Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777634

TWITTER BOT DETECTION USING MACHINE LEARNING ALGORITHMS

P. Sai Karthik Reddy and P. Sai Nath and Dr. J. Vijayashree*
School of Computer Science and Engineering, Vellore Institute Of Technology, Vellore, India

Abstract A staggering number of individuals use social media platforms nowadays, which encompass a wide range of media. It is estimated that Twitter has 330 million monthly active users. Twitter is a social networking site where users may express their ideas and opinions on a variety of situations happening around the world, it may share information about the economy, stocks, important information about business, politics etc. With millions of users actively using it, it is one of the most widely used social networking sites worldwide. It is one of the fastest methods of information transfer. Can be said as one of the quickest ways to deliver information. A Twitter account that has been programmed to carry out social media tasks automatically by scheduling tweets is known as a Twitter bot. This found that a growing number of problematic bot accounts are disinformation, comedy, and promote unverified material, which can negatively affect several concerns. It may alter user confidence, public perceptions of a problem, or even the social order. Thus, it is mandatory to know that tweets are made by real humans or bots. For the implementation of our project, we have used different machine learning algorithms like KNN, Gaussian Naïve Bayes, Decision Tree Classifier, Random Forest. We will be training the dataset with these algorithms. As a result, we concluded Random Forest is the best algorithm which has highest accuracy among all the others. Random Forest machine learning technique is ultimately chosen because it can avoid most overfitting and produce a generalized model that can be used accurately right after training. The flask server was utilized to connect our model to the web content. Our analysis of our framework's results shows that we can reasonably determine if a user is a person or a bot.

Keywords Bot detection, Decision Tree Classifier, Gaussian Naïve Baye, KNN, Random Forest, Twitter Bot.

1 Introduction

Twitter, created by Jack Dorsey in 2006, is a social media platform experiencing rapid growth. Individuals can transmit brief messages referred to as tweets, with a character limit of 140., resulting in an average of 500 million daily tweets. A hashtag (#) is used to track and follow specific subjects, with popular hashtags referred to as trending topics. Twitter connections work both ways. Individuals can send tweets immediately to those who follow them. By retweeting, content can be shared with a larger audience. Users from diverse backgrounds utilize Twitter for various purposes. On average, approximately 6,000 tweets are posted every second, resulting in approximately 360,000 tweets per minute.

Twitter is facing a concerning problem of an increasing number of fake or automated accounts, commonly known as "bot" accounts. These accounts either spread harmful content or are utilized to

amplify the visibility of legitimate accounts. The Twitter API facilitates the creation of these bots, which can be generated within seconds. The excessive volume of tweets produced by these bots can manipulate trending topics on the platform. For example If an account is found to predominantly retweet rather than create original tweets, has a low number of followers despite posting frequently, lacks a profile picture and biography, and shares identical tweet content with another user simultaneously, it may indicate that the account is fake or automated. According to research carried out by the University of Southern California and Indiana University, approximately 15% of Twitter accounts that are active are bots. In 2019, Twitter suspended around 26,000 accounts, which accounts for less than 1% of the total number of active users on the platform, estimated to be around 300 million. The malicious objectives of Twitter bots include: 1) spreading false information and rumors, 2) damaging someone's reputation, 3) generating fake conversations to steal credentials, 4) leading users to fraudulent websites, and 5) influencing the popularity of individuals or groups by manipulating their opinions. Netflix bot is a Twitter account that shares updates regarding new shows and content available on Netflix. @HundredZeros is another Twitterbot that tweets out links to freely accessible eBooks on Amazon. Twitter does not implement a "captcha" during the account creation process, making it easy for an automated system to generate a large number of bots that can then be run through the available Twitter Dataset. Various characteristics, such as the followers count, friends, location, user name, verification status, liked , id, bio, URL, and listed count, are used to extract features for detecting bots. The Spearman correlation coefficient is applied during feature extraction. The process of collecting data is adjusted or optimized to effectively detect bots. Various techniques have been employed, including DT, Naive Bayes, Random Forest, KNN, Bag of Words. As per a study, the Random Forest algorithm provides the highest accuracy, which is 90%, compared to the Gaussian Naive Bayes algorithm with 83% accuracy, and the KNN algorithm with 70% accuracy. The Decision Tree algorithm also has a 78% accuracy.

