visualizing, data analysis and data mining, you can use the Scikit-learn (sklearn) machine learning library . It is built on top of the NumPy, SciPy, and matplotlib libraries. Scikit-learn features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, kmeans and DBSCAN. It was initially developed as a Google summer code project in 2007.

The two most essential libraries are Keras and TensorFlow [18]. Keras itself is capable of running on top of TensforFlow, CNTK, or Theano.

It enables fast experimentation and prototyping. Keras core structure is a model, a way to organize layers. It was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). The name Keras means horn is Greek.

Another viral library is TensorFlow [18], which was developed by Google Brain with Google's AI organization for internal use initially. It is now an open-source software library that is ideal for machine learning, mainly using neural networks.

CONCLUSION

In this paper, we use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic are 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 backpropagation, minimizing the final cost function and adjusting each

neuron's weight and bias. In this paper, we outline the classes and libraries involved. We also discuss the sigmoid function and how are the weights determined and used. We also consider the parameters of the social network page which are the most important to our solution.

REFERENCES

[1] https://www.statista.com/topics/1164/social-networks/

[2] https://www.cnbc.com/2018/01/31/facebook-earnings-q4-2017arpu.html

[3] https://www.cnet.com/news/facebook-breach-affected-50-millionpeople/

[4] https://www.facebook.com/policy.php

[5] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12). USENIX Association, Berkeley, CA, USA, 15-15.

[6] Akshay J. Sarode and Arun Mishra. 2015. Audit and Analysis of Impostors: An experimental approach to detect fake profile in an online social network. In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015 (ICCCT '15). ACM, New York, NY, USA, 1-8. DOI: https://doi.org/10.1145/2818567.2818568

[7] Devakunchari Ramalingam, Valliyammai Chinnaiah. Fake profile detection techniques in large-scale online social networks: A comprehensive review. Computers & Electrical Engineering, Volume 65, 2018, Pages 165-177, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2017.05.020.

[8] https://www.enigmasoftware.com/top-20-countries-the-most-cybercrime

[9] pages.cs.wisc.edu/~bolo/shipyard/neural/local.html

[10] https://stackoverflow.com/questions/40758562/can-anyone-explain-mestandardscaler

[11] https://pandas.pydata.org

[12] https://www.tutorialspoint.com/python_pandas/index.htm

[13] http://www.numpy.org

[14] https://www.mathworks.com/products/matlab.html

[15] http://www.deeplearning.net/software/theano/

[16] https://scikit-learn.org/stable/

[17] https://keras.io

[18] https://www.tensorflow.org