ISSN 2063-5346

USING ML TECHNIQUES REVEALING SNEAKY SOCIAL BOTTING ON TWITTER BY ANALYZING USER PROFILE ATTRIBUTES

Mohammad Azmathullah[1] Dr.Abdul Rasool Mohammed[2]

Article History: Received: 10.05.2023 Revised: 29.05.2023 Accepted: 09.06.2023

Abstract

Due to the widespread use of social media, scammers attempt to deploy malicious social bots that produce fake tweets, attempt to build relationships with other users by pretending to be followers or attempt to create several fake accounts that engage in malicious actions. Additionally, they frequently post malicious URLs that direct real people to malicious web servers. Therefore, it is crucial to distinguish between legitimate accounts and bot accounts. It has been found that profile-based features and URL features, such as redirected URLs, spam data, frequency of URL sharing, etc., are better indicators of bots than social factors. In this study, we propose a novel method that exposes malicious bots on social networks by utilizing profile-based attributes and Deep Learning algorithms. We apply the aforementioned model to the Twitter data set and see that it performs better than other methods. We also made an effort to create a web application that might demonstrate that the aforementioned strategy performs better than other models that are already in use.

Keywords—*Learning automata (LA), malicious social bots, online social networks (OSNs), trust..*

[1]Research Scholar, Dept. of Computer Science and Engineering, Lords Institute of Engineering & Technology, Hyderabad, Telangana

[2] Associate Professor, Dept. of Computer Science and Engineering, Lords Institute of Engineering & Technology, Hyderabad, Telangana

DOI:10.48047/ecb/2023.12.9.30

I. INTRODUCTION

Social media platforms have a large number of accounts where users can post and share material. Since there are millions of profiles, it is impossible to personally verify that an account is legitimate and not a fraudulent account because of this[1]. As a result, users may unknowingly share their personal information and it may be used for other illegal activities. Over time, many fake accounts have been created, acting like malevolent bots that may harm legitimate users by spamming, posting, and sharing URLs that direct users to malicious servers, etc. Therefore, it is absolutely necessary[3]. To monitor and, if necessary, remove all bot accounts, one must be able to determine whether a Twitter account is one or not. The majority of currently used strategies make use of social activity-related traits, but it has been noted that profile-based features and URL features play a significant part in identifying malicious bots[2]. Therefore, in this research, we develop a novel method that can recognize a bot utilizing features from a profile. We fed the proposed model, which was created using cutting-edge machine learning techniques, the Twitter data set, and saw that it performed better than the existing systems.

II. RESEARCH BACKGROUND

A. Problem Statement

Due to the widespread use of social media, there has been a significant increase in cybercrimes such as phishing, spamming, and other forms of abuse by nefarious social bots that attempt to post malicious content to users and direct them to websites where the user's personal information can be obtained and used inappropriatel[4]y. To prevent cybercrimes from happening, it is imperative that these malicious bots are found. In this project, we suggest a cutting-edge method for locating malicious bots based on the user's profile features[5]

B. Aim Of The Project

The primary goal of the project is to locate harmful bots in Twitter accounts using aspects of the user's profile. We have incorporated the Twitter data set into the proposed model, which was created utilizing cutting-edge machine-learning methodologies.

C. Scope Of The Project

The project's objectives are restricted to calculating the proposed model's accuracy and identifying harmful bots. The system administrator uses test data to test the proposed model and training data to train the model. The "results.csv" file contains the test data's results. The project does not include deactivating bots, monitoring user accounts, or maintaining user accounts. Although malicious bots can be exposed using URL-based features, this method has not been used in this study because the BOT Prediction algorithm has an accuracy of roughly 93% for the given dataset..

D. Technical Approach

The technological strategy to solve the issue is listed below:

1. Dataset identification
2. Exploratory data analysis
3. Dataset preparation and NLP approaches
4. Running the dataset through many algorithms to see which one best fits the situation.
5. Developing a final classifier model and training the final classifier
6. Validating the ultimate classifier and recording the outcomes.