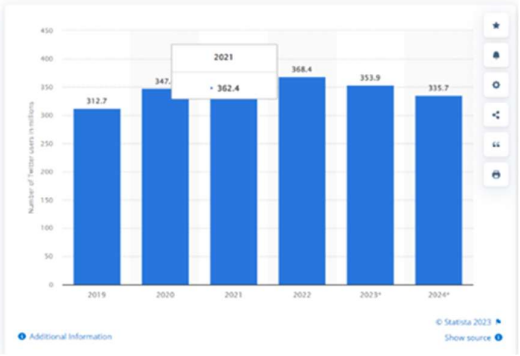

Fig-1-Number of Twitter users worldwide from 2019 to 2024(in millions)

2 RELATED WORK AND BACKGROUND

Various methods have been utilized by people worldwide to detect bots on Twitter [1]. This research aims to explore diverse techniques for identifying, analyzing, and categorizing bots while also determining the extent to which bots utilize hashtags [6]. The dataset created was subjected to pre-processing, and fake accounts were identified using machine learning algorithms. Most bot detection techniques rely on supervised machine learning, which necessitates the use of manually labeled data [7]. This research employs publicly available

datasets to train the classifier model. To identify bots, various features have been utilized, such as username length, reposting frequency, temporal patterns, sentiment expression, follower-to-friend ratio, and message variability [8].

Our investigation into methodologies has revealed that there are primarily two approaches: one based on language modeling and the other on account attributes [3]. This strategy is based on the presumption that bot accounts would vary from human accounts in several keyways. Overall, these studies demonstrate the effectiveness of machine learning algorithms in detecting bots on Twitter.[11] They also highlight the importance of considering factors such as account behavior, language use, and content topics in identifying bot accounts.

KNN:

K-Nearest Neighbor (KNN) is a machine learning technique used for regression and categorization tasks that relies on the proximity of new data points to existing ones in the training set. This method is commonly used for Twitter bot detection. However, using KNN for this purpose can have disadvantages such as being computationally expensive and not being able to handle noisy data, which can lead to inaccurate results with low accuracy.

Gaussian Naive Bayes:

Gaussian Naive Bayes is used for categorization exercises that assumes the features are normally distributed and independent of each other, which makes it appropriate for continuous data. It determines the conditional probability of each class based on the input features and selects the highest probability class. This algorithm often used for Twitter bot detection, but it can have limitations in this context, such as being prone to overfitting and not being able to capture intricate relationships between features. Consequently, its accuracy may be limited in some scenarios.

Decision Tree:

It is a popular algorithm mainly used to build a model by repeatedly dividing the data into smaller subsets based on the most significant feature at each node, eventually leading to a leaf node that represents a decision or classification. This method is commonly used for Twitter bot detection, where it recursively splits the data based on informative features until it can assign a class label to the sample. Decision trees can handle both categorical and numerical data and can also identify important features for classification. However, decision trees are prone to overfitting and may not generalize well to new data, so ensemble methods like Random Forest are often preferred for Twitter bot detection tasks.

Random Forest:

Random Forest is machine learning algorithm that can be used for regression tasks and classification tasks. Random Forest is well-suited for handling high-dimensional and noisy data, as well as interactions between features. Due to its robustness, it has become a popular choice for Twitter bot detection. By identifying important features for classification, Random Forest can aid in interpreting the results.

In Twitter bot detection tasks, Random Forest algorithm is known for achieving higher accuracy than other popular algorithms such as KNN, Decision Tree, and Gaussian Naive Bayes. This is due to its ability to construct many decision trees on random data subsets, which mitigates overfitting and enhances generalization. Moreover, Random Forest is well-suited for handling high-dimensional data and feature interactions, which makes it a robust

algorithm in this context. Consequently, Random Forest is typically favored over other algorithms for its capability to achieve high accuracy in bot detection tasks.

