COMPARISON OF SOCIAL BOT DETECTION TECHNIQUES

By

BHAGYASRI VALLABHANENI

Bachelor of Technology in Computer Science

GITAM UNIVERSITY

Hyderabad, Telangana

2017

Submitted to the Faculty of the
GraduateCollege of the
OklahomaStateUniversity
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2019

COMPARISON OF SOCIAL BOT DETECTION

TECHNIQUES

Thesis Approved:

Dr. K. M. George

ThesisAdviser

Dr. Johnson P Thomas

Dr. EsraAkbas

ACKNOWLEDGEMENTS

I would like to extend my gratitude to the Computer Science Department at Oklahoma State University for providing me an opportunity to learn and gain knowledge in my area of interest.

My sincere thanks to my Advisor Dr. K. M. George, head of the Computer Science Department, for providing his immense support and guidance, andfor enlightening me with remarkable ideas throughout my research.

A special thanks to the committee members, Dr. Johnson P Thomas and Dr. EsraAkbas for providing their guidance and significant insights.

My profound gratitude to my parents, Mr. Subbarao and Mrs. Subhasri, and my brother Rohith Vallabhaneni for being my pillars of strength, believing in me and supporting me throughout my work.

Name: BHAGYASRI VALLABHANENI

Date of Degree: DECEMBER 2019

Title of Study: COMPARISON OF SOCIAL BOT DETECTION TECHNIQUES

Major Field: COMPUTER SCIENCE

Abstract: Online Social Networks act as a major platform for communication. The origin of social bots is one of the consequences of increasing popularity and utilization of social networks by people. A social bot is an automated application that clones the behavior of a human and creates a faux impression on real users. TheSocial bot can be classified as either benign and malicious based on their actions. Benign bots are used to perform tasks a lot quicker than humans, sharing vital information like weather reports, etc. Whereas, malicious bots begrime the social media with false information and may also be involved in malicious activities such as spamming, stealing private information, creating noise within the conversations, etc. This nature of bots led to the necessity of social bot detection techniques.

Various social bot detection techniques have been proposed based on different algorithms. In this research, proposed social bot detection techniques are reviewed and several of them are implemented. A comparison of these techniques based on their input requirements, approach, and accuracy is performed. The implementation of the techniques has been applied to three completely different data sets collected from the Twitter social network. Four metrics: precision, recall, accuracy, and Cohen's Kappa coefficient are calculated using the results obtained by implementing the techniques. These metrics have been used to decide the efficiency of techniques and provide a comparison of them.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The utilization of online social networks has increased expeditiously ever since their evolution. 72% of America's population use at least one online social network such as Twitter, Facebook and LinkedIn [1]. The enthralling features of social networks have led to an increase in their popularity and usage. The estimation of active users on Twitter from the beginning of January 2019 to the end of April 2019 is 138 million [2].

The rise in the popularity of social networks has also given rise to the existence of social bots. A social bot is an application that simulates the actions of a legitimate user on social media. The accounts managed by these bots are referred to as spam accounts or autonomous accounts. The number of such autonomous accounts on social media is increasing rigorously. The number of active autonomous accounts on twitter is estimated to be between 9 to 15 percent [3]. Every year there has been an increase in the number of spam accounts detected on Twitter. The estimation of detected spam accounts on Twitter in the last years was 6.4 million and 9.9 million in December 2017 and May 2018 respectively [4].

Social bots can be categorized into benign and malicious bots based on their actions [5]. Benign bots are used mainly for sending automatic responses, sharing important information such as weather, news, etc. In contrast, malicious bots are created with a motive to causedestruction.

With a fake identity, they steal data, perform spam activities, mislead people by spreading false information and create noise during the debates. Therefore, the origin of social bots has both advantages and disadvantages associated with it.

On a positive side, social bots can perform tasks much faster than humans, they help in saving time and act as productive customer service agents. The bots like Siri, Google Assistant and Alexa are used for improving customer engagement. On the other side, the malicious bots can have disadvantages such as:

- One of the malicious activities is astroturfing. It is an act of creating a faux impression on real users [6]. Social bots can create a large impact on political affairs [7]. 3.8 million tweets were tweeted by 400,000 social bots regarding the political discussion which was about one-fifth of the conversation in the 2016 U.S. elections [8].
- The second major issue is the spreading of false news. The fake news may include rumors, false information, satires or reports[9]. This may misdirect the genuine users.
- Bots may also involve in cybercrime by accessing personal and private information[10]. They may involve in brand defaming activities.

The malicious nature of the social bots gave rise to the innovation of various bot detection techniques. Several methods based on different approaches have been proposed to detect the spam accounts on social networks.

Two aspects motivated us for this research. Firstly, it is the importance of bot detection techniques. Nowadays the data collected from social media has become the basis for data analysis. Based on the output of this analysis, many organizations decide their business plans and strategies, analyze the customer reaction or evaluate their brand value. With the presence of active automated accounts, the data being analyzed is generated by both legitimate and bot accounts. Hence the output generated does not ensure genuine user opinion.

Secondly, the necessity of understanding the accuracy and efficiency of the existing bot detection techniques. The existence of several bot detection techniques also increases the necessity to understand their methodology, efficiency, and scalability. This understanding is needed to decide on a suitable technique for a set of specific features. Different aspects of the implementation such as input requirements, outcomes, algorithms, run time, robustness, scalability are to be examined to decide on the efficiency of the technique.

CHAPTER II

REVIEW OF LITERATURE

2.1 RELATED WORK

This section provides the classification and implementation details of the existing detection techniques [Fig 1]. Social bot detection techniques can be categorized into three types based on their implementations [7]. The three types of bot detection techniques are (1) Structure-Based Detection, (2) Feature-Based Detection and (3) Crowdsourcing.

2.1.1 Structure BasedBot Detection Approach

The structure of a network can be referred to as a graph, representing the relationship among the user accounts. The structure-based approach is also referred to as a graph-based approach. Based on their structure, the bot detection techniques can be implemented using three different approaches: (1) Random walk-based approach, (2) Community detection approach and (3) Markov random field-based approach.

2.1.1.1 Random WalkBased Approach

This approach is implemented by generating random paths from one node to another in the network structure. The next node in the process of path generation is chosen randomly. Based on this algorithm, seven different detection techniques have been proposed. They are SybilGuard, SybilLimit, SybilInfer, Sybil Rank, SybilResist, Criminal Account Inference Algorithm, and SybilWalk.

