

A Project Report on:

Detection Of Bot Accounts On Social Networks Using Big Data Mining Tools

Prepared by :

Admission No.

Student Name

U17CO002

KANEESHA GANDHI

U17CO023

HARSHIT SODAGAR

U17CO028

DEVANSHI BHATIA

U17CO074

JAY RATHOD

Class : B.TECH. IV (Computer Engineering) 7th Semester

Year : 2020-2021

Guided by : DR. DIPTI RANA



DEPARTMENT OF COMPUTER ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY,
SURAT - 395 007 (GUJARAT, INDIA)





Student Declaration

This is to certify that the work described in this project report has been actually carried out and implemented by our project team consisting of

Sr.	Admission No.	Student Name
1	U17CO002	Kaneesha Gandhi
2	U17CO023	Harshit Sodagar
3	U17CO028	Devanshi Bhatia
4	U17CO074	Jay Rathod

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. We understand that our result grades would be revoked if later it is found to be so.

Signature of the Students:

Sr.	Student Name	Signature of the Student
1	Kaneesha Gandhi	
2	Harshit Sodagar	
3	Devanshi Bhatia	
4	Jay Rathod	

Certificate

This is to certify that the project report entitled Detection Of Fake Accounts On Social Networks Using Big Data Mining Tools is prepared and presented by

Sr.	Admission No.	Student Name
1	U17CO002	Kaneesha Gandhi
2	U17CO023	Harshit Sodagar
3	U17CO028	Devanshi Bhatia
4	U17CO074	Jay Rathod

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE:


GUIDE

JURY

HEAD OF DEPT.

Contents

List of Figures	ii
Abstract	iii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Report Organization	2
2 Literature Review	3
3 Proposed Framework and Implementation	7
3.1 Introduction	7
3.2 Block Diagram	7
3.3 Data Preprocessing	9
3.3.1 Identifying missing values and imbalance in the data	10
3.3.2 Feature Independence using Spearman Correlation	12
3.3.3 Implementing different algorithms - Decision Tree Classifier, Multi- nomial Naive Bayes Classifier, Random Forest Classifier, XGB Clas- sifier	13
3.3.4 Implementing Jaccard Similarity Model	14
3.3.5 Check Legitimacy Index	14
3.4 Implementation	15
3.4.1 Dataset	15
3.4.2 Data Preprocessing	15
3.4.3 Decision Tree Classifier	15
3.4.4 Multinomial NB Classifier	16
3.4.5 Random Forest Classifier	17
3.4.6 XGB Classifier	18
3.4.7 Hybrid Ensemble Model	18
3.4.8 Implementation of Jaccard Similarity Model	19
4 Conclusion and Future Work	20
4.1 Conclusion	20
4.2 Future Work	20
References	22
Acknowledgement	23

List of Figures

1	Classifying user accounts as fake or legitimate	9
2	Heatmap of training data	10
3	Distribution of number of friends vs followers	11
4	Friends and followers count representation	11
5	Listed count representation	12
6	Spearman Correlation	13
7	Snapshot of the dataset	15
8	Decision Tree ROC Curve	16
9	Multinomial NB ROC curve	17
10	Random Forest ROC curve	17
11	XG Boost ROC curve	18
12	Voting Classifier ROC curve	19

Abstract

The online social media has completely transformed the way people communicate. However, every revolution brings with it some negative impacts. Due to its popularity amongst tons of global users, these platforms have a huge volume of data. The ease of access with minimal verification of new users on social media has led to the creation of the bot accounts used to collect private data, spread false and harmful content and also poses many security threats. A lot of concerns have been raised with the increment in the quantity of bot accounts on different social media platforms. Also showed a high imbalance between bot and non-bot accounts where the imbalance is a result of 'normal behavior' of bot users. The research aims at identifying the artificial bots accounts on Twitter using various machine learning algorithms and content-based classification based on features provided on the platform and recent tweets of users respectively.

Keywords:*Data imbalance, Online Social Networks (OSNs), Bot Account Detection*

1 Introduction

In this modern era of data dominance, the authenticity of data is a major issue for everyone. With the growing presence of Online Social Networks (OSNs), people have started using it as their preferred medium of communication. Everyone and anyone can use these platforms for sharing their personal information, news, opinions and even their current mood. Some of the most popular OSNs are Facebook, Twitter, Instagram, Google Plus, Reddit and LinkedIn. It is not only limited to individuals using it for personal purposes, but governments, organizations, commercial enterprises and even politicians are using these platforms to increase their reach to the masses. It also makes getting response from the audience much easier while making it convenient to convey their messages directly to a wider population.

But due to popularity of OSNs, they are attractive targets for malicious entities that are trying to exploit the vulnerabilities of these platforms. OSNs have a bulk of fake accounts that are either operated by other humans or by artificial social bots. These fake accounts are generally made to take advantage of the weaknesses of the network and thus the genuine users become victims of these malicious activities.

A social bot is a software to automate user activities. These activities can be generating pseudo posts which look like human generated, re-posting photos, articles etc, adding likes and comments on other posts and increasing their social network by connecting with other accounts. The level of sophistication of these bots ranges from dummy like bots that aggregate information from posts and re-post them to bots that are capable of infiltrating human conversations. Social bots have pros and cons for the users of the OSNs.

Thus, it has become necessary to identify these fake accounts on the OSN platforms to preserve the security and privacy of the users.

1.1 Motivation

In today's world, social media has gained a lot of popularity. It is not just a platform for interacting anymore, it has evolved a lot. As a result, with more users online, there is more data, and this is collected and manipulated for false information with the help of fake accounts. These fake accounts are created for a variety of purposes, for example, collecting personal user data, selling data to third parties, spreading false information about any trending topic/news, etc. Researchers all over the world are trying to identify these fake accounts so that privacy and security of users is preserved.

1.2 Objectives

The primary goal is to detect fake user accounts on social media platforms using a variety of big data mining tools. The aim is to take a random set of user accounts from a social media platform and then a variety of machine learning algorithms are used to filter out the fake users. The implementation will be done on a data set collected from twitter to check the accuracy of the proposed model in the real world.

1.3 Report Organization

Chapter 1 of the report gives a brief introduction, the application of the chosen project in real world, motivation and objective behind the work, and the organization of the report. Chapter 2 comprises of the literature review and the theoretical background related to the project. Chapter 3 of the report comprises of the proposed algorithm and flowcharts of the project. Chapter 4 of the report deals with the simulation and working of this project along with the results. Finally, Chapter 5 recapitulates the report and talks about the possible future work for the project.

2 Literature Review

Mitigating fake accounts has attracted the attention and curiosity of many researchers. Thus, extensive research has been carried out in this direction.