Technique	Accuracy score
One class [1]	>95%
Decorate classifier [1]	F1 score of 0.88 or 88%
SVM [2]	70.3%
Neural networks [2]	90.1%
Random forest [2]	99.0%
Network-centric [3]	95.2%
K_f_reimp [4]	>98.0%
Statistical [5]	97.55% for Deep forest
Content-based [5]	92.83% and 94.25% for supervised Word2vec
Hybrid [5]	> 95% for synthetic minority overlapping
Machine learning [3]	97.6% for clustering with 126 features
Petri net model [8]	>97% for the random forest
Organized behaviour [9]	>97% for the random forest
SocialBotHunter [10]	99.0%
Spam detection [11]	SVM gives best performance
Malware detection [11]	Same as spam detection
Phishing detection [11]	Naïve Bayes classifier with clustering gives good results.

Fig2-accuracy scores of different algorithms

3 METHODOLOGY:

3.1 Data collection and pre-processing: To develop a Twitter bot classification model, we need to obtain Twitter account data for both authentic and bot accounts, as well as information such as friends_count, followers_count, and account creation date. Before preprocessing the data using Spearman correlation, we must first identify the relation between the accounts from the dataset.

3.2 Feature Extraction: Afterward, various features, including follower and friend count, location, screen name, verification status, URL, and listed count, will be extracted from the dataset.

3.3 Model Building: To construct the bot detection model, we will employ the dataset with all the algorithms like Random Forest, Naive Bayes, KNN, and Decision Trees. We will also be assessed to compare the performance of those all algorithms .

3.4 Model Evaluation: We will evaluate the performance of the Random Forest model using metrics like accuracy, precision, recall, and F1 score. The findings will be contrasted with the outcomes of other machine learning algorithms examined.

3.5 Real-Time Implementation: Finally, we will integrate the Random Forest model into a web application to enable real-time detection of Twitter bots.

4 PROPOSED SYSTEM:

To construct the model, we initially train our dataset using various mentioned algorithms for determining the best-suited among them for our needs. We experimented with KNN, Random Forest, Naive Bayes, and Decision Tree Classifier to identify the most effective algorithm. We trained our dataset using four different algorithms: KNN, Random Forest, Naïve Bayes, and Decision Tree Classifier. We pre-processed the dataset before supplying it to these algorithms. Then, we decided which algorithm was the most effective to use as the basis for our system. Based on our investigation, we discovered that the random forest method, which is a collection of decision trees, was the most accurate and effective algorithm. Usually, it offers better accuracy than a standard decision tree.The dataset included a set of 16 attributes, which were of different types, such as binary, numerical, and descriptive. 1) id 2) id_str 3) Screen name 4) Location 5) Description 6) Url 7) Follower's count 8) Friends count 9) Listed count 10) Created at 11) Favorites count 12) Verified 13) Statuses count 14) Lang 15) Status 16) Default profile

Among the 16 attributes in the dataset, user_name, name, bio, rank, listed_count and verified are binary attributes, while the remaining attributes are numerical values. To convert the non-binary attributes into binary-valued, a comparison was made with certain words. The analysis revealed that there are certain words that frequently appear in the descriptive fields of bot accounts, such as bot, cannabis, tweet me, mishear, XXX, sex, chick, prison, follow me, virus, jack, etc. Hence, the above words are found in the bio of the users account profile or the tweets he make or receive, it is considered as a bot account. The classification of bot or genuine account was done using various attributes such as friends count, followers count, and listed count. The dataset was parted into 80% for train dataset and 20% for test the model. The analysis revealed that the most crucial attribute among the nine attributes used was the 'verified' attribute. The model's predictions were heavily influenced by the "verified" attribute, as most accounts predicted as bots had "FALSE" in the verified attribute. However, accounts with "FALSE verified" do not necessarily mean they are bots, and the opposite is also true. Additionally, the model was able to identify many non-popular bot accounts, as these types of accounts often try to stay hidden. Overall, the Random Forest algorithm was effective in accurately predicting whether an account was a bot or genuine.

