International Journal of Computer and Internet Security.
ISSN 0974-2247 Volume 15, Number 1 (2023), pp. 1-6

https://dx.doi.org/10.37624/IJCIS/15.1.2023.1-6

Instagram Fake Profile Detection

A.Nageswari[1], Ciddenki Suchitha Reddy[2], Sai Tejaswini Kommidi[3], Mahathi Tanikella[4] and Anusha P[5]

[1]Assistant Professor,
Department of Information Technology, GNITS, Hyderabad, India
[2,3,4,5,]Undergraduate Student,
Department of Information Technology, GNITS, Hyderabad, India
Corresponding Author Email: irpreddy@gnits.ac.in

Abstract

All fake users' main intention is to send friend requests to normal users to hack their machine or to steal their data. By analyzing these features of all accounts, the classifier will mark whether the user profile is fake or genuine. This model uses MLP classifier and can process a large dataset of accounts at once. The MLP classifier will be trained with all previous users fake and genuine account data and then when we give new data then the system analyses the data and displays whether the account is fake or not. We have also listed the classes and libraries involved. Also, the accuracy of this classifier is very high compared to other machine learning algorithms. The technologies used in this project are HTML, CSS, Python and Flask.

Keywords: Fake profiles, MLP Classifier, Friend requests

1. INTRODUCTION

Now a days, the dependency of people on computer technology has increased to a very high level leading to crimes like data breach and hacking. These attacks are done in a very attractive way so that the users tend to believe them easily by creating fake profiles using attractive account names and details. Currently, there is little incentive for social networks to improve their data security. These breaches often target social media networks like Instagram. By 2022Instagram has reached a total population of 2.35 billion users making it the most popular choice of social media. Social media networks make revenue from the data provided by users. Average users do not realise that they give up their personal information the moment they start using social media networks. The solution presented in this paper intends to focus on the dangers of a bot or person in the form of a fake profile on your social media. This solution for this can be made

using an algorithm. The languages that we chose are flask, python, HTML, CSS. The classifier 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 is a fake friend request fishing for information. Our algorithm would work with the help of the details of social media accounts and the details that the user put on their accounts, as we need a dataset to train our model and then verify whether the profiles are fake or not.

2. RELATED WORK

[1] The paper is grounded in the fact that the posts of real users reveal varied categories of emotions such as joy, sadness, anger, fear, etc. based on their life experiences. On the contrary, fake users share posts to accomplish a specific purpose, and therefore, it is highly likely that their post content will contain similar types of emotions. By analyzing the unique dataset obtained by the software in combination with machine learning techniques, they developed classifiers, which can predict which Facebook profiles have high probabilities of being fake.

[2] This paper focuses on the analysis of individual social network profiles, with the aim of identifying the characteristics or a combination thereof that help in distinguishing the legitimate and the fake accounts. Specifically, various features are extracted from the profiles and posts, and then machine learning algorithms are used to build a classifier capable of detecting fake accounts.

[3] This paper is based on detecting fake profiles considering the characteristics of the user profile. This research utilized data mining techniques to detect fake profiles. A set of supervised (ID3 decision tree, k-NN, and SVM) and unsupervised (k-Means and k-medoids) algorithms were applied to 12 behavioral and non-behavioral discriminative profile attributes from a dataset of 982 profiles.

3. PROPOSED SYSTEM

To implement the detection of fake profiles we used MLP classifier we used the following architecture:

Data Collection: Collect data from a social media platform like Instagram because it is very popular these days and is also prone to many attacks recently. The data includes information about user activity, content posted, and other relevant features that can help distinguish real and fake profiles. We collected information related to the profile picture, username, length of the username, full name, length of the full name, name is same as username, description length, external URL, private account or not, number of posts, number of followers, number of people the user follows, and whether the account is fake or not for training.

The data is to read using the readCSV() method. This is read into the training set. The profile pic is a binary value representing whether the profile picture is present (1) or not (0). Nums/length username is ratio of the number of numbers in the username to the total length of username. Nums/length full name is the ratio of the number of numbers in the full name to the total length of full name. Name==username tells whether the username is same as the name of the user. Description length is the total length of the

description given by the user under their account. External URL is also a binary value which represents the presence of a link leading to external websites. Private is a binary value representing whether the account is private (1) or not (0). Posts is an integer value of the number of posts posted by the user. Followers represents the number of followers of the account. Follows represents the number of accounts the account is following. Fake is a binary value representing whether the account is fake (1) or not (0) which is used for training the classifier.