Authors of [1] depicted the feasibility of launching automated attacks in correspondence with identity theft: Profile cloning and cross-site profile cloning. They then provided solutions and suggestions to protect the privacy of users such as providing more information on the authenticity of the issued requests to the receiver and making the CAPTCHAs more difficult to decode. However, sending detailed information about each user request, i.e., country information based on IP address, profile creation date, leads to high computational overhead. Moreover, in terms of CAPTCHA and reCAPTCHA, not only does the human interaction slow down the communication and possibly lead to false positives, but both contribute to a bad customer experience, which is the last thing a website owner wants.

In [3], the writers computed profile similarity after searching and collecting all identity profiles that have name similar to that of given input identity (IID) and a Profile Set. In case, the similarities found for a particular IID was larger than the prescribed thresholds, then it was added to the Suspicious Identity List (SIL) and after suitable verification, it was declared either fake or genuine. The problem with this approach, however, was that the use of similarity measures do not consider the strength of network of friendships shared among users. But in reality, it is believed that the more the connection of shared network between two users, the greater is their similarity.

Kontaxis et al. [4] proposed a modular approach for detection of fake accounts on social networks. The key concept behind its logic was that it utilized user-specific (or user-identifying) information, which was obtained from the user's original social network profile and this information was used to locate similar profiles across all social networking sites. After obtaining results, depending on the rarity of the profile, suitable methods of inspection for suspects was carried out. Finally, the user was presented with a list of possible profile clones and a score indicating their degree of similarity with his own profile.

In [14], the efficiency of detecting fake accounts on social networks was increased by calculating the similarities between user accounts using graph adjacency matrix and then applying the Principal Component Analysis (PCA) algorithm for feature extraction. After that, Synthetic Minority Over-sampling Technique (SMOTE) was used for data balancing. Subsequently, linear Support Vector Machine (SVM), Medium Gaussian SVM

and regression, and logistic algorithms were used to classify the nodes. At the end, different classifier algorithms were implemented for evaluating the performance of the above scheme. Weakness of this proposed model however was that it required the fake accounts to work in the network in order to organize them as legitimate or fake ones, by surveying their friend's networks. Detecting fake accounts before any user activity was hence not possible by this scheme.

A novel unsupervised method of recognizing and segregating bot accounts from legitimate user accounts was presented in [7]. This approach identified bots using correlated user activities. The two-fold procedure included: (1) developing the warped correlation finder and (2) using this finder to detect bots. This system called DeBot guaranteed high precision and was able to detect bots that other methods failed to spot. The presence of highly synchronous cross-user activities revealed abnormalities and was a key to detecting automated accounts according to this scheme. This modular approach however, required iteration over three independent parameters (Base window, number of buckets, maximum lag) each time, while keeping the other parameters fixed. This in turn increased computational oncosts and made the whole system complex.

Jennifer Golbeck et al. [15] proposed a mechanism of detecting bots on social media platforms using Benford's Law. Application of Benford's Law in social networks was done by inspecting the friend count for each account's friends. The main idea was that an account's friends' friend counts should follow Benford's expected distribution. Moreover, the person's friends who were drawn from this distribution were presumed to follow this too. The hypothesis behind this work was that social connections made by bots would be unnatural and would thus violate Benford's Law. In this paper, it was proposed that successful comparison of distribution of first significant digits (FSDs) in an account's network to the expected distribution of Benford's Law could be done using the Pearson chi-square test. The p-value obtained by these tests was treated as a measure of adherence to the Benford's Law. High p-values indicated close matching of a node's FSD distribution with the Benford distribution. On the contrary, very low p-value depicted poor match. This model paved a way for detecting bots on social media platforms and due to close relation with other distributions like Zipf's Law and the Pareto Distribution, it opened new doors for detection of fraudulent behavior on social media platforms. The main limitation of this work was the sample size. The size of retweet and like bot samples were quite small due to cost and time necessary to validate the data.

A categorization method was proposed by Erşahin et al. [10] to detect spam accounts on Twitter. Manual collection of datasets was done. Username, number of friends and

followers, profile and background image, content of tweets, number of tweets, description of account, etc., were analyzed. The experiment consisted of 501 fake and 499 real accounts, where 16 features from the information that were obtained from Twitter APIs were identified. Fake accounts were classified using two experiments. The first experiment used the Naïve Bayes learning algorithm on the Twitter dataset which includes all aspects without discretization and the second experiment performed Naïve Bayes learning algorithm on Twitter dataset after discretization.

Meda et al. [9] put forth a technique which utilized sampling of non-uniform features inside a machine learning algorithm by the adaptation of random forest algorithm to recognize spammer insiders. Integration of bootstrap aggregating technique with unplanned selection of features is incorporated into this scheme. The dataset collected was based on indefinite user behaviors in order to test the performance of random forest algorithm. The features were divided into 2 sub categories, namely, domain expert selection and random selection. The aim was to reproduce two contradictory situations during feature selection. The first group included domain experts for the feature choice and the second group involved random selection of features. The outputs received reveal the power of enriched feature sampling technique.

Gharge et al. [11] initiated a method, which was classified on the basis of two new features. First was the recognition of spam tweets without any information regarding users and second was exploration of language for spam detection on Twitter trending topic at that particular time. The entire model processing consisted of the following steps: (1) Tweet collection in correspondence to the prevailing trending topics on Twitter. After their storage, the tweets were analyzed. (2) Labelling of spam was performed across all datasets for detecting malignant URL. (3) Feature extraction using language as a tool for segregating fake tweets from real ones. (4) Classification of dataset was done in order to train and instruct the model for acquiring knowledge for spam detection. (5) Finally, tweets are taken as input and determined whether they are spam or non-spam. The experimental setup was prepared for determining the accuracy of the system.

A machine learning framework was presented by [8] that leveraged a combination of network, temporal features and metadata to identify the extremist users, and predicted content adopters and interaction reciprocity in social media. They used a distinct dataset which contained several tweets which were generated by thousands of users who were manually reported, identified and/or suspended by Twitter because of their involvement with extremist campaigns. They used learning models like Logistics regression and Random forest for the same. In correspondence to the issues presented by the above-mentioned schemes, we establish an efficient model for the detection of fake accounts on social net-

works using big data mining tools.

The author in [5] first shed light into the fact that twitter has become a very attractive target for bots to abuse in the past few years. The author collected data by crawling twitter and based on the recognized features collected through this data, humans, bots and cyborgs can be differentiated on twitter. This is done by observing that humans have complex performance or high entropy while bots and cyborgs have periodic timing or low entropy.

The author [2] extracted graph-based features like the number of friends and followers for twitter users and also identified relationships between them. They also used the content of users' tweets and applied various algorithms on the content-based and graph-based features for classification.