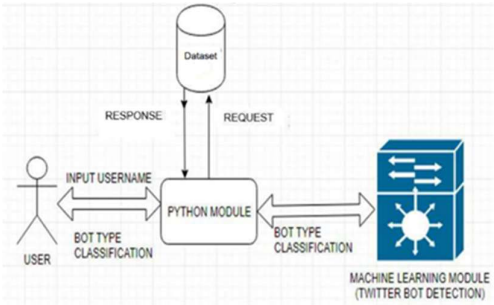

Fig3-Architecture Diagram

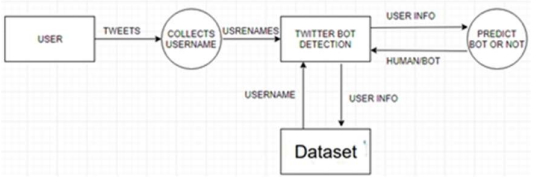

Fig4-DataFlow Diagram

5 IMPLEMENTATION

The project involved implementing machine learning algorithms in Jupyter Notebook. The necessary libraries were imported, and the bot and non-bot categories were defined as 1 and 0, respectively. User account details such as ID, screen name, status, profile pic, friends count, listed count, favourites, etc., were used as attributes, and ID and screen name were converted to binary format. Pearson correlation matrix was used to identify related attributes as in Fig5, Fig 6

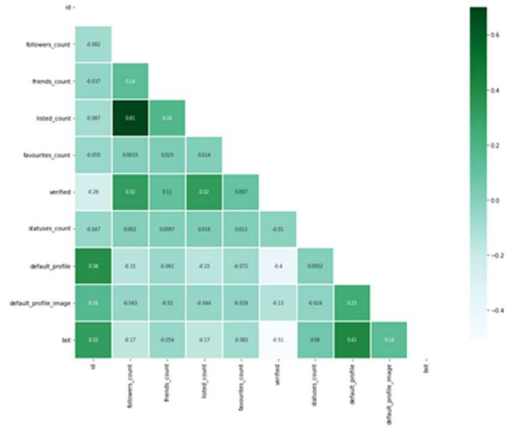

Fig5-pearson Correlation Matrix

	id	followers_count	friends_count	listed_count	favourites_count	verified	statuses_count	default_profile	default_profile_image	bot
0	8.160000e+17	1291	0	10	0	False	78554	True	False	1
1	4.843621e+09	1	349	0	38	False	31	True	False	1
2	4.303727e+09	1086	0	14	0	False	713	True	False	1
3	3.063139e+09	33	0	8	0	False	676	True	True	1
4	2.955142e+09	11	745	0	146	False	105	False	False	1

Fig6-related attributes

and clustered graphs were plotted for user attributes such as friends count, followers count, favorites, and listed count as in Fig7.

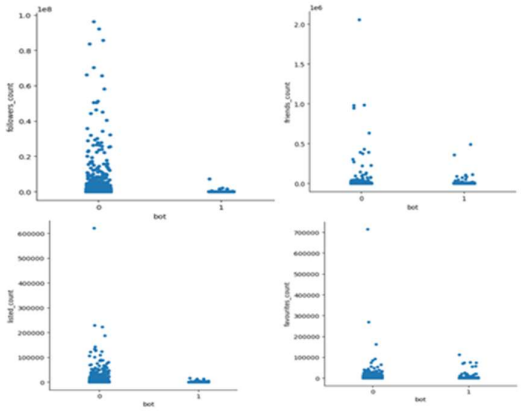

Fig7-graph of attributes like follwers_count,listed_count,friends_count of bot an nonbot

The dataset was trained using KNN, Gaussian, Random Forest, and Decision Tree, with Random Forest having the highest accuracy as shown in Fig8.

```
Modelacc is 95.9203%
 ModelPrecision is 91.4428%
 ModelRecall is 100.0000%
ModelF1 Score is 0.9593
```

Fig8-graphs of KNN,RandmForest,GaussianNB ROC and AUC

The next step involved using an algorithm with a bag of words, where IDs were converted to int from binary using lambda function. Bag of words were collected through research for bot detection, and the results were analyzed using different criteria such as the presence of similar words in profile or description, account verification, listed count, and friends count. The accuracy scores were displayed in Fig 9.