Data Pre-processing: Cleaning the data, removing any unnecessary columns or rows, and converting the data into a suitable format for the MLP classifier. Also, removing any duplicates and outliers from the data.

Feature Extraction: Extract important features from the data that can be used to differentiate between real and fake profiles. These features can include user activity level, the number of friends, the content posted by the user, the frequency of posts, and other social network characteristics. The feature extraction is done by the classifier internally by the MLP classifier by the multiple layers present in the neural network by using the MLP Classifier from sklearn.neural_network.

Split the Data: Now, splitting the data into training and testing sets. The training data will be used to train the MLP classifier, while the testing data will be used to evaluate its performance. We have split our data into train and test size of 0.8 and 0.2 respectively using train_test_split from sklearn.

Train the MLP Classifier: Use the training data to train the MLP classifier. Start with a simple architecture and gradually increase its complexity until it performs well on the training data. Now by using the instance of MLP Classifier we trained the model.

Test the MLP Classifier: Using the testing data to evaluate the performance of the MLP classifier. Not only accuracy but precision, recall, and F1 score are also calculated. To fully evaluate the effectiveness of a model, we examined both precision and recall. F1 score calculates how many times a model made a correct prediction across the entire dataset.

Improve the MLP Classifier: If the accuracy is low, try different hyperparameters, adjust the feature extraction method, etc.

Deploy the MLP Classifier: Once the MLP classifier performs well on the testing data, deploy it to detect fake profiles on social media platforms.

Once the database is collected from the information available in Instagram, the account details, pre-processing is performed on the database to make it ready for feature extraction. Now split the data into train and test data in the ratio of 0.8 and 0.2. Using the train data, train the MLP Classifier. Now the classifier is ready to predict the output for any new data set. Now the test data is used to test the classifier. Now the classifier is ready to predict output for any dataset. Not only the result, but also the accuracy, precision, recall and f1 score are calculated and displayed. If the accuracy is not enough the admin can modify the database by removing the redundant sets and noise in the database.

The MLP (Multilayer Perceptron) classifier is an artificial neural network (ANN) which has multiple layers of interconnected nodes or neurons. It is a supervised learning algorithm used for classification tasks.

The MLP classifier is composed of three types of layers: input, hidden, and output layers. The input layer receives the input data, which is then processed by the hidden layers before reaching the output layer. Each layer consists of multiple neurons, and each neuron is connected to every neuron in the next layer.
While training, the MLP classifier adjusts the weights of the connections between the neurons to reduce the error between the predicted output and the actual output. The process of adjusting the weights is called backpropagation.
The MLP classifier is capable of learning complex patterns and relationships in the input data, making it suitable for a wide range of classification tasks. It is a popular choice for tasks such as image classification, speech recognition, and natural language processing.
The MLP classifier is a powerful machine learning algorithm that can learn complex patterns and relationships in the input data, making it a useful tool for many classification tasks like detection of fake profiles.
This system detects a profile with highest accuracy than other machine learning algorithms because MLP Classifier is better than other Machine Learning algorithms.
Implementing detection of fake profiles using MLP classifier involves data collection, data pre-processing, feature extraction, training, testing, improvement, and deployment.