The author [12] demonstrated a framework for detecting bots on twitter using a system based on machine learning that extracted thousands of features based on six classes: tweet content and sentiment, network patterns, users and friends meta-data, and activity time series [6].

Moving to bot detection at the tweet-level, and therefore having training large data orders, made the issue of bot detection way more susceptible to the use of deep learning models [13]. Such techniques benefited from the huge quantities of labeled data, displaying very good performance in several contexts where such resources were obtainable from mastering games to image classification.

In correspondence to the issues presented by the above-mentioned schemes, this research establishes the requirement of an efficient model for the detection of bot accounts on social networks using big data mining tools.

3 Proposed Framework and Implementation

3.1 Introduction

We have implemented the machine learning algorithms of algorithms such as Naïve Bayes Classifying algorithm, Clustering Classifier, Decision Tree, Hybrid Classifier, and various other custom algorithms based on what is required. The primary goal is to detect bot accounts on social media platforms using a variety of big data mining tools. The aim is to take a random set of user accounts from a social media platform and to filter out the bot accounts using machine learning algorithms. Word Embedding techniques using Python are also carried out to find the similarities between tweets of bot accounts to classify them as non-legitimate. The experimentation is performed on a labelled data set collected from Kaggle (Jain, 2019) to check the accuracy of the proposed model.

In today's world, social media has gained a lot of popularity. It is not just a platform for interacting any-more, it has evolved a lot. As a result, with more users online, there is more data, and this is collected and manipulated for false information through the help of non-legitimate accounts. These accounts are created for a variety of purposes, for example, collecting personal user data, selling data to third parties, spreading false information about any trending topic/news, etc. Researchers all over the world are trying to identify these fake accounts so that privacy and security of users is preserved. The main motive of this project is to detect bot accounts on social media platform twitter.

The following are the different stages of our model:

1. Identifying the missing data
2. Identifying imbalance in the data
3. Implementing different algorithms - Decision Tree Classifier, Multinomial Naive Bayes Classifier, Random Forest Classifier, XGB Classifier. Implementing Hybrid Ensemble Learning model using the weak learning models
4. Implementing Jaccard Similarity test
5. Combining the result of both the models for final prediction

3.2 Block Diagram

As mentioned, there is a need to identify the bot accounts on social media. Here, proposed a model that distinguishes between bot accounts and legitimate user accounts.

This section describes the methodology for the implementation of bot detection system model using Twitter dataset. The pictorial representation of the same is given in Figure 1 below. The following sections show the different stages of the proposed framework:

1. **Input Username:** The user of the proposed framework needs to verify the particular username is bot user or legiti-mate user then he/she has to enter the username in the available user interface. The user interface will check whether the username is available on twitter or not. If yes, then follows the next step, otherwise provide the similar username for the selection.
2. **User Data Extraction:** The proposed framework is requiring the features and the tweets available for the given user, so all the features available on the twitter will be extracted along with the tweets of that user for the further steps. Also, this user will be tested with the available tweeter users, so if the system is used first time, then the same information will be fetched for the maximum users possible. And this in-formation will be kept for the further usage and later only some few latest users information will be collected for the efficient result. The data avail from the twitter is not directly usable by the machine learning algorithm, so further the proposed framework follows the step of data prepro-cessing to prepare the data for the remaining steps.
3. **Data Preprocessing:** The users' features and tweets are required to process before they apply to the machine learning algorithm to make it usable, by performing the following steps.
 - (a) **Identifying the missing data:** Missing data are defined as any values that are not available or entered by the user. A heat map is plotted to showcase which all attributes contain what frequency of null/missing values for further action on them. As shown here in Figure, "location", "description", and "url" have the maximum missing values (depicted in yellow) whereas "status" and "has_extended_profile" has only a few of them.
 - (b) **Identifying imbalance in the data:** The data analysis shows that whenever the listed count is between the 10,000 to 20,000 from that 5% of them are bots and 95% are non bot/legitimate user accounts as shown in Figure. This sets the imbalance in ecosystem. Similarly for the verified case as well, an observation of data imbalance for bots' and non bots' friends and followers was made. The Figure depicts the imbalance of accounts classified as bot and non-bot. This imbalance of accounts, forced the important heuristics to apply which is discussed in step 5.

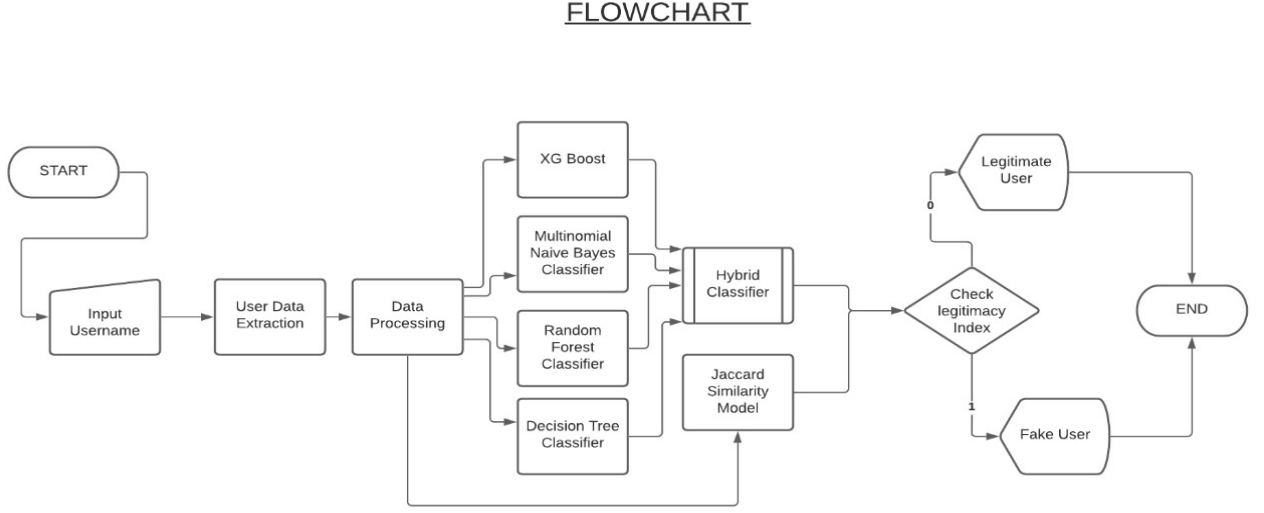


Figure 1: Classifying user accounts as fake or legitimate

This is a model that distinguishes between bot accounts and legitimate user accounts. This section describes the methodology for the implementation of bot detection system model using Twitter dataset.

3.3 Data Preprocessing

We collected user data from Twitter API for training our model. The `get_user()` method of the API class in Tweepy module is used to get information of the specified user. After analyzing and removing the unnecessary fields of user data, we prepared our sample dataset. This was done using Tweepy and CSV libraries.