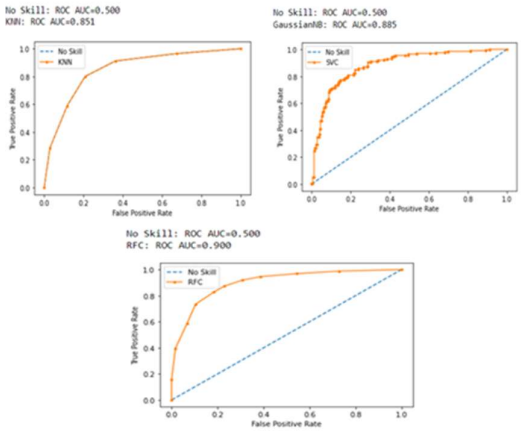

Fig9-Displays the result of the model accuracy and different scores.

A web application was developed for easy usage, where users could provide the required inputs, and the model would analyze them and display the results as shown in Fig10, Fig 11and Fig 12.

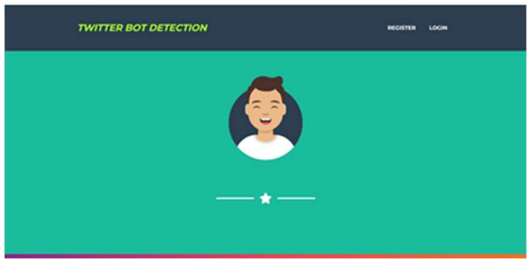

Fig10-Login Page

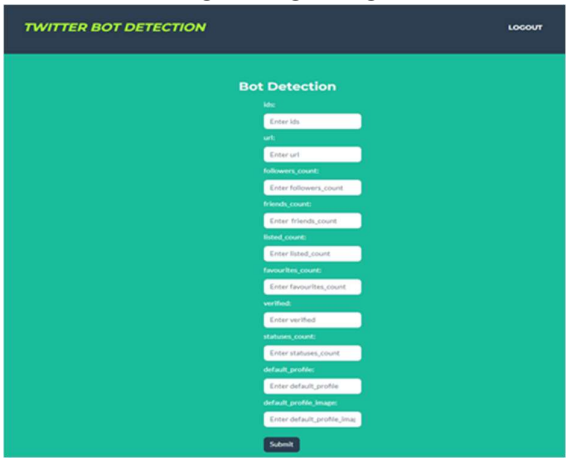

Fig11-Input Page

Fig12-Result Page

6 CONCLUSION

The extensive usage of social media has led to problems including the transmission of harmful material, the spread of incorrect information, and the use of phoney followers for popularity. These problems are frequently linked to using automated or bot accounts. The end goal of our project was to develop a system that could identify whether a Twitter account was a bot or not. Through our system development and machine learning evaluation process, we have drawn several conclusions. We have successfully developed a bot detection system using machine learning with the random forest classifier. To achieve this, we designed and implemented a system that takes an account's username, ID, status, verification, listed count, and number of followers as input from the user account and classifies it as either a human user or a bot. The Random Forest classification algorithm was used to build the model, which achieved an accuracy rate of 95%. We also added a bag of words that included typical and inappropriate words used by bots to improve accuracy, resulting in a 95% accuracy rate. Our random forest algorithm is effective as it can be used for both text-based and image classification. For the future conclusion we need to developed few more models and with using some other features which helps to find the bot in more précised and accurate way.