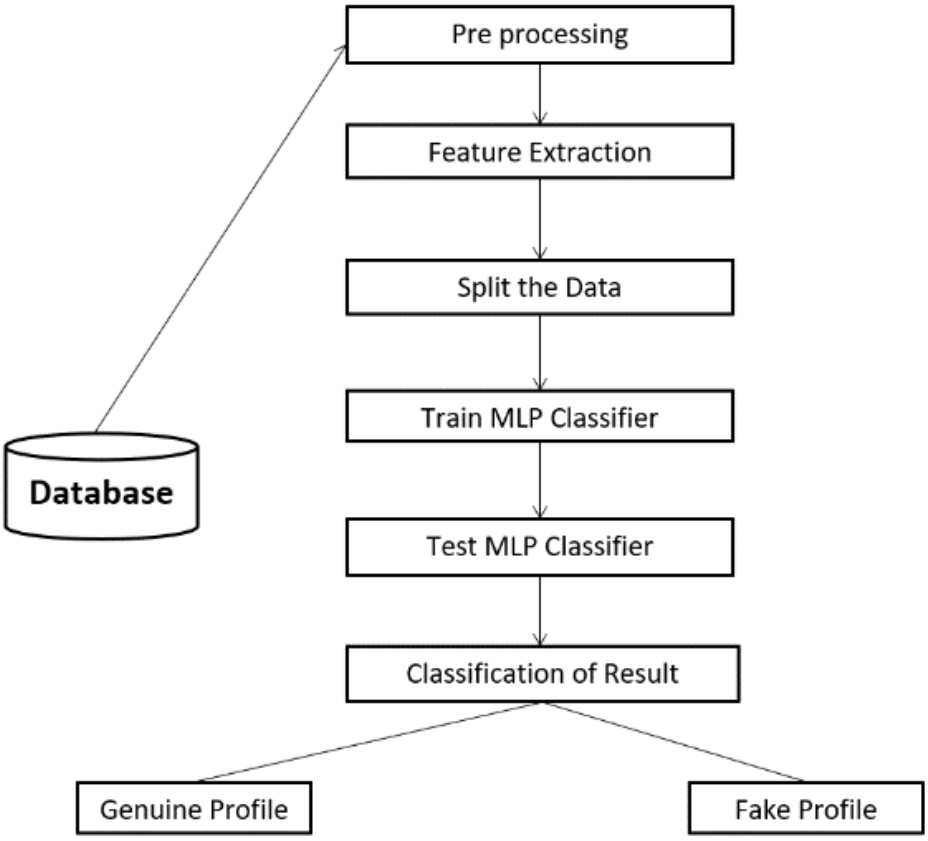

Figure 3.1: System Architecture

4. IMPLEMENTATION

To develop this system, we used MLP classifier with scikit-learn for simplicity, we used python and Flask as it is a light web framework. We created two files model.py and server.py.

In the model.py file, we developed and trained our model, in a server.py, we coded to handle POST requests and return the results and finally in the request.py, we sent requests with the features to the server and received the results.

In th**e** model.py file we developed our MLP model and trained it. We predicted whether a profile is fake or not based on the dataset provided to the model. Here we used the dataset 'Instagram fake spammer genuine accounts' from Kaggle.

Importing the libraries that we are going to use to develop our model.py numpy and pandas to manipulate the matrices and data respectively, sklearn.model_selection for splitting data into train and test set and sklearn.neural_networkto train our model using MLPClassifier. requests sends requests to the server and render_template() returns the result to the respective html file.

Here we imported the dataset using pandas library and later separated the features and label from the dataset. Now, we have split our data into train and test size of 0.8 and 0.2 respectively using train_test_split from sklearn. The object is instantiated as a calssifier of the class MLPClassifier() and trained using X_train and y_train. Later the predicted results are stored in the predicted.

In the server.py file, we have used the flask web framework to handle the requests that we will get from the request.py. And then we imported numpy to create the array of requested data.

Now, we created an instance of the Flask() and loaded the model. Here, we have the methodrender_template. The result is stored into the variable named result and we return this variable to the respective html file using render_template() method.

Finally, we will run our server. Here we have used port 1122 and have set debug=True since if we get any error, we can debug it and solve it.

5. CONCLUSION AND FUTURE SCOPE

This idea came up with Artificial Neural Networks besides machine learning algorithms because they construct the structure of human brain. From the social media sites, we can easily find fake profiles by implementing this technique. In this system to point out the fake profiles we have taken the Instagram dataset because many accounts have been hacked these days on Instagram. Implementing this system in social media sites can decrease the chances of hacking and help naïve users think before accepting any request. Also, the accuracy of the result using MLP Classifier is higher than other Machine Learning algorithms. We reached an accuracy of about 98 percent. Our major complication is that a person can have numerous Instagram accounts which is an added benefit for those who create fake profiles and accounts on social media sites. We can add 12-digit Aadhar card number while creating an account, as a result we can limit single account for single user and there is no probability of fake profiles in social networks. We can improve this project by using the external URL directly in the dataset so that the link can be checked whether it is malicious or not also.

6. REFERENCES

[1] Adam Breuer, Roee Eilat, and Udi Weinsberg. 2020. Friend or Faux: Graph Based Early Detection of Fake Accounts on Social Networks. In Proceedings of The Web Conference 2020 (WWW ’20), April 20–24, 2020, Taipei, Taiwan. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380204

[2] Aleksei Romanov, Alexander Semenov, Oleksiy Mazhelis and JariVeijalainen (2017). Detection of Fake Profiles in Social Media. In Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017), pages 363-369.

[3] Dr. Sanjeev Dhawan, Ekta (2016). Implications of Various Fake Profile Detection Techniques in Social Networks. In the proceedings of IOSR Journal of Computer Engineering (IOSR-JCE).

[4] Mohammed Basil Albayati* & Ahmad Mousa Altamimi (2019). Identifying Fake Facebook Profiles Using Data Mining Techniques. Article in Journal of ICT Research and Applications-September 2019.