The following image shows one of sample user data obtained.

Labelled Twitter dataset has been extracted from Kaggle for training our model. We then divided the data into two datasets - Bot dataset and Non-bot dataset for performing exploratory data analysis. Exploratory data analysis is a process where initial investigation is performed on the data to discover patterns, spot anomalies, check assumptions etc with the help of graphical representations. We have performed exploratory data analysis for both bot and non-bot datasets using the following algorithm -

1. Identifying missing values and imbalance in the data.
2. Feature extraction.
3. Feature engineering.
4. Dropping unnecessary attributes.

3.3.1 Identifying missing values and imbalance in the data

For identifying the missing values in the data, `get_heatmap()` function is used from the seaborn library. In Figure 3, yellow colour represents the missing values and purple represents the filled/not missing values. The heatmap clearly shows that maximum number of missing values are present in the attributes - location, description and url. There are a few missing values in status and has_extended_profile columns as well.

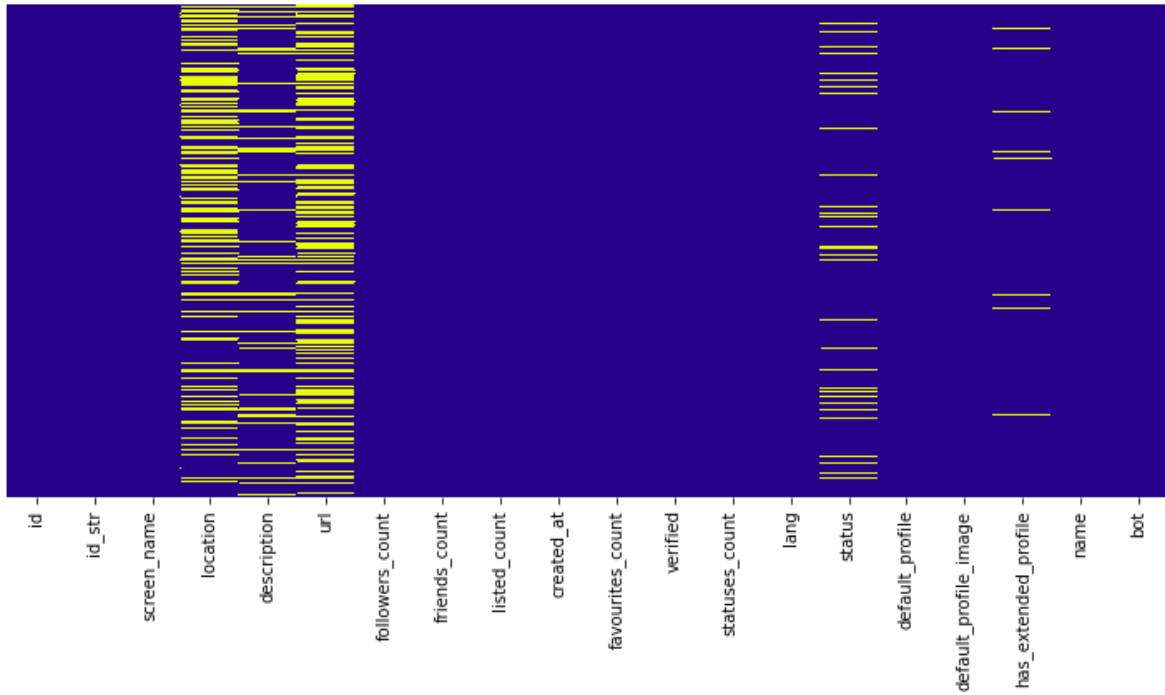


Figure 2: Heatmap of training data

By plotting the number of friends vs followers for bots and non-bots(as shown in Figure 4), we concluded that bots follow more people and have less number of friends. Whereas non-bot accounts have a balanced number of followers and friends. Hence followers and friends can be used as an indication of differentiation between bot and non-bot accounts.

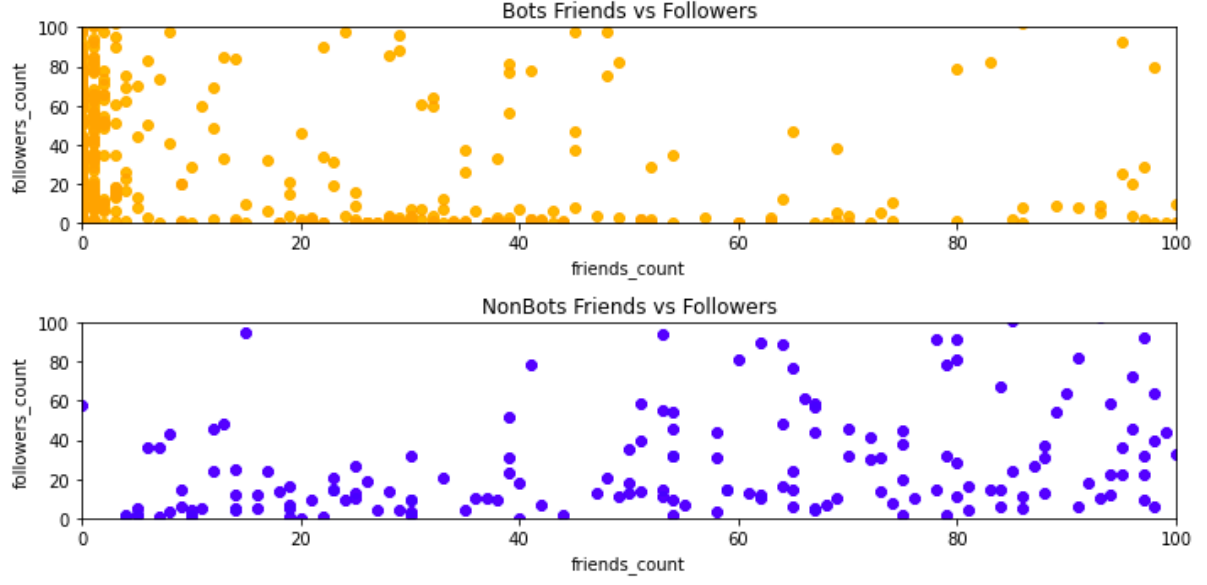


Figure 3: Distribution of number of friends vs followers

By plotting and comparing the listed_count, friends_count and followers_count for bot and non-bots accounts, we found a lot of imbalance in the data. Imbalance is clear in the following figures.