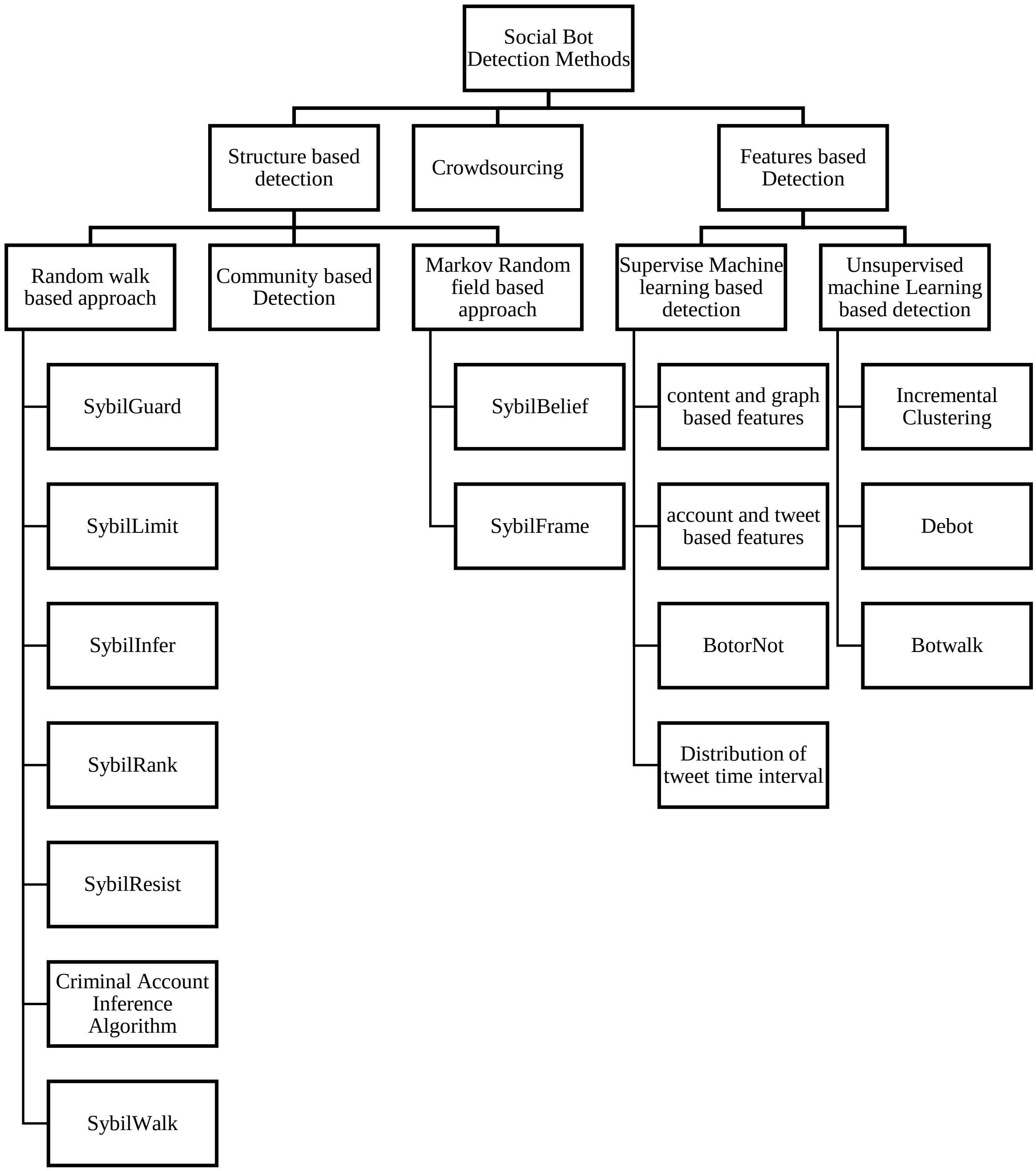

Figure 1: Classification of social bot detection techniques[7]

2.1.1.1.1 SybilGuard

According to Haifeng Yu [11], social network can be separated into two regions. First is the honest region. This region provides the relationship among the legitimate user accounts. Second is the Sybil region. The Sybil region comprises of automated accounts and their connections. A bot account can have multiple identities but can only have one edge connected to the honest region. Every node generates random paths of a fixed length w, which is equal to 2000. A node on the network is categorized as a legitimate user account if its random path intersects with the path of an honest node.

2.1.1.1.2 SybilLimit

Haifeng yu[12]proposed another random walk-based detection technique to address the two major limitations of SybilGuard. The first drawback is, it cannot restrict the Sybils if the length w is above 2000. The second is it works on the assumption of the fast mixing nature of the networks. To address these limitations, SybilLimit accepts only 10 nodes along with the path generation. This produces 200 times more productivity than SybilGuard. Though SybilLimit is better compared to SybilGuard, both the techniques show their vulnerability when honest nodes compromise [13].

2.1.1.1.3 SybilInfer

The SybilIfer technique, proposed by Danezis[14], ensures (1) the existence of a minimum of one honest node within the network (2) the awareness of the nodes about the complete network topology (3) existence of a conventional connection between the regions. This technique addresses the limitations of SybilGuard and SybilLimit, by working efficiently even when extremely high numbers of nodes behave vulnerably[13][7]. This technique enforces the Bayes Theorem to verify the likelihood of a node being Sybil.

2.1.1.1.4 SybilRank

SybilRank, proposed by Cao [15] works based on choosing random paths in the network. The network is built as an undirected graph and the nodes are ranked based on their behavior. This approach involves three stages: trust propagation, trust normalization, and ranking. Though this algorithm provides a low false positive and false negative rate, Stringhini[13]proved the assumptions made for this implementation to be false and unrealistic.

2.1.1.1.5 SybilResist

SybilResist[16]involves the implementation of multiple phases for bot detection. In the first phase, if the threshold value is less than the value of the node, it is considered as an honest node and is not considered in further phases. In the second phase, the calculation of variance and mean for the list of suspicious nodes is performed. In the last phase, the region comprising the automated accounts is detected. The vulnerability of the high threshold nodes is a major limitation of this technique. The fairness of this method also varies based on their structural changes[7].

