PREVENTING SOCIAL MEDIA PHISHING IN SOCIAL NETWORKS

Department of Information Technology
Vignan Institute of Information Technology University
Beside VSEZ, Duvvada, Vadlapudi post,Gajuwaka, Visakhapatnam - 530049
* vignaniit.edu.in

Abstract — In this paper, we employ machine learning, specifically a man-made neural network, to evaluate the probability that a social account is authentic or not. We also describe the relevant classes and libraries. We also examine the sigmoid function and the selection and application of the weights. Finally, we take into account the social network page's parameters, which are crucial to the offered solution.

I. INTRODUCTION

Facebook became the most popular social media platform in 2017 after reaching a total user base of 2.46 billion. Users' information is used by social media networks to generate cash. The usual user is unaware that when they use the service of a social media network, their rights are forfeited. Social media firms stand to earn greatly at the expense of the user. Facebook generates income from adverts and user data whenever a user posts new content, such as locations, images, likes, and dislikes. More specifically, the average American user produces around $26.76 every three months. When several users are involved, that sum soon grows.

In the current digital era, the growing reliance on technology has made the average person more susceptible to crimes like data breaches and potential identity theft. These attacks can happen suddenly and frequently without warning to those who have experienced a data breach. Social networks currently have little reason to strengthen their data security. These hacks frequently target Twitter and Facebook, among other social media platforms. Banks and other financial organisations will also be on their radar.

Social media platforms getting hacked seem to be a newsworthy topic every day. About 50 million Facebook users were impacted by a recent knowledge breach. Facebook outlines a number of clearly stated provisions that detail how they use user data. The policy does very little, if anything, to stop the ongoing invasion of privacy and security. The built-in security measures on Facebook appear to be overcome by the fake profiles.

Bots and phoney profiles represent the opposite threat of personal information being collected for illegal reasons. Bots are computer programmes that collect data about users without their knowledge. Web scraping is the term for this activity. Even worse, this action is legal. On social networks, bots are frequently concealed or seem as phoney friend requests to get private data.

This paper's proposed approach aims to highlight the risks posed by a bot that poses as a false social media presence. An algorithm would be available to implement this solution. Python is the language we decided to utilise. The algorithm would be prepared to determine whether a user is receiving a friend request from a legitimate person, a bot, or a phoney friend request that is gathering information. Since we might need a training dataset from the social media companies to build our model and then determine if the profiles are phoney or real, our algorithm would function with their help. The method might potentially be used as a standard layer on the user's web browser as a browser plug-in.

II. PROBLEM IDENTIFICATION

Malicious users make fake profiles to phish login data from clueless users. A fake profile will send friend solicitations to numerous users with public profiles. These fake profiles lure clueless users with pictures of individuals that are viewed as alluring.

When the user acknowledges the demand, the proprietor of the fake profile will spam friend solicitations to anybody this user is a friend.

The fake profile's items regularly have joins that lead to an outside site where the harm occurs. An ignorant inquisitive user tapping the terrible connection will harm their PC. The expense can be pretty much as basic as contracting an infection to as terrible as introducing a rootkit transforming the PC into a zombie. While Facebook has a thorough screening to keep these fake records out, it just takes one fake profile to harm the PCs of
many.

III. RELATED WORK

Sybil rank, a ranking graph-based method, was created in late 2012 to effectively identify fraudulent profiles. To spread trust, the algorithm combines an early terminated random walk technique with seed selection. The computational expense is expressed in O (nlogn). According to the quantity of interactions, tags, wall postings, and friends throughout time, profiles are ranked. High-ranking profiles are taken into account to be genuine, whereas low-ranking ones are deemed to be false. Unfortunately, it was discovered that this system was largely unreliable because it overlooked the possibility that actual accounts may be ranked poorly and false profiles could be ranked well.

A unique method for identifying bogus profiles was proposed by Sarode and Mishra. They created a script to retrieve the seen data and used the Facebook graph API tool to gain access to many profiles. This gathered data is later used to create the features that the classifier will incorporate into their algorithm. The information is first in JSON format, which is then further converted to a structured format (CSV) that is simpler for machine learning techniques to understand. Later on, the classifier will function more effectively thanks to these comma separated values. The authors experimented with both supervised and unsupervised machine learning methods. Supervised machine learning algorithms performed better in this example, with an accuracy rate of approximately 98%. The dataset for supervised machine learning is divided into training and testing sets. They used 80% of the samples to coach the classifier and the rest to test it.