References:

[1] Wei, F., & Nguyen, U. T. (2019, December). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In 2019 First IEEE International conference on trust, privacy and security in intelligent systems and applications (TPS-ISA) (pp. 101-109). IEEE.

[2] Feng, S., Tan, Z., Wan, H., Wang, N., Chen, Z., Zhang, B., ... & Luo, M. (2022). TwiBot-22: Towards graph-based Twitter bot detection. arXiv preprint arXiv:2206.04564.

[3] Schnebly, J., & Sengupta, S. (2019, January). Random forest twitter bot classifier. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0506-0512). IEEE.

[4] Zahra, A. A., Widyawan, W., & Fauziati, S. (2020). Development of bot detection applications on twitter social media using machine learning with a random forest classifier algorithm. IJITEE (International Journal of Information Technology and Electrical Engineering), 4(2), 66-73.

[5] Ramalingaiah, A., Hussaini, S., & Chaudhari, S. (2021, August). Twitter bot detection using supervised machine learning. In Journal of Physics: Conference Series (Vol. 1950, No. 1, p. 012006). IOP Publishing.

[6] Barhate, S., Mangla, R., Panjwani, D., Gatkal, S., & Kazi, F. (2020, December). Twitter bot detection and their influence in hashtag manipulation. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-7). IEEE.

[7] Kosmajac, D., & Keselj, V. (2019, September). Twitter bot detection using diversity measures. In Proceedings of the 3rd International Conference on Natural Language and Speech Processing (pp. 1-8).

[8] Efthimion, P. G., Payne, S., & Proferes, N. (2018). Supervised machine learning bot detection techniques to identify social twitter bots. SMU Data Science Review, 1(2), 5.

[9] Alothali, E., Zaki, N., Mohamed, E. A., & Alashwal, H. (2018, November). Detecting social bots on twitter: a literature review. In 2018 International conference on innovations in information technology (IIT) (pp. 175-180). IEEE.
[10] Rodríguez-Ruiz, J., Mata-Sánchez, J. I., Monroy, R., Loyola-Gonzalez, O., & López-Cuevas, A. (2020). A one-class classification approach for bot detection on Twitter. Computers & Security, 91, 101715.
[11] Shevtsov, A., Tzagkarakis, C., Antonakaki, D., & Ioannidis, S. (2022). Explainable machine learning pipeline for Twitter bot detection during the 2020 US Presidential Elections. Software Impacts, 13, 100333.
[12] Narayan, N. (2021, September). Twitter bot detection using machine learning algorithms. In 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-4). IEEE.
[13] Van Der Walt, Estée, and Jan Eloff. "Using machine learning to detect fake identities: bots vs humans." IEEE Access 6 (2018): 6540-6549.
[14] Karataş, Arzum, and Serap Şahin. "A Review on Social Bot Detection Techniques and Research Directions." In Proc. Int. Security and Cryptology Conference Turkey, pp. 156-161. 2017.
[15] V.S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, et al., “The DARPA Twitter Bot Challenge,” Computer (Long. Beach. Calif.), Vol. 49, No. 6, pp. 38–46, 2016.
[16] Hagar, Abdulnaser A., et al. "Big Data Analytic Using Machine Learning Algorithms For Intrusion Detection System: A Survey." International Journal of Mechanical and Production Engineering Research and Development (IJMPERD) 10 (2020): 6063-6084.
[17] Elgeldawi, Enas, et al. "Detection and characterization of fake accounts on the pinterest social networks." Int. J. Comput. Netw. Wirel. Mob. Commun 4 (2014): 21-28.
[18] Gavhane, Sachin Prakash, and Vijay Maruti Shelake. "Intrusion Detection System Using Optimal C4. 5 Algorithm." Int. J. Comput. Sci. Eng. Inf. Technol. Res. 4.2 (2014): 5-14.
[19] Naik, N. O. R. A., et al. "Leukemia Prediction Using Random Forest Algorithm." International Journal Of Computer Science Engineering And Information Technology Research 8.3 (2018): 1-8.