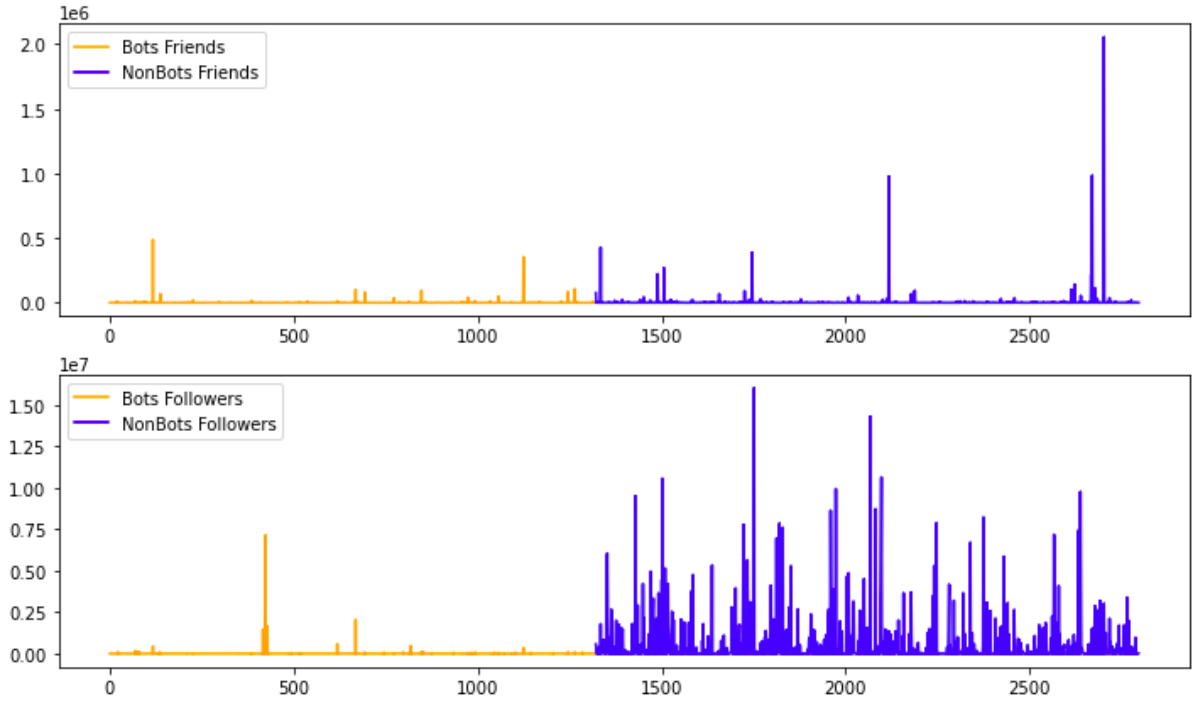


Figure 4: Friends and followers count representation

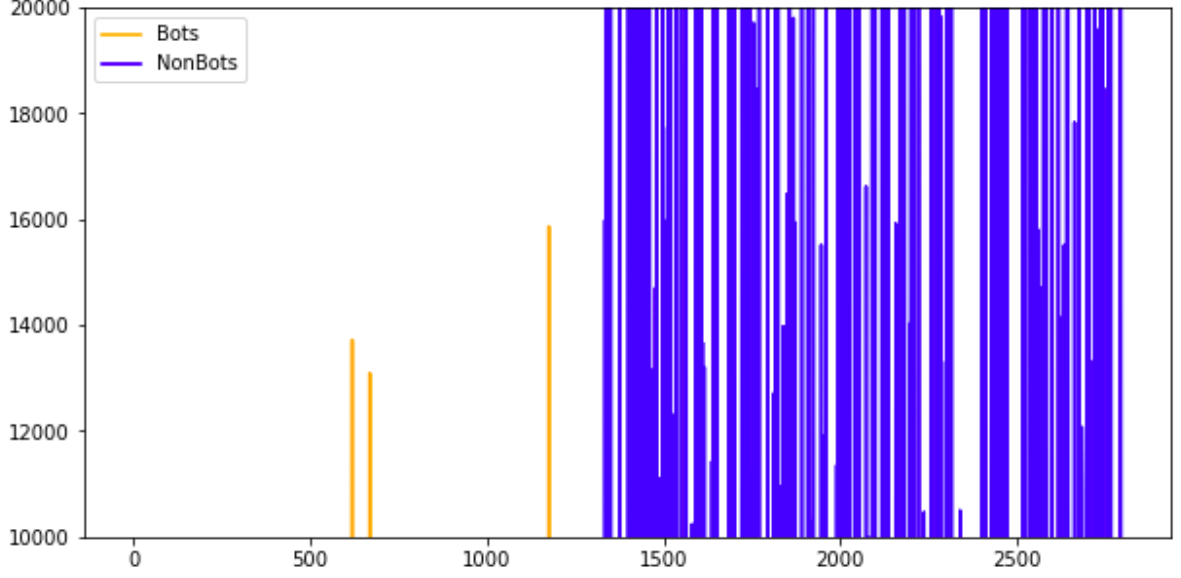


Figure 5: Listed count representation

3.3.2 Feature Independence using Spearman Correlation

To decide the attributes that we need to consider for the required algorithm we used the Spearman's rank-order correlation technique. The Spearman's rank-order correlation is the nonparametric form of the Pearson product-moment correlation. Spearman's correlation coefficient measures the strength and direction of relationship between two ranked variables.

Spearman's correlation, unlike Pearson's correlation, works on determining the direction and strength of the monotonic relationship between two variables. Pearson's correlation, on the other hand, works on determining the direction and strength of the linear relationship between two variables. A monotonic relationship is one where the values of the two variables are such that if one increases, the other also increases, or if one increases, the other decreases.