2.1.1.1.6 Criminal Account Inference Algorithm

Criminal Account Inference algorithm [17] works on the hypothesis, stating “the bot accounts share identical phrases and links in the posts on social media”. The criminal accounts found in this algorithm can be categorized as bot accounts. This technique is enforced on the infirm network graph, by traversing along with the nodes in the network randomly from the user to its followers.

Alike SybilRank [15], the CIA [17] conjointly provides scores to each node, based on which the nature of the node is identified. A node with a higher score is outlined as a criminal account. The potency of the technique is compromised if a false identity of pre-labeled nodes is provided.

2.1.1.1.7 SybilWalk

Jinyuan Jia [18]proposed a random walk-based detection method to address the limitations of existing methods (SybilGuard[11], SybilLimit[12], SybilInfer[14], SybilRank[15], SybilResist[16], and CIA[17]). This method involves 3 stages: (1) Building a labeled social network, (2) Defining the badness score, and (3) Computing the score iteratively. The efficiency of SybilWalk[18] is higher compared to other random walk-based methods.

The adaptability of random walk-based approaches is very low. All the techniques based on this approach assume a closely linked structure. Mohasein[13][19] has proved that the assumption of these approaches, regarding the graph connectivity is not ideal. The implementation of these techniques is tedious, unreliable and requires a complete, and accurate structure of the network which is not possible.

2.1.1.2 CommunityBased Detection Approach

The random walk-based approaches assume the social network as one big community and cannot be divided further. Leskovec [13] [20] proved this to be wrong and proved the possibility of the division of network structure into communities. Vishwanath [13] [21] proved the possibility of dividing the Twitter network into two communities: Sybil and non-Sybil. Using the user graph, Tan [13] [7] proposed a community-based technique to find the Sybil’s.

2.1.1.3 Markov Random Field-Based Approach

Vishwanath [21] discovered that vulnerability to Sybil attack may increase with the implementation of community detection. Two techniques enforce this approach for bot detection. They are SybilBelief[22]and SybilFrame[23].

2.1.1.3.1 SybilBelief

This method involves three stages. Firstly, binary values 0 or 1 are assigned to each node in the graph. Secondly, a random probability is defined for each of these nodes. Thirdly, the loopy belief propagation is applied to calculate the probability of a node being Sybil or benign is calculated[22]. This approach identifies and offers ranks to the nodes within the network. Compared to the different techniques mentioned, this technique is more powerful but not scalable.

2.1.1.3.2 SybilFrame

SybilFrame[23]is a two-stage classification mechanism. Stage 1: Fetching the node related information. Stage 2: Enforcing loopy belief approach on the information fetched. Using this probability, the spam nodes are identified and ranked. 68.2% of Sybil nodes can be detected by SybilFrame[23] which is greater than the Sybils identified by SybilBelief[22][13].

2.1.2 Feature Based Detection Approach

These approaches involve the use of machine learning-based classification techniques for bot detection. Based on the type of input data providedthese techniquescan be divided into two categories: supervised machine learning-based approach and unsupervised machine learning-based approach.

2.1.2.1 Supervised Machine Learning-Based Approach

In this approach, the labeled data is provided as input to the system to detect the automated accounts. Four different techniques were proposed based on multiple features and different supervised machine learning algorithms. Based on the features, the approaches are divided into four types[13]: Content and graph-based approach, account and tweet-based approach, BotorNot[24]and distribution of tweet time interval.

2.1.2.1.1 Content and Graph-Based Approach

Wang [25, 13] proposed a supervised machine learning approach for bot detection based on content and graph-based features. Different classification methods such as support vector machines (SVM), decision tree (DT), neural network (NN) and Naive Bayesian (NB) were implemented to detect the spam accounts. Naïve Bayesian showed better results among all the four algorithms.

Stringhini[26, 13]classified the users as spammers and legitimate users using machine learning algorithms based on six features: follower ratio, similarity among messages, URL ratio, number of friends, number of tweets and friends list.

2.1.2.1.2 Account and TweetBased Approach

Chu [27, 13] observed 5 million twitter accounts to differentiate among human and automated accounts. Based on the account properties and content of the tweet, the author proposed a four-stage classification process. The four stages are Detection of periodic timing by computing condition entropy, Spam detection through Bayesian classification, using account-related features for calculating the bot deviation and implementing a random forest for decision making.

2.1.2.1.3 BotorNot

Davis [24], proposed a BotorNot technique that applies 1000 different features based on the friends, user profile, network, sentiment features, temporal and content of the tweet[13]. This system grades the likelihood of an account to be a bot i.e. its computes the percentage of an account to be a bot. This uses the Random Forest technique for detecting the bots.

2.1.2.1.4 Distribution of Tweet Time Interval

Tavares and Faisal [28]proved that the behavior of automated and legitimate accounts can be distinguished using the time gap between their tweets. They used the Twitter network and categorized the bots based on Naive Bayes classification technique. They studied the duration delay between the latest twenty tweets of the user based on which they performed the categorization.

2.1.2.2 Unsupervised Machine Learning Based Approaches

In this approach, the unlabeled data is provided as input to the system to detect the automated accounts. Three different techniques were proposed based on multiple features and different unsupervised machine learning algorithms. They are Incremental clustering[29], DeBot[30], and BotWalk[31].

2.1.2.2.1 Incremental Clustering Approach

Gao [29]modeled the tweets as a combination of the description and URL, where the description is the text of the post and the URL is the list of links specified in the text. Using this model, Gao categorized the accounts as legitimate and automated. Gao observed that the similarity among the two descriptions will increase the likelihood of the account being a bot. Using incremental clustering, the author identified the spam clusters from the list of suspicious profiles.

2.1.2.2.2 DeBot

DeBot [30] detects the automated accounts on Twitter by using warped correlation. It involves four phases for the classification of the accounts. Firstly, the indexer collects the tweets from the network. Secondly, using hashing, the users are assigned to buckets. Thirdly, the listener collects the data for suspicious profiles. Lastly, using a single linkage, the list of automated accounts is found. This technique provides higher precision in comparison to the above-mentioned approaches.

2.1.2.2.3 BotWalk

BotWalk[31]was proposed for bot detectionbased on four categories of features: (i) metadata, (ii) content, (iii) network-based and (iv) temporal. It is implemented using two techniques isolation-based and distance and angle-based. The system builds a feature matrix, performs normalization and enforces the anomaly detection techniques for classification.