III. SYSTEM ANALYSIS

A. Research Gap

Due to the widespread usage of social media sites like Facebook, Twitter, etc., malicious individuals attempt to utilize bots to create phony accounts, control users' attitudes, and send them to malicious websites by spamming. Because they rely on qualities that search bots employ to build relationships with real consumers, traditional techniques to identify search bots are ineffective. It is impossible to tell

manually if the account is a bot or not. Numerous social feature-based techniques have been developed, but they are not particularly useful.

.

B. Proposed System

The "BOT PREDICTION ALGORITHM" is a revolutionary technique used by the proposed system to recognize bots utilizing URL data and relationship features. The profile-based elements listed in the table below would assist in spotting malicious social bots and determining whether users were being forwarded to bogus websites where they may enter their personal information. The features listed below have made it possible to recognize bots with great accuracy.

Below are the features that are being used in the project:

Feature Name	Description
Id	Twitter Id of the twitter account
followers_count	No. of followers for the user
friends_count	No. of friends for the user
verified	Boolean value which describes if a user is verified or not
name	Name of the user in the twitter account
description	Description of the user
screen_name	Displayed screen name on the twitter account
status	Latest status of the user
listed_count	No of users who really want to follow the user.

BOT Prediction algorithm

The Bot Prediction Algorithm's steps are as follows:

1. Download the dataset.
2. Applying feature engineering to the verified and id columns.
3. Changing id to an integer.
4. Changing verified to a vector.
5. Verify whether the screen name or name contains the character "b0t"
6. Verify if the user is authentic
7. See if the description includes BuzzFeed.
8. See if the listed count exceeds 16000

Advantages:

- High precision
- Extendable to real-time settings..

IV. PROJECT IMPLIMENTATION

Proposed Modular Implementation

Below is the proposed modular implementation of the project. It consists the below Admin modules:

Admin Module:

The admin of the system is responsible for the activities like:

1. Uploading the dataset
2. An examination of Twitter user data.
3. Evaluation of several machine learning methods using the Twitter bot dataset.
4. Create a model for detecting malicious bots.
5. Examine how well the algorithms performed on the provided dataset.
6. Using test data, check the model for harmful bot prediction.

A. SYSTEM DESIGN

1. Data Flow Diagram: Admin

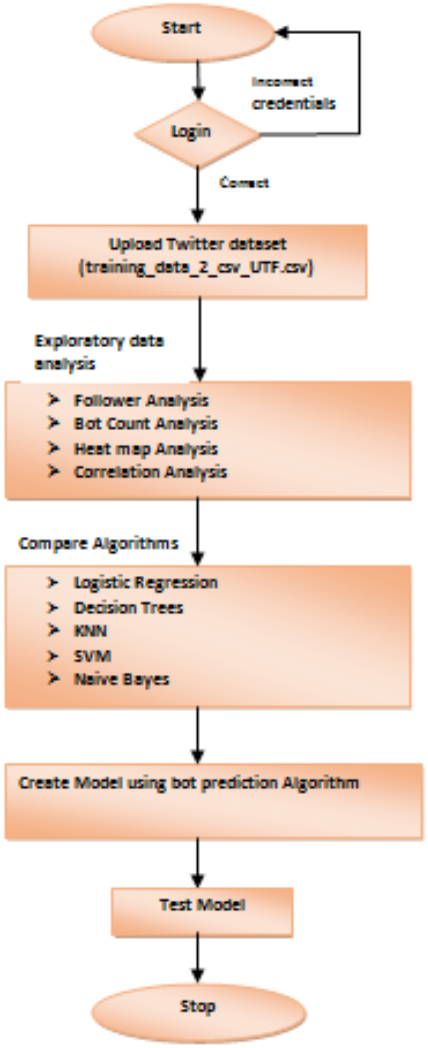

Figure 1: A Data Flow Diagram for Admin

V. IMPLEMENTATION AND RESULT ANALYSIS

A. Project execution process:

1. Upload Dataset

The system administrator can upload datasets that are used to train machine learning models on this page. To upload a file to a server, an administrator must first choose the file by clicking the Choose file button, then click the Upload button. A success message indicating that the file was successfully uploaded would be shown once the upload was finished. We are utilising the datasets test data 4 students and training data 2 csv UTF reviews for this project.