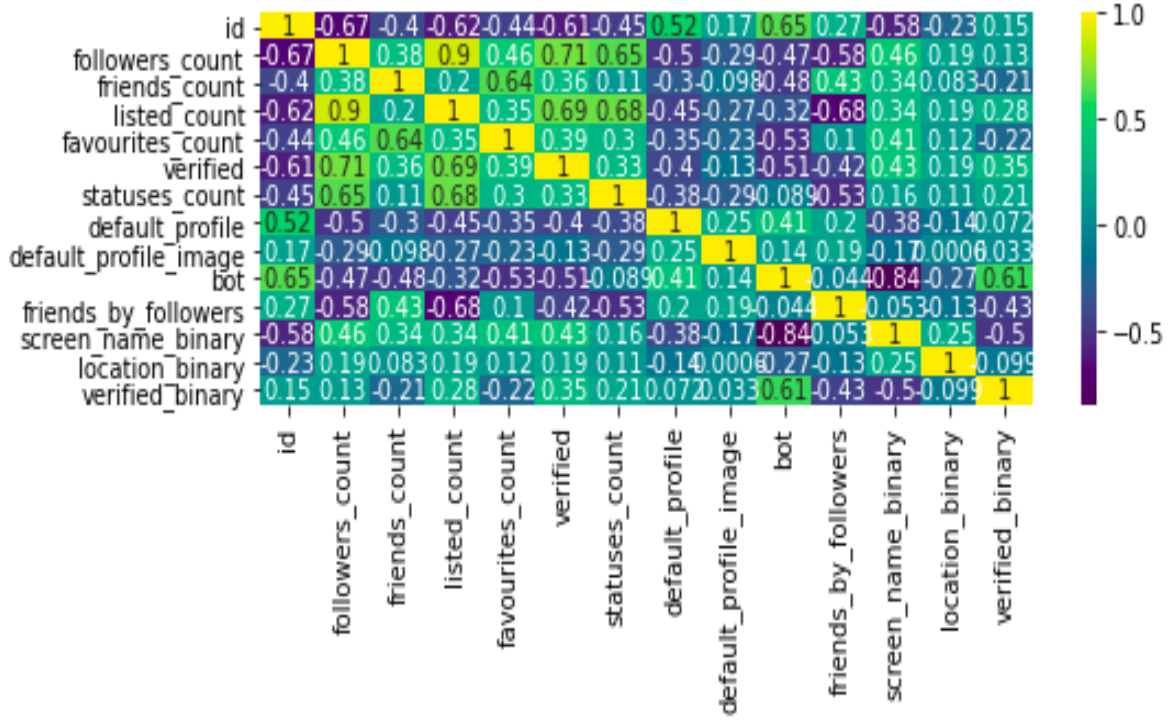


Figure 6: Spearman Correlation

From the resultant table we can conclude that: There is no correlation between id, statuses_count, default_profile, default_profile_image and target variable. There is strong correlation between verified, listed_count, friends_count, followers_count and target variable. We cannot perform correlation for categorical attributes. So we will take screen_name, name, description, status into feature engineering, while use verified, listed_count for feature extraction.

Further, as a part of feature engineering, we created a "bag of words" model which identifies if the account is a bot account or not. To accomplish this we converted the screen name, name, description and status into binary vectors using our own vectorizing algorithm. At the end we got the entire training data converted into binary format. Finally as a part of feature extraction, we discarded the unnecessary features after the spearman correlation test and kept only those with high correlation rank value.

3.3.3 Implementing different algorithms - Decision Tree Classifier, Multinomial Naive Bayes Classifier, Random Forest Classifier, XGB Classifier

On the obtained features from the feature engineering and feature extraction process, the variety of classifiers are considered due to their typical characteristics and mix set of attributes. The categorical classifiers considered are Decision Tree Classifier, Multinomial NB classifier, which alone generate poor testing accuracy. The Random Forest Classifier is also considered here though it is complex. Together with the XGB classifier considered

which can show the highest training accuracy, precision and recall among all. Thus, emphasized to have the hybrid ensemble learning model and used the above mentioned algorithms as weak learning models.

3.3.4 Implementing Jaccard Similarity Model

Here, for the proposed framework to identify the user as legitimate or bot, the heuristic considered is that the tweets posted by a bot account should be similar to each other as they are created with a certain agenda. So, together with the classification, the similarity between tweets of a user given account with others is also performed to find out accurate classification of the account into bot or not-bot account.

Generally to find similarity between the texts, cosine similarity or Jaccard similarity can be used. Jaccard similarity takes solely a novel set of words from each document or sentence. In contrast, cosine similarity considers the overall vector lengths. Jaccard similarity is preferred for the cases where repetition of words does not make a difference. In case of significant duplication of words, cosine similarity should be used. In this case, context makes a lot more difference than duplication. Hence the ideal technique to be used is Jaccard similarity.

Jaccard similarity is a statistical method used to find the diversity and similarity between documents using a set of unique words used in the documents.

It measures the proportion of the number of common words in two documents. It is defined as the intersection of the two documents over the union of the two documents as shown in equation 1.

$$J(Doc_1, Doc_2) = \frac{Doc_1 \cap Doc_2}{Doc_1 \cup Doc_2} \quad (1)$$

3.3.5 Check Legitimacy Index

The final step is to check the legitimacy index which is the combination of both Jaccard similarity index ranging from 0 to 1 and classification true positive rate ranging from 0 to 1. So, here considered the average ceiling of both which is approx. 0 or 1 and based on this value, the user is classified as bot user, if this index is 1, otherwise classified as legitimate user.

3.4 Implementation

3.4.1 Dataset

To assess the proposed framework, here the training dataset is obtained from Kaggle (Jain, 2019). The training dataset consists of 1,321 bot instances and 1,476 non-bot instances. The training dataset sample, with a few columns and rows is shown in the figure below.

	id	id_str	screen_name	location	description	url	followers_count	friends_count	listed_count
0	8.160000e+17	"815745789754417152"	"HoustonPokeMap"	"Houston, TX"	"Rare and strong PokŽmon in Houston, TX. See m..."	"https://t.co/dnWuDbFRkt"	1291	0	10
1	4.843621e+09	4843621225	kernyeahx	Templeville town, MD, USA	From late 2014 Socium Marketplace will make sh...	NaN	1	349	0
2	4.303727e+09	4303727112	mattlieberisbot	NaN	Inspired by the smart, funny folks at @replyal...	https://t.co/P1e1o0m4KC	1086	0	14
3	3.063139e+09	3063139353	sc_papers	NaN	NaN	NaN	33	0	8

Figure 7: Snapshot of the dataset

3.4.2 Data Preprocessing

Preprocessing of the data was done to clean the tweets and the text attributes available in the data like screen_name, name, description and status. For preprocessing links, images, hashtags, @mentions, emojis, stop words and punctuations were removed. Contractions were expanded, for example, “what’s” was converted to “what is”. Stemming and tokenizing was done later on as a part of preprocessing.

3.4.3 Decision Tree Classifier

A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter. In Decision Tree Classification a new example is classified by submitting it to a series of tests that determine the class label of the example. These tests are organized in a hierarchical structure called a decision tree. For our dataset, we have utilized the decision tree model from the sklearn library in Python. After applying the decision tree model, we have obtained a training accuracy of 88.2% and a testing accuracy of 87.85%.

It also gives training and testing precision of 90.5% and 91.1% respectively. Moreover, the recall obtained for training and testing data is 84.4% and 83.6% respectively.