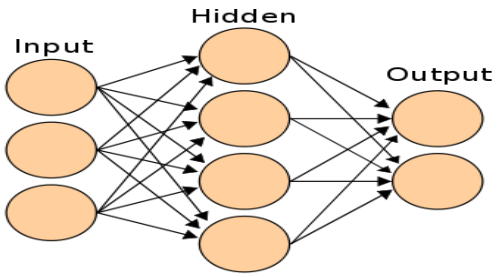
Sybil Frame classifies at multiple levels. There are both content-based and structure-based methodologies. The dataset is analysed using a content-based technique, which extracts data necessary to compute historical information about nodes and edges. Using a Markov random field and loopy belief propagation, which uses prior knowledge, the structure-based technique correlates nodes. The first step of the Sybil Frame technique uses a content-based approach, and the second stage uses a structure-based approach.

IV. PROPOSED SOLUTION

I'm applying artificial neural networks created through machine learning to evaluate the likelihood that a friend request is genuine or not. Every neuron (node) processes each equation through a Sigmoid function to keep the answers between the range of 0.0 and 1.0. This could easily be multiplied by 100 at the output end to provide us with the likelihood that the request is malicious. Our approach would consist of a single deep neural network with a single hidden layer.

Formulation of Neural network :

Let’s start by understanding formulation of a simple hidden layer neural network. A simple neural network can be represented as shown in the figure below:



The most amazing fact in an ANN is the linkage between nodes. The ANN algorithm only uses inputs as known values. Weights represent the nodes' connectivity during this method.

Following is the framework in which artificial neural networks work :

1. To begin the algorithm, assign random weights to all or some of the links.
2. Determine the activation rate of Hidden nodes by using the inputs and the ensuing (input -> Hidden node) linkage.
3. Calculate the activation rate of the output nodes using the connections and hidden nodes' activation rates.
4. Determine the output node's error rate and recalibrate every link between hidden nodes and output nodes.
5. Cascade the error to Hidden nodes using the weights of and error discovered at output node.
6. Adjust the weights between the input nodes and the hidden nodes.
7. Continue using this approach until the convergence requirement is reached.
8. Score the activation using the ultimate linkage weights. rate of the output nodes.

6.2.2 Sigmoid Activation Function :

Something that is curved in two directions is described by the term "sigmoid." We are only interested in one of the several sigmoid functions. Given that it is known as the logistic function, the mathematical formulation is simple:

f(x) = 1/1+e-x

The maximum value of the curve is determined by the constant L, hence the constant k affects how steep the transition is. The plot below displays examples of the logistic function for various values of L; thus, the plot below displays curves for various values of k.

The threshold value is where classification occurs, and the graph of the sigmoid activation function will have an S-shape.

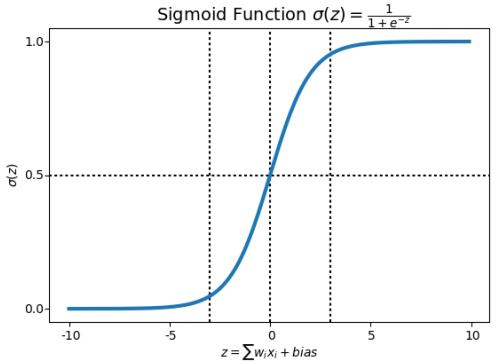


fig : Sigmoid Graph

IV. SIMULATION RESULTS

a. 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 .

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. 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. 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 of (1). 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.

b. 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 . Keras itself is capable of running on top of TensforFlow, CNTK, or Theanoc. 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 , 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.

V. CONCLUSIONS

In this paper, we use machine learning, to be specific an artifical neural network to figure out what are the possibilities that a friend request is valid are or not. Every condition at every neuron (node) is put through a Sigmoid function. We use a dataset collection by Facebook or other informal communities. This would permit the introduced deep learning calculation to become familiar with the examples of bot conduct by backpropagation, limiting the last expense function and changing every neuron's weight and predisposition.

In this paper, we frame the classes and libraries involved. We additionally examine the sigmoid function and how are not set in node and utilized. We likewise consider the boundaries of the social organization page which are the most essential to our answer.

REFERENCES

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

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