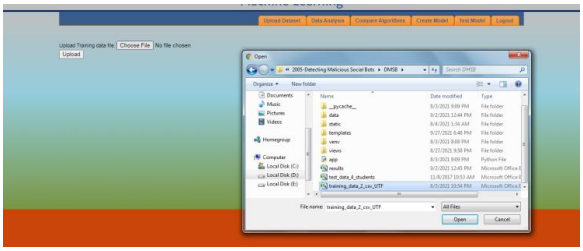

Figure 3: Upload Dataset

2. Data Analysis

Exploratory data analysis is done on the dataset to uncover patterns, find missing data, and establish links between different output characteristics using graphs, statistics, etc.

a) Follower Analysis:

The follower analysis is displayed in the graph below.

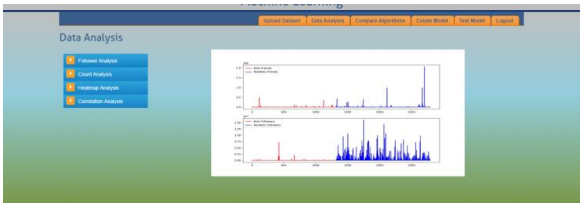

Figure 5: Follower Analysis

b) CountAnalysis:

The below graph shows the Count analysis.

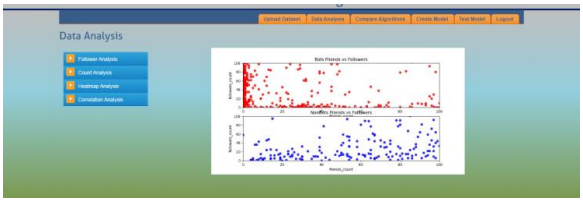

Figure 6: Count Analysis

3. Compare Algorithms

The administrator can use this page to train several algorithms on a dataset and determine each algorithm's test accuracy.

a) Logistic Regression

The test accuracy is 0.86 when the dataset is fed into the logistic regression algorithm, as we see.

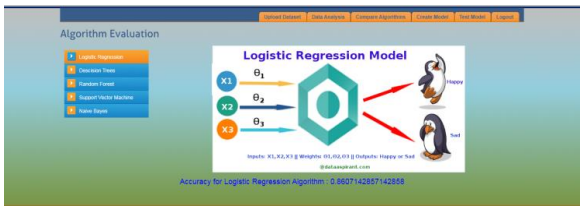

Figure 7: Logistic Regression

b) **Decision Trees**

The test accuracy is 0.8785714285714286 when the dataset is fed into the decision tree method, as we can see.

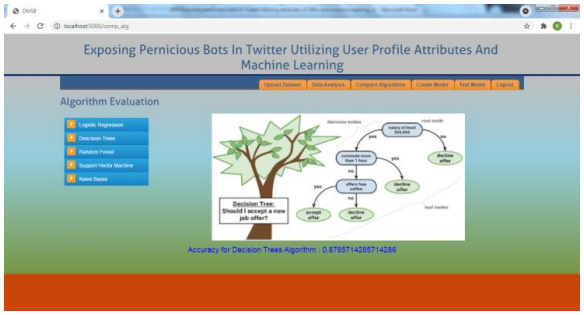

Figure 8: Decision Trees

c) Support Vector Machine

The test accuracy is 0.6178571428571429 when the dataset is fed to the Support Vector Machine method.

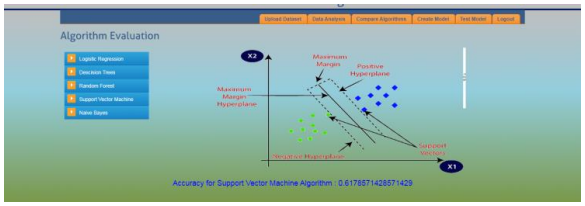

Figure 9: Support Vector Machine Algorithm

4. Create Model

The Create Model button can be used to create the Bot Prediction Model. After pressing the button, a success message is presented and the model is built. BOT Prediction is 96.94% accurate.