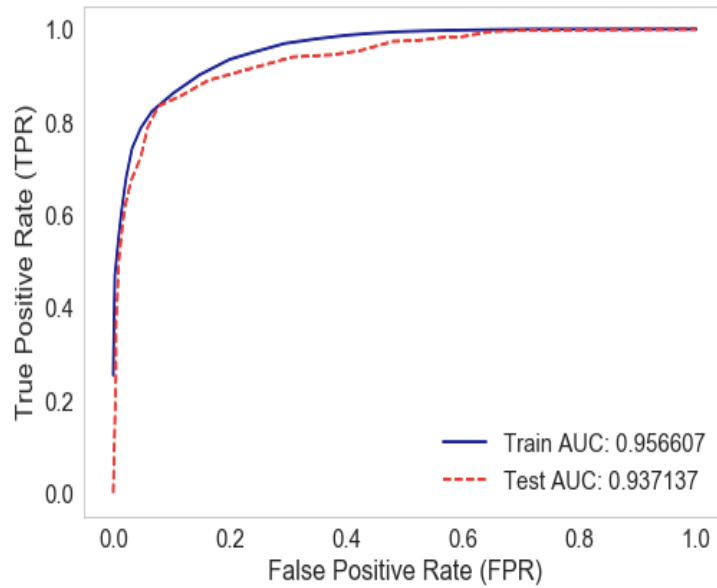


Figure 8: Decision Tree ROC Curve

The below graph is a depiction of Area Under the Receiver Operating Characteristics (AUROC) for the Decision Tree Classifier model. As we know AUC - ROC curve is a performance measurement for classification problem at various thresholds settings, where ROC is a probability curve and AUC represents degree or measure of separability. Since AUC is close to 1, Decision Tree gives good performance, however it may be overfitting as it is 0.937. Hence, there is scope of trying other machine learning models.

3.4.4 Multinomial NB Classifier

The training accuracy was found to be 67.8% and testing accuracy was found to be 69.7%. It also gives training and testing precision of 59.3% and 62.5% respectively. Moreover, the recall obtained for training and testing data is 96.2% and 97.1% respectively. Figure below shows the ROC Curve for the Multinomial NB Classifier.

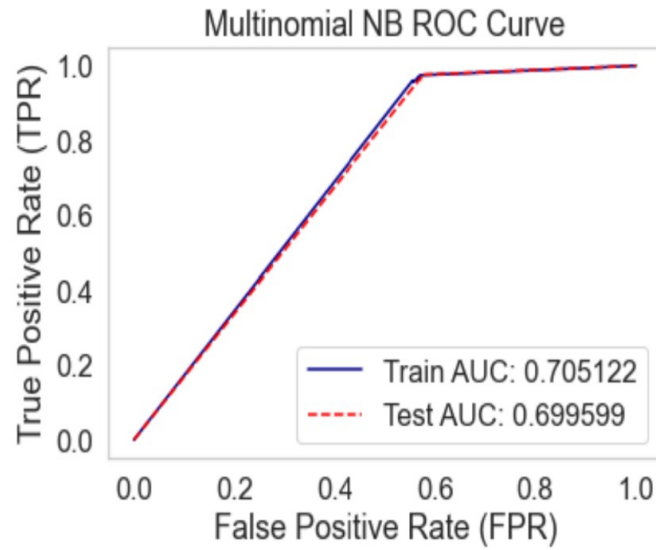


Figure 9: Multinomial NB ROC curve

3.4.5 Random Forest Classifier

The training accuracy was found to be 84.8% and testing accuracy was found to be 84.4%. It also gives training and testing precision of 86.5% and 87.6% respectively. Moreover, the re-call obtained for training and testing data is 79.6% and 79.8% respectively. Figure below shows the ROC Curve for the Random Forest Classifier.

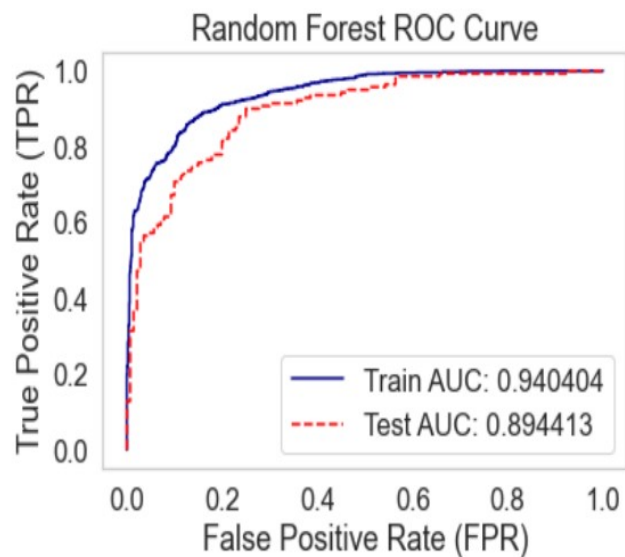


Figure 10: Random Forest ROC curve

3.4.6 XGB Classifier

The training accuracy was found to be 98.8% and testing accuracy was found to be 83.5%. It also gives training and testing precision of 99.2% and 84.5% respectively. Moreover, the re-call obtained for training and testing data is 98.2% and 82.1% respectively. Figure below shows the ROC Curve for the XG Boost Classifier.

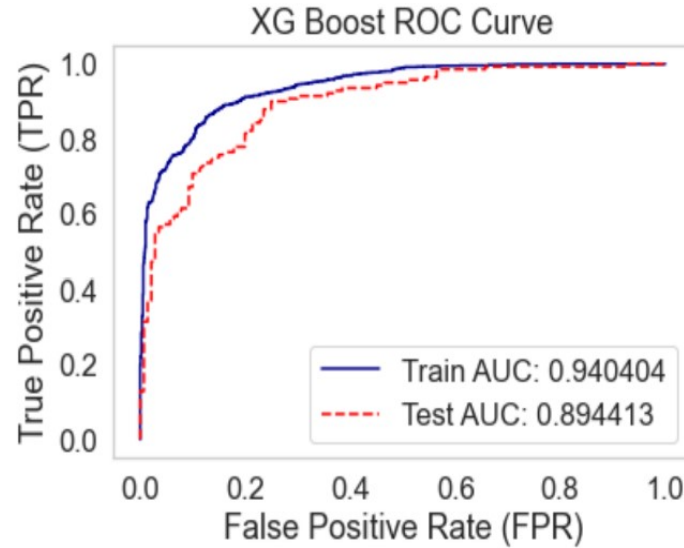


Figure 11: XG Boost ROC curve

3.4.7 Hybrid Ensemble Model

The above mentioned algorithms were used as weak learners to build a hybrid ensemble learning model. The training accuracy was found to be 92.3% and testing accuracy was found to be 90.0%. It also gives training and testing precision of 91.1% and 88.8% respectively. Moreover, the recall obtained for training and testing data is 92.7% and 91.4% respectively. Figure below shows the ROC Curve for the Hybrid Ensemble Model.