2.1.3 Crowdsourcing

Over the past years, many websites have come up that perform crowdsourcing like Amazon's Mechanical Turk or MTurk. Wang proposed a crowdsourcing based bot detection mechanism, that is enforced as a two-layer method[32]. The primary is the filtration layer. In this layer, a catalog of suspicious profiles is separated from the honest accounts using any one of the previously mentioned approaches. The second is the crowdsourcing layer, during which the spam accounts are identified from the list of suspicious profiles. The people involved in the crowdsourcing layer are known as tuskers. In comparison to alternative approaches mentioned, the false positive and false negative rates are very low for this method. To enhance accuracy, inaccurate tuskers are eliminated by a voting system. The tuskers are supplied with information of users for the method of classification into legitimate and bot accounts. The privacy of the user is at risk in this approach. It might compromise on hiding private information. The implementation of this approach is costly, as we need to hire people for performing the classification[7].

2.2 PROBLEM STATEMENT

The importance of bot detection increases the necessity for efficient bot detection techniques. Numerous techniques using different algorithms have been established for bot detection. This thesis will (i) Identify the efficiency of selected techniques, compare their efficiencies and identify the efficient technique, (ii) Analyze the input requirements of each technique, (iii) Determine the outcomes of each technique, (iv) Identify the change in efficiency of techniques based on the data sets, and (v) Identify the tweet and user objects based on which the technique is performed, (vi) Find the Precision, Recall and Cohen's Kappa Coefficient for each technique.

CHAPTER III

METHODOLOGY

3.1 DATA COLLECTION

The bot detection techniques are evaluated on the Twitter social network. The related data are extracted from the network using the Twitter API. A Twitter application is created and the access keys and tokens are generated. Using these keys, the tweets are streamed into the Hadoop cluster through the Apache Flume. The streamed tweets are available in the JSON format. Three different data sets were collected, based on various keywords for one month. The sets of data streamed using Twitter API are tabulated below [Table 1].

Table 1: List of data sets collected based on specific keywords for one month

Dataset	**Keywords**	**Duration**
Trump Data Set	Donald Trump	Jan. 1 2019 to Feb. 1 2019
Immigration Data Set	Immigration, child separation, parent, illegal immigration	Feb. 1 2019 to March 1 2019
Food Data Set	diarrhea, vomiting, abdominal pain, puke	March 1 2019 to April 1 2019

3.2 DATA PREPROCESSING

This is an obligatory step before the implementation of detection techniques. A tweet can be split into two sections. One is the tweet object, that provides the details related to the tweets and the other is the user object, that provides the user-specific details.From the list of tweets collected, only the required tweet and user objects are extracted. The objects extracted from the tweet are tabulated below [Table 2] (Ref. Table 20, Rows 1 and 2).

Table 2: List of tweet and user objects extracted from the tweets

Attribute	**Object**	**Description**
Friends_count	User	The number of accounts the user is following.
Favourites_count	User	Total number of tweets liked by the user in his lifetime.
Description	User	The description given by the users about them.
Created_at	User	Date and time the account is created.
Screen_name	User	The name that uniquely identifies the user.
Id_str	User	Unique ID for each user account
Verified	User	Returns true if the account is verified, else returns false.
Statuses_count	User	Total number of tweets by the user.
Follow_request_sent	User	Number of follow request sent by the user.
Followers_count	User	The number of users following the user.
Deafault_profile_image	User	Returns true if the profile image is default otherwise false.
Retweet_count	Tweet	Count of the times the tweet is retweeted.
Retweeted	Tweet	Returns true if it is a retweet otherwise false.
Favorite_count	Tweet	The number of times it has been favorited by the users.
Text	Tweet	The content of the tweet.
favorited	Tweet	Returns true if it is favorited by the user, else returns false.
In_reply_to_screen_nam	Tweet	Gives the screen name to which the tweet is being replied.

From the screen names extracted, 10,000 unique user accounts are chosen randomly. For these accounts, the most recent twenty tweets are also collected. Using the most recent twenty tweets of the user,Wang [23] manually labelled 500 accounts into bots and legitimate accounts. The author showed that using the most recent twenty tweets of the user along with other user and tweet objects, an account can be categorized into bot and legitimate accounts [23].Using all the above data, the required features are calculated. These features are together combined into a dataset. The dataset is formatted as a CSV file and the fields of the file are listed below [Table 3] (Ref. Table 20, Row 2)

Table 3: List of features used as input for bot detection techniques

Features	**Data Type**	**Description**
Follower ratio	Integer	The ratio of number of followers to number of friends
Number of URLs	Integer	The count of http links in a single tweet
Average URLs	Integer	The average number of http links in all the twenty tweets
Text	String	List of most recent 20 tweets
Number of hashtags	Integer	The count of hashtags used in a single tweet
Average hashtags	Integer	The average number of hashtags in all the twenty tweets
location	String	The place from where the tweet was created.
Timestamps	Time	The creation time of the tweet.
Retweet count	Integer	The total number of times the tweet was retweeted.
Similarity index	Integer	The value is 1 if the tweets are similar, else 0
Number of user mentions	Integer	The count of user mentions in a tweet
Unique URLs	Integer	The count of unique URLS in all 20 tweets of the user.
Unique hashtags	Integer	The count of unique hashtags in all 20 tweets of the user.

3.3 MANUAL DETECTION

The three datasets listed in the table [Table 1] are used to evaluate the bot detection techniques. The Trump dataset is considered as Dataset I. To evaluate the performance of detection techniques being implemented, the dataset I is manually checked to find the list of bot accounts and human accounts. The Immigration dataset and Food dataset are considered as dataset II and III respectively. The bot accounts in data set II and III were listed using the BotorNot[24] application. The number of human and bot accounts identified in the three data sets are as below [Table 4].

Table 4: Number of legitimate and bot accounts detected manually for the datasets.

Dataset	Detection method	Number of legitimate accounts	Number of bot accounts
Dataset I	Manual Detection	962	9038
Dataset II	BotorNot	2567	7433
Dataset III	BotorNot	1968	8032

3.3.1 Implementation of Manual DetectionTechnique

The manual detection is performed on Dataset I. 10,000 unique user account were identified with 20 tweets considered for each user. This approach is conducted in two steps: (1) Performing the detection using content and graph-based approach to obtain suspicious profiles; (2) Using the output from step 1 as input and manually verifying the accounts.

In step 1, the 10,000 unique users are taken as input and features are extracted. Two features are considered in this approach:the number of followers and the number of friends. Naïve Bayes classification is applied to obtain the suspicious profiles.

In step 2, the suspicious profiles obtained from step 1 are taken as input and a manual verification on the identity of the accounts is performed. For the identification process, the metrics considered are:the follower and following ratio, duplicate tweets, ratio of tweet and retweet rate, no description and profile picture, numbers as username, similar tweet content, and distribution of tweet time interval.

3.3.2 Implementation of BotorNotDetection Technique

The number of the bot and legitimate accounts in Dataset II and III are detected using the BotorNot[24]application. This system assigns every user a score from 0 to 5 defining the likelihood of an account to be a bot. The accounts havinga score greater than 3.5 are considered bots and accounts scoring below 3.5 are categorized as legitimate human users.

3.4 EVALUATION METRICS

The results of the detection techniques enforced are compared against the results obtained using the manual and BototNot[24] detection. The evaluation metrics used for comparing the performance of the techniques are Precision, Recall, Accuracy, and Cohen's Kappa Coefficient.

The False Positive, False Negative, True Positive, and True negative values are calculated to find the values of evaluation metrics. The True Positive refers to the number of bot accounts detected correctly. It is determined by identifying the number of bot accounts detected correctly among the accounts classified as bots by manual detection. The True Negative refers to the number of legitimate accounts detected correctly. It is determined by identifying the number of legitimate accounts detected correctly among the accounts classified as legitimate by manual detection. The False Negative refers to the number of legitimate accounts detected incorrectly. It is determined by identifying the number of legitimate accounts termed as bot accounts by the technique. The False Positive refers to the number of bot accounts detected incorrectly (Ref. Table 20, Row 3).It is determined by identifying the number of bot accounts given by manual detection, categorized as legitimate accounts by the technique.

The Precision refers to percentage of positive results that are detected correct. It is calculated as:

$$\text{Precision} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive + False\ Positive}}$$

The Recall refers to the percentage of actual positive results that are categorized correctly. Itis calculated as (Ref. Table 20, Row 3):

$$\text{Recall} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive + False\ Negative}}$$

The Accuracy is the percentage of total accounts classified correctly. It is calculated as

$$\text{Accuracy} = \frac{\boldsymbol{True\ Positive + True\ Negative}}{\boldsymbol{True\ Positive + True\ Negative + False\ Positive + False\ Negative}}$$

The Cohens Kappa Coefficient is calculated to verify the agreement between both the results. The Cohen’s Kappa Coefficient is given by (Ref. Table 20, Row 4):

$$\mathbf{k} = \frac{\boldsymbol{Po - P_e}}{\boldsymbol{1 - P_e}}$$

where, Po $\rightarrow$ the ratio of number of results in agreement to the total results

$P_e \rightarrow$ the probability of chance of agreement

The value of k varies from 0 to 1. Based on the value, the agreement of the results is obtained. The type of agreement based on the co-efficient value can be divided into 7 categories [Table 5](Ref. Table 20, Row 3).

Table 5: The type of agreement of the results based on Cohen's Kappa Coefficient

Coefficient value	**Type of agreement**
0.10 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 0.99	Near perfect agreement
1	Perfect agreement

CHAPTER IV

FINDINGS

The implementation of the various bot detection techniques is carried out and the list of the bot and legitimate human accounts categorized by each technique are obtained. The structure-based techniques[7] are not implemented due to two reasons. First, is the requirement of a complete social network to detect the Sybil's. To enforce these techniques, a complete graph of the network is required, which is not feasible. They require complete information about all the users and their relationships. Second is their adaptability. The feature-based techniques have been proved to work more efficiently than structure-based techniques. The implementations and outcomes of the feature-based approaches are discussed in this section.

4.1 IMPLEMENTATION OF CONTENT AND GRAPH BASED APPROACH

The content-based features considered for this approach are the number of duplicate tweets, the number of HTTP links, and the number of user mentions in the most recent twenty tweets of the user. The graph-based features considered are the number of friends, number of followers, and the follower ratio. The follower ratio is the ratio of the number of followers to the sum of the number of followers and friends. If the number of links in a tweet is very high, then the likelihood of the account being a bot increases.

Using the Levenshtein distance, the similarity among the tweets is calculated to find the duplicate tweets. The Tweets with a higher number of links, increases the probability of the user being a bot. This approach is implemented using four different classification algorithms. They are Naïve Bayes, Decision Trees, SVM, and k-nearest neighbor. Out of the four, Naïve Bayes classification produced better results. The three datasets were classified to detect the bots based on the four algorithms [Table 6].

Table 6: Results obtained using Content and Graph Based Approach

Dataset	**Algorithm**	**True Positive**	**True Negative**	**False Positive**	**False Negative**
Dataset I	Naïve Bayes	862	8554	484	100
	Decision Tree	680	8598	440	282
	SVM	622	8611	427	340
	K – nearest neighbor	620	8460	578	342
Dataset II	Naïve Bayes	2276	7068	365	291
	Decision Tree	2392	7043	390	175
	SVM	2112	7113	320	455
	K – nearest neighbor	2200	6789	644	367
Dataset III	Naïve Bayes	1699	7778	254	269
	Decision Tree	1704	7587	445	264
	SVM	1747	7622	410	221
	K – nearest neighbor	1549	7589	443	419

The Precision, Recall, Accuracy, and Cohen's Kappa Coefficientare calculated based on the above results [Table 7].

Table 7:Evaluation Metrics calculated using Contentand Graph Based Approach

Dataset	**Algorithm**	**Precision**	**Recall**	**Accuracy**	**Cohen's Kappa Coefficient**
Dataset I	Naïve Bayes	64.04	89.6	94.16	0.69
	Decision Tree	60.07	70.6	92.78	0.59
	SVM	59.2	64.6	92.3	0.57
	K – nearest neighbor	51.7	64.4	90.8	0.52
Dataset II	Naïve Bayes	86.18	88.66	93.44	0.83
	Decision Tree	85.98	93.18	94.35	0.85
	SVM	86.84	82.28	92.25	0.79
	K – nearest neighbor	77.36	85.7	89.89	0.74
Dataset III	Naïve Bayes	86.99	86.33	94.77	0.84
	Decision Tree	79.29	86.59	92.91	0.78
	SVM	80.99	88.77	93.69	0.80
	K – nearest neighbor	77.76	78.71	91.8	0.74

It is observed that Naïve Bayes yields better precision values for datasets I and III compared to the other three algorithms. This shows the number of positive results is greater when the classification is done using Naïve Bayes algorithm. Naïve Bayes also yields highest Recall, Accuracy, and Cohen's Kappa Coefficient values for both Datasets I and III. In the case of Dataset II, the Decision Tree algorithm yields higher Precision, Recall, Accuracy, and Cohen's Kappa Coefficient values than the other algorithms. This shows that the Naïve Bayes algorithm yields better results for the Content and Graph Based Approach [Fig 2].

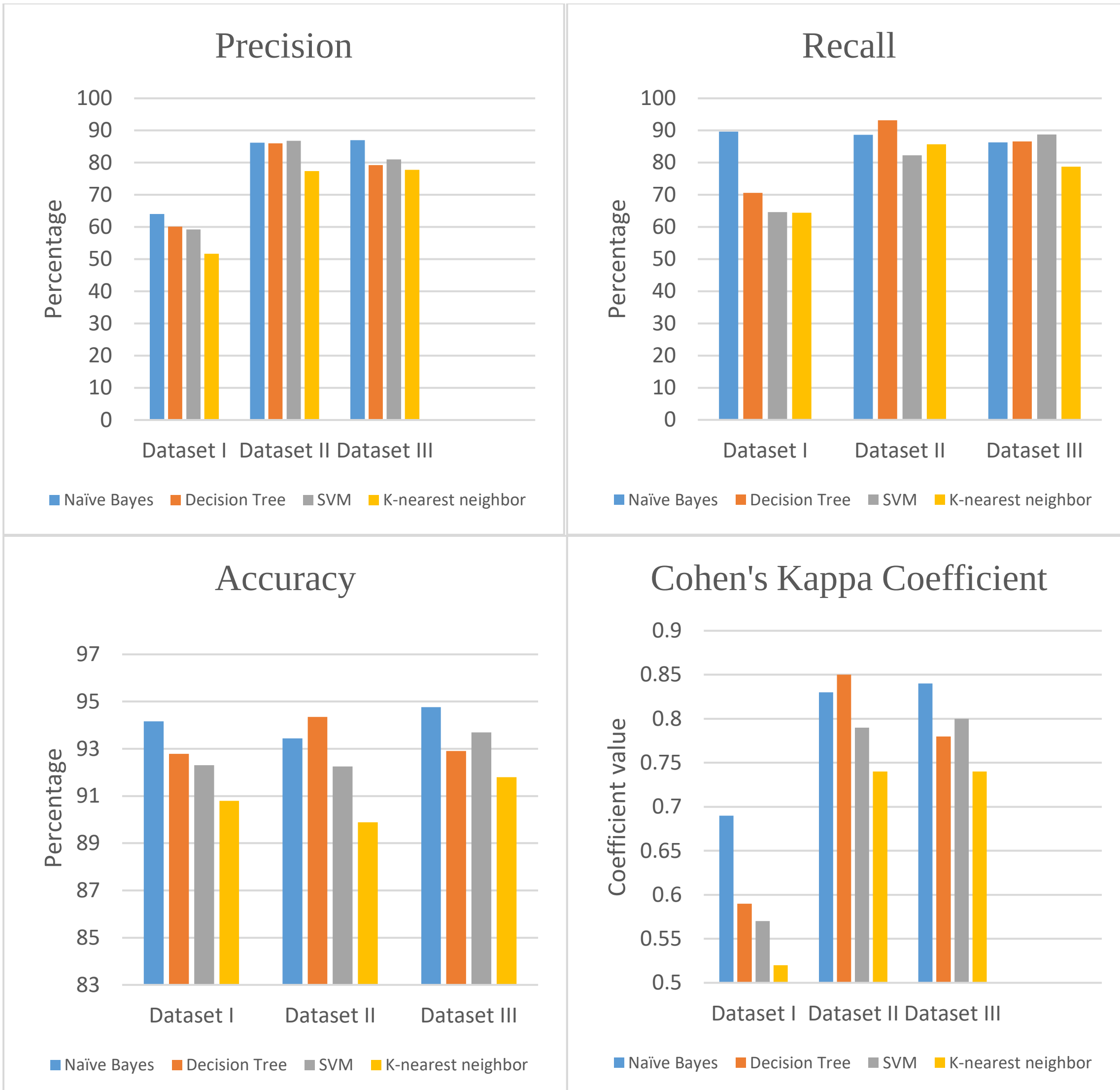

Figure 2: Evaluation Metrics of Content and Graph Based Approach

4.2 IMPLEMENTATION OF ACCOUNT AND TWEET BASED APPROACH

Three types of features are used for the implementation of this approach. First, is the interaction-driven features. They include the number of unique hashtags and the average number of hashtags. Second, is the tweets-driven features. They include the total number of hashtags and links, the average number of hashtags and links, and total and average number of user mentions. The third is URL-driven features. They include the total and average number of

URLs. This technique is implemented using three different algorithms. The classification algorithms are Random Forest, Naïve Bayes, and SVM. Based on the number of accounts classified as a bot and human, the True Positive, True Negative, False Positive, and False Negative values are obtained [Table 8]. Based on these values the Precision, Recall, Accuracy, and Cohen's Kappa coefficient are calculated [Table 9].

Table 8: Results obtained using Account and Tweet Based Approach

Dataset	**Algorithm**	**True Positive**	**True Negative**	**False Positive**	**False Negative**
Dataset I	Random forest	622	8669	369	340
	SVM	426	8901	137	536
	Naïve Bayes	593	8774	264	369
Dataset II	Random forest	1972	7229	204	595
	SVM	1949	7003	430	618
	Naïve Bayes	2221	7322	111	346
Dataset III	Random forest	1464	7689	343	504
	SVM	1133	7562	470	835
	Naïve Bayes	1522	7678	354	446

Table 9: Evaluation Metrics calculated using Account and Tweet Based Approach

Dataset	**Algorithm**	**Precision**	**Recall**	**Accuracy**	**Cohen's Kappa Coefficient**
Dataset I	Random forest	62.76	64.66	92.9	0.56
	SVM	44.28	75.67	93.27	0.52
	Naïve Bayes	61.64	69.19	93.67	0.62
Dataset II	Random forest	90.63	76.82	92.01	0.78
	SVM	81.93	75.93	89.52	0.69
	Naïve Bayes	95.24	86.52	95.43	0.87
Dataset III	Random forest	81.02	74.39	91.53	0.72
	SVM	70.68	57.57	86.95	0.55
	Naïve Bayes	81.83	77.34	92.00	0.74

The Naïve Bayes algorithm yields the highest precision, Accuracy and Cohen's Kappa Coefficient values for all the three Datasets. It shows the Naïve Bayes gives higher positive results, and accuracy in comparison to the other two classification techniques. For the Recall values, Naïve Bayes produces higher results for the Datasets II and III. However, for the Dataset I, the SVM algorithm yields greater values. This shows that the Naïve Bayes algorithm works the best and yields better results for Account and Tweet Based Approach [Figure 3].

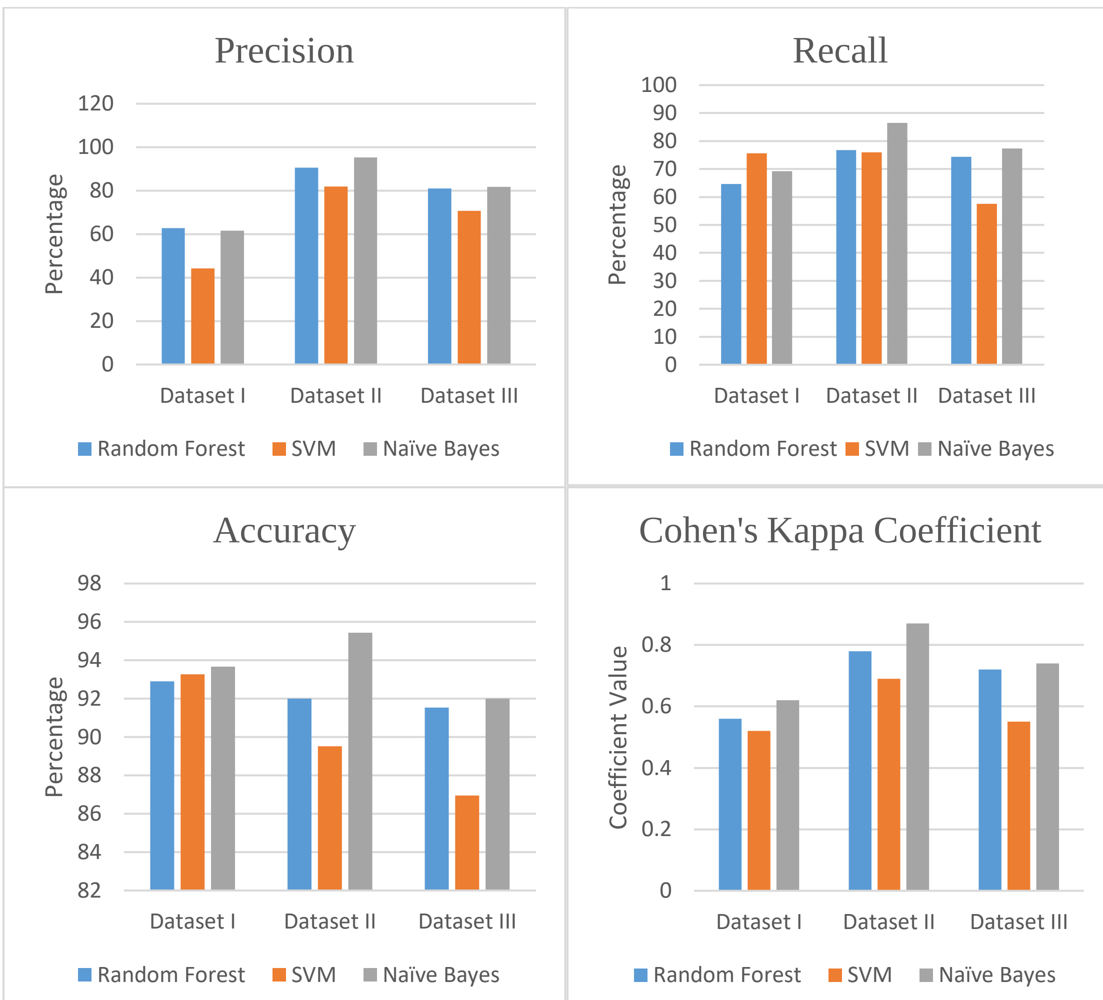

Figure 3: Evaluation Metrics of Account and Tweet Based Approach

4.3 IMPLEMENTATION OF DISTRIBUTION OF TWEET TIME INTERVAL APPROACH

This approach uses the time interval between the tweets as the feature for classifying the user accounts into humans and bots. The probability density function is computed for each account. Depending on this function, a classification score is calculated for each account. Based on the scores, if the bot class score for the account is high it is classified as a bot, if the score is low it is classified as a human account. The Naïve Bayes classification algorithm is used in this approach. Based on the number of accounts classified as a bot and human, the True Positive, True Negative, False Positive, and False Negative values are obtained [Table 10]. Based on these values the Precision, Recall, Accuracy and Cohen's Kappa coefficient are calculated [Table 11].

Table 10: Results obtained using Distribution of Tweet Time Interval Approach

Dataset	**Algorithm**	**True Positive**	**True Negative**	**False Positive**	**False Negative**
Dataset I	Naïve Bayes	466	8035	1003	496
Dataset II	Naïve Bayes	2144	6765	668	423
Dataset III	Naïve Bayes	1496	7526	506	472

Table 11: Evaluation Metrics calculated using Distribution of Tweet Time Interval Approach

Dataset	**Algorithm**	**Precision**	**Recall**	**Accuracy**	**Cohen's Kappa Coefficient**
Dataset I	Naïve Bayes	31.72	48.44	85.01	0.30
Dataset II	Naïve Bayes	76.24	83.52	89.09	0.72
Dataset III	Naïve Bayes	74.73	76.02	90.22	0.69

4.4 IMPLEMENTATION OF INCREMENTAL CLUSTERING APPROACH

The tweet text and the URLs included in these tweets are the main features considered in this approach. This approach involves the preprocessing of the URLs before performing the detection process. The URLs that are incomplete need to be reconstructed in this step of preprocessing. Based on the similarity between their texts and URLs, the tweets sharing the same URL are clustered together. The clustering process involves two steps: (1) Clustering the tweets that share the same URL (2) Merging the cluster of tweets that share similar text content.

To identify the clusters holding spam accounts, two features are used. One is the number of unique IDs of the users in the cluster, termed as distributed property. Two is the median of the tweet time interval of all the tweets in the cluster, termed as bursty property. These two features together form a pair-value property <distributed property, bursty property>. The threshold of this value is set to <5, 1.5> i.e. any cluster having a value greater than the threshold value is considered as Spam cluster.

Based on the number of accounts classified as a bot and human, the True Positive, True Negative, False Positive, and False Negative values are obtained [Table 12]. Based on these values the Precision, Recall, Accuracy and Cohen's Kappa coefficient are calculated [Table 13].

Table 12: Results obtained using Incremental Clustering

Dataset	**True Positive**	**True Negative**	**False Positive**	**False Negative**
Dataset I	529	8256	782	433
Dataset II	1872	6534	899	695
Dataset III	1133	7562	470	835

Table 13: Evaluation Metrics calculated using Incremental Clustering

Dataset	**Precision**	**Recall**	**Accuracy**	**Cohen's Kappa Coefficient**
Dataset I	40.35	54.99	87.85	0.4
Dataset II	67.56	72.93	84.06	0.59
Dataset III	70.68	57.57	86.95	0.55

4.5 IMPLEMENTATION OF DEBOT APPROACH

This approach is implemented in four stages. In the first stage, the time series is formed for the activities of the user at a time interval of T hours. In the second stage, using the hash function the users are hashed into multiple buckets. The number of buckets is set to be 2000. If the occurrence of a user is more than 5 times in a bucket, then that bucket qualifies. The number of occurrences of a user in a bucket for the bucket to be qualified is given by w divided by 4, where w is the lag time allowed. The value of w is constant, which is 20 seconds. In the third stage, the users with more than five occurrences in the qualified buckets are collected and a time series is formed again, but this time it is based on all the user activities. In the fourth stage, it uses the single linkage clustering technique to form clusters. The clusters that provide a False Positive value are considered as legitimate human accounts and the remaining clusters are considered as bot accounts.

In the single linkage clustering stage, the distance matrix is calculated. The distance matrix is calculated between the time series obtained using the user activities at time interval T. The time interval T is fixed, which is 2 hours. The minimum value in the matrix is found and the two clusters corresponding to the minimum value are merged. This process is executed iteratively until the final large cluster is formed and the bot clusters are identified.

Based on the number of accounts classified as a bot and human, the True Positive, True Negative, False Positive, and False Negative values are obtained [Table 14]. Based on these values the Precision, Recall, Accuracy and Cohen's Kappa coefficient are calculated [Table 15].

Table 14: Results obtained using DeBot Approach

Dataset	**True Positive**	**True Negative**	**False Positive**	**False Negative**
Dataset I	824	8760	278	138
Dataset II	2482	7238	195	85
Dataset III	1868	7624	408	100

Table 15: Evaluation Metrics calculated using DeBot Approach

Dataset	**Precision**	**Recall**	**Accuracy**	**Cohen's Kappa Coefficient**
Dataset I	74.77	85.65	95.84	0.76
Dataset II	92.72	96.69	97.2	0.92
Dataset III	82.07	94.92	94.92	0.85

4.6 IMPLEMENTATION OF BOTWALK APPROACH

Four different features are considered in this approach. First is the metadata-based features. They include: number of tweets in lifetime, the creation time of the account, the location of the tweet, and the privacy of the account i.e. if it is protected or verified. Second is the temporal-based features. They include: the time interval between tweets and the average number of tweets per day. Third is the content-based features. They include: the number of hashtags, number of URLs, average number of hashtags, average number of URLs, average number of tweets with hashtags, average number of tweets with URLs, retweet count of the tweet, and similarity index of tweets.

This approach is enforced using two techniques: Isolation based and Distance and angle based. In the Isolation-based technique [31], the feature matrix is split by randomly selecting a column c and split value s, where

$$\boldsymbol{Min(c) \leq s \leq Max(c)}$$

This results in the formation of k number of trees. The anomaly score is calculated for all the trees. The anomaly score is given by,

$$\boldsymbol{S(x,n) = 2 - \frac{E(h(x))}{c(n)}}$$

Where, $h(x)$is the path length of the node. $E(h(x))$ is the average of $h(x)$ and C(n) is the average length of unsuccessful search, that is given by

$$\boldsymbol{C(n) = 2h(n-1) - \frac{2(n-1)}{n}}$$

If the $S(x,n)$ value is close to 1 it is considered as a bot. It the value is close to 0 it is considered as a legitimate human account.

In the Distance and angle based technique, a normal node is found by calculating the median, which is given by

$$\boldsymbol{c = median\ (col)\ \forall\ col\ in\ F}$$

The distance between the user and the normal node c is calculated. The distance is calculated using the Euclidean distance formula. The calculate the score to classify into bots and legitimate humans the cosine distance is calculated, that is given by

$$ABD(x) = \frac{x\,.\,c}{||x||.\,||c||}$$

Based on the number of accounts classified as a bot and human, the True Positive, True Negative, False Positive, and False Negative values are obtained [Table 16]. Based on these values the Precision, Recall, Accuracy and Cohen's Kappa coefficient are calculated [Table 17].

Table 16: Results obtained using BotWalk Approach

Dataset	**Algorithm**	**True Positive**	**True Negative**	**False Positive**	**False Negative**
Dataset I	Isolation-based	762	8792	246	200
	Distance and angle based	749	8587	451	213
Dataset II	Isolation-based	2103	7287	146	464
	Distance and angle based	2319	6753	680	248
Dataset III	Isolation-based	1593	7836	196	375
	Distance and angle based	1233	7762	270	735

Table 17: Evaluation Metrics calculated usingBotWalk Approach

Dataset	**Algorithm**	**Precision**	**Recall**	**Accuracy**	**Cohen's Kappa Coefficient**
Dataset I	Isolation-based	75.6	79.2	95.5	0.75
	Distance and angle based	62.42	77.86	93.36	0.65
Dataset II	Isolation-based	93.51	81.92	93.9	0.83
	Distance and angle based	77.33	90.34	90.72	0.77
Dataset III	Isolation-based	89.04	80.95	94.29	0.81
	Distance and angle based	82.04	62.65	89.95	0.65

The Isolation-based technique performs better and yields better Precision, Accuracy and Cohen's Kappa Coefficient values for all the three Datasets. It also yields greater results of Recall for the Datasets I and III. This clearly shows that Isolation-based technique works more efficiently than the Distance and angle-based technique [Fig 4].