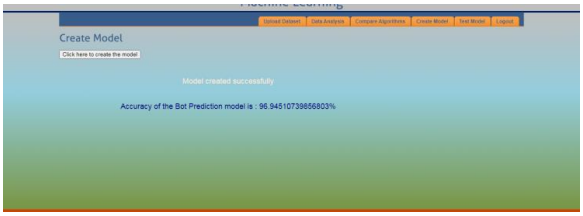

Figure 10: Create Model

5. Test Model:

The model can be tested using the below screen.

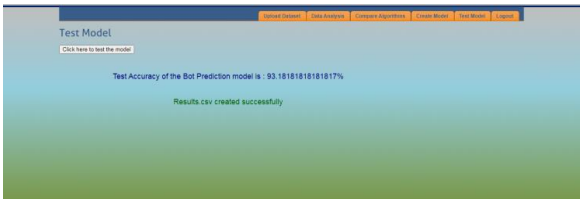

Figure 11: Test model

B. Metrics Evaluation :

Accuracy- One parameter for assessing classification models is accuracy. Informally, accuracy is the percentage of accurate predictions made by our model.

Macro avg - The macro average, also known as the precision, memory, and f1 score, is the arithmetic mean of each individual class. When all classes must be treated equally, macro average scores are used to assess the classifier's overall performance in comparison to the most popular class labels.

Weighted avg- A calculation that accounts for the varied levels of significance of the numbers in a data set is known as a weighted average.

Metrics for Algorithms in CompAlg.py

Classification report contains the complete metric information of the evaluated algorithm. They are Precision, Recall, F1-Score, Support

Precision – What percent of your predictions were correct?

Precision is the capacity of a classifier to avoid classifying as positive anything that is

in fact negative. It is described for each class as the proportion of true positives to the total of true and false positives.

TP – True Positives

FP – False Positives

Precision – Accuracy of positive predictions.

Precision = TP/(TP + FP)

1) Recall – What percent of the positive cases did you catch?

The capacity of a classifier to locate every successful instance is known as recall. It is described as the proportion of true positives to the total of true positives and false negatives for each class.

FN – False Negatives

Recall: Fraction of positives that were correctly identified.

Recall = TP/(TP+FN)

2) F1 score – What percent of positive predictions were correct?

The F1 score is a weighted harmonic mean of recall and precision, with 1.0 representing the best result and 0.0 the lowest. F1 scores typically perform worse than accuracy measures because they incorporate precision and recall into their computation. It is often recommended to compare classifier models using the weighted average of F1, rather than overall accuracy.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

The amount of real instances of the class in the given dataset is known as support. The requirement for stratified sampling or rebalancing may be indicated by unbalanced support in the training data, which may point to structural flaws in the classifier's reported scores.

1) Metrics using Decision Tree Algorithm:

DT Accuracy Score: 0. 8785714285714286

DT Precision Score: 0. 8785714285714286

DT Recall Score: 0. 8785714285714286

DT F1 Score: 0. 8785714285714286

Decision Tree accuracy: 87.85714285714286

Decision Tree Classification Report:

Precision	recall	f1-score	support
0	0.76	0.73	0.75 1507
1	0.899	0.895	0.893 1313
accuracy		0.88	4514
macro avg	0.81	0.81	0.88 4514
weighted avg	0.81	0.81	0.88 4514

2) Metrics using Random Forest Algorithm:

RF Accuracy Score: 0.8646743464776252

RF Precision Score: 0.8646743464776252

RF Recall Score: 0.8646743464776252

RF F1 Score: 0.8646743464776252

Random Forest accuracy: 86.46743464776252

Random Forest Classification Report:

Precision	recall	f1-score	support
0	0.88	0.73	0.80 1507
1	0.86	0.97	0.91 1313
accuracy		0.86	4514
macro avg	0.86	0.86	0.86 4514
weighted avg	0.86	0.85	0.86 4514

3) Metrics using Naive Bayes Algorithm:

NB Accuracy Score: 0. 6976190476190476

NB Precision Score: 0. 6976190476190476

NB Recall Score: 0. 6976190476190476

NB F1 Score: 0. 6976190476190476

Naive Bayes accuracy: 69.76190476190476

Naive Bayes Classification Report:

	precision	recall	f1-score	support
0	0.76	0.71	0.74	1507
1	0.76	0.71	0.73	1313
accuracy			0.69	4514
macro avg	0.75	0.74	0.69	4514
weighted avg	0.75	0.75	0.70	4514

4) Metrics using Logistic Regression Algorithm:

LR Accuracy Score: 0.8595480726628267

LR Precision Score: 0.8595480726628267

LR Recall Score: 0.8595480726628267

LR F1 Score: 0.8595480726628266

Logistic Regression accuracy: 85.95480726628267

Logistic Regression Classification Report:

	Precision	recall	f1-score	support
0	0.85	0.81	0.83	1507
1	0.84	0.94	0.89	1313
2	0.88	0.84	0.86	1694
accuracy			0.86	4514
macro avg	0.86	0.86	0.86	4514
weighted avg	0.86	0.86	0.86	4514

5) Metrics using Support Vector Machine Algorithm:

SVC Accuracy Score: 0. 6178571428571429

SVC Precision Score: 0. 6178571428571429

SVC Recall Score: 0. 6178571428571429

SVC F1 Score: 0. 6178571428571429

Support Vector Machine accuracy: 61.78571428571429%

Support Vector Machine Classification Report:

	Precision	recall	f1-score	support
0	0.78	0.73	0.75	1507
1	0.75	0.88	0.81	1313
accuracy			0.61	4514
macro avg	0.79	0.80	0. 61	4514
weighted avg	0.80	0.79	0.61	4514

VI. CONCLUSION

Conclusion : In this research, we attempt to put into practice a deep learning model that exposes harmful bots on the Twitter network by using profile-based features. One could determine whether or not the account is used to upload content is a bot based on the aforementioned characteristics. In this research, the profile-based features-based Bot Prediction model was built and trained using the Twitter data set. Additionally, we tried to produce test outputs by inputting test data and testing the accuracy of it. The results of the experiments demonstrate that the suggested method provides the maximum accuracy. This methodology, which employs profile-based characteristics to expose malicious bots and URL-based features to identify spammed content, will eventually be expanded to larger datasets and real-time situations

References

[1] P. Shi, Z. Zhang, and K.-K.-R. Choo, "Detecting malicious social bots based on clickstream sequences," IEEE Access, vol. 7, pp. 28855–28862, 2019.

[2] G. Lingam, R. R. Rout, and D. V. L. N. Somayajulu, "Adaptive deep Q-learning model for detecting social bots and influential users in online social networks," Appl. Intell., vol. 49, no. 11, pp. 3947–3964, Nov. 2019.

[3] D. Choi, J. Han, S. Chun, E. Rappos, S. Robert, and T. T. Kwon, "Bit.ly/practice: Uncovering content publishing and sharing through URL shortening services," Telematics Inform., vol. 35, no. 5, pp. 1310–1323, 2018.

[4] S. Lee and J. Kim, "Fluxing botnet command and control channels with URL shortening services," Comput. Commun., vol. 36, no. 3, pp. 320–332, Feb. 2013.

[5] S. Madisetty and M. S. Desarkar, "A neural network-based ensemble approach for spam detection in Twitter," IEEE Trans. Comput. Social Syst., vol. 5, no. 4, pp. 973–984, Dec. 2018.

[6] H. B. Kazemian and S. Ahmed, "Comparisons of machine learning techniques for detecting malicious webpages," Expert Syst. Appl., vol. 42, no. 3, pp. 1166–1177, Feb. 2015.

[7] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, "A framework for real-time spam detection in Twitter," in Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS), Jan. 2018, pp. 380–383.

[8] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), 2017, p. 3.

[9] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "Key challenges in defending against malicious socialbots," Presented at the 5th USENIX Workshop Large-Scale Exploits Emergent Threats, 2012, pp. 1–4.

[10] G. Yan, "Peri-watchdog: Hunting for hidden botnets in the periphery of online social networks," Comput. Netw., vol. 57, no. 2, pp. 540–555, Feb. 2013.