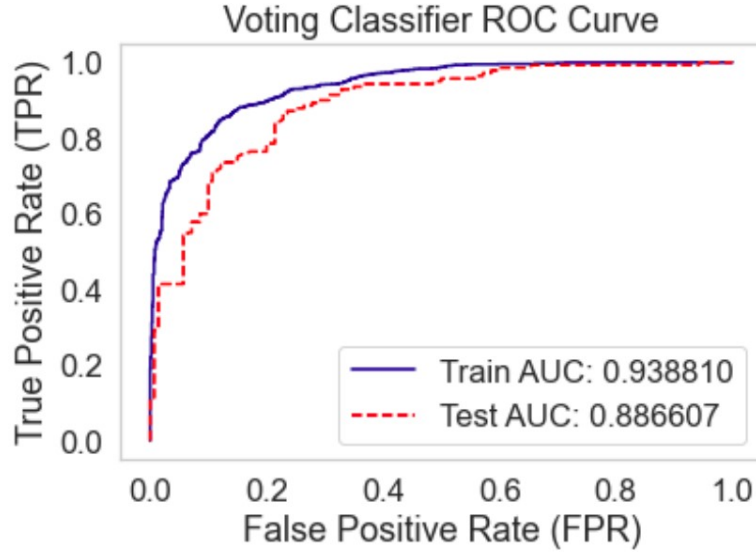


Figure 12: Voting Classifier ROC curve

3.4.8 Implementation of Jaccard Similarity Model

Here, to find similarity between the tweets, Jaccard similarity is used. In this similarity test, the experiment depicted the Jaccard similarity score between the 100 most recent tweets of a given Twitter account. Before the similarity score was computed, the tweets made available from the dataset were in their raw form and had to be preprocessed first for effective use.

Further, the tokenizing of the tweets performed using RegExp Tokenizer and lemmatized the tweets using WordNet Lemmatizer from NLTK library. After this, the Jaccard Similarity score was computed. The accuracy up to 93.2% is achieved after combining the results from both the Hybrid model and the Similarity model.

4 Conclusion and Future Work

4.1 Conclusion

This research addresses the issue of presence of artificial bot accounts on the social media platform particularly Twitter and the threat they poses to the privacy of the legitimate users. Social media has become a major source of information and many bot accounts are created with the purpose of spreading misinformation. The primary focus of the research is to build a model that identifies the bot accounts on Twitter which are imbalanced compared to normal user accounts. In the proposed model various machine learning algorithms were used as weak learners to make a hybrid model that could successfully classify 90% of the accounts using the preprocessed Twitter data. The project also consists of classification based on Jaccard Similarity model which uses recent tweets posted by the users. This content based classification along with the hybrid model was able to classify 93.2% accounts correctly. With the help of the model these bot accounts can be identified and removed from the platform to reduce the harm to the society.

4.2 Future Work

Some methods, such as guided learning approaches, were extensively discussed in this work. To comprehend, reinforce or discover new results, several approaches require more exploration. It stimulates the research community to discover new approaches and improve existing approaches. The above model can be utilized for real time applications. With the awareness of bot accounts among users, it would become convenient and highly lucrative for people and organizations to detect them on OSNs. In future, this model will be compared with other available techniques and by including the usage of network information of users.

References

- [1] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, “All your contacts are belong to us: Automated identity theft attacks on social networks,” in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09, Madrid, Spain: Association for Computing Machinery, 2009, pp. 551–560, ISBN: 9781605584874. DOI: 10.1145/1526709.1526784. [Online]. Available: <https://doi.org/10.1145/1526709.1526784>.
- [2] A. H. Wang, “Detecting spam bots in online social networking sites: A machine learning approach,” in *IFIP Annual Conference on Data and Applications Security and Privacy*, Springer, 2010, pp. 335–342.
- [3] L. Jin, H. Takabi, and J. B. Joshi, “Towards active detection of identity clone attacks on online social networks,” in *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '11, San Antonio, TX, USA: Association for Computing Machinery, 2011, pp. 27–38, ISBN: 9781450304665. DOI: 10.1145/1943513.1943520. [Online]. Available: <https://doi.org/10.1145/1943513.1943520>.
- [4] G. Kontaxis, I. Polakis, S. Ioannidis, and E. P. Markatos, “Detecting social network profile cloning,” in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2011, pp. 295–300.
- [5] S. Dehade and A. Bagade, “A review on detecting automation on twitter accounts,” *Eur. J. Adv. Eng. Technol*, vol. 2, no. 2, pp. 69–72, 2015.
- [6] A. Bessi and E. Ferrara, “Social bots distort the 2016 us presidential election online discussion,” *First Monday*, vol. 21, no. 11-7, 2016.
- [7] N. Chavoshi, H. Hamooni, and A. Mueen, “Debot: Twitter bot detection via warped correlation,” in *Icdm*, 2016, pp. 817–822.
- [8] E. Ferrara, W.-Q. Wang, O. Varol, A. Flammini, and A. Galstyan, “Predicting online extremism, content adopters, and interaction reciprocity,” in *International conference on social informatics*, Springer, 2016, pp. 22–39.
- [9] C. Meda, E. Ragusa, C. Gianoglio, R. Zunino, A. Ottaviano, E. Scillia, and R. Surlinelli, “Spam detection of twitter traffic: A framework based on random forests and non-uniform feature sampling,” in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, 2016, pp. 811–817.
- [10] B. Erşahin, Ö. Aktaş, D. Kılınç, and C. Akyol, “Twitter fake account detection,” in *2017 International Conference on Computer Science and Engineering (UBMK)*, IEEE, 2017, pp. 388–392.

- [11] S. Gharge and M. Chavan, “An integrated approach for malicious tweets detection using nlp,” in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, IEEE, 2017, pp. 435–438.
- [12] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, “Online human-bot interactions: Detection, estimation, and characterization,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, 2017.
- [13] S. Kudugunta and E. Ferrara, “Deep neural networks for bot detection,” *Information Sciences*, vol. 467, pp. 312–322, 2018.
- [14] M. Mohammadrezaei, M. E. Shiri, and A. M. Rahmani, “Identifying fake accounts on social networks based on graph analysis and classification algorithms,” *Security and Communication Networks*, vol. 2018, 2018.
- [15] J. Golbeck, “Benford’s law can detect malicious social bots,” *First Monday*, vol. 24, no. 8, Aug. 2019. DOI: 10.5210/fm.v24i8.10163. [Online]. Available: <https://journals.uic.edu/ojs/index.php/fm/article/view/10163>.

Acknowledgement

We take this opportunity to express heartfelt gratitude to our project guide, Dr. Dipti P. Rana, Assistant Professor in Computer Engineering Department, SVNIT Surat for her valuable guidance, constant encouragement, and helpful feedback all throughout various stages of work.

We would also like to thank our Head of Department, Dr. M. A. Zaveri, Computer Engineering Department for all the support. We are very grateful to SVNIT Surat and its staff for providing us with this opportunity which aided us in acquiring required knowledge to be successful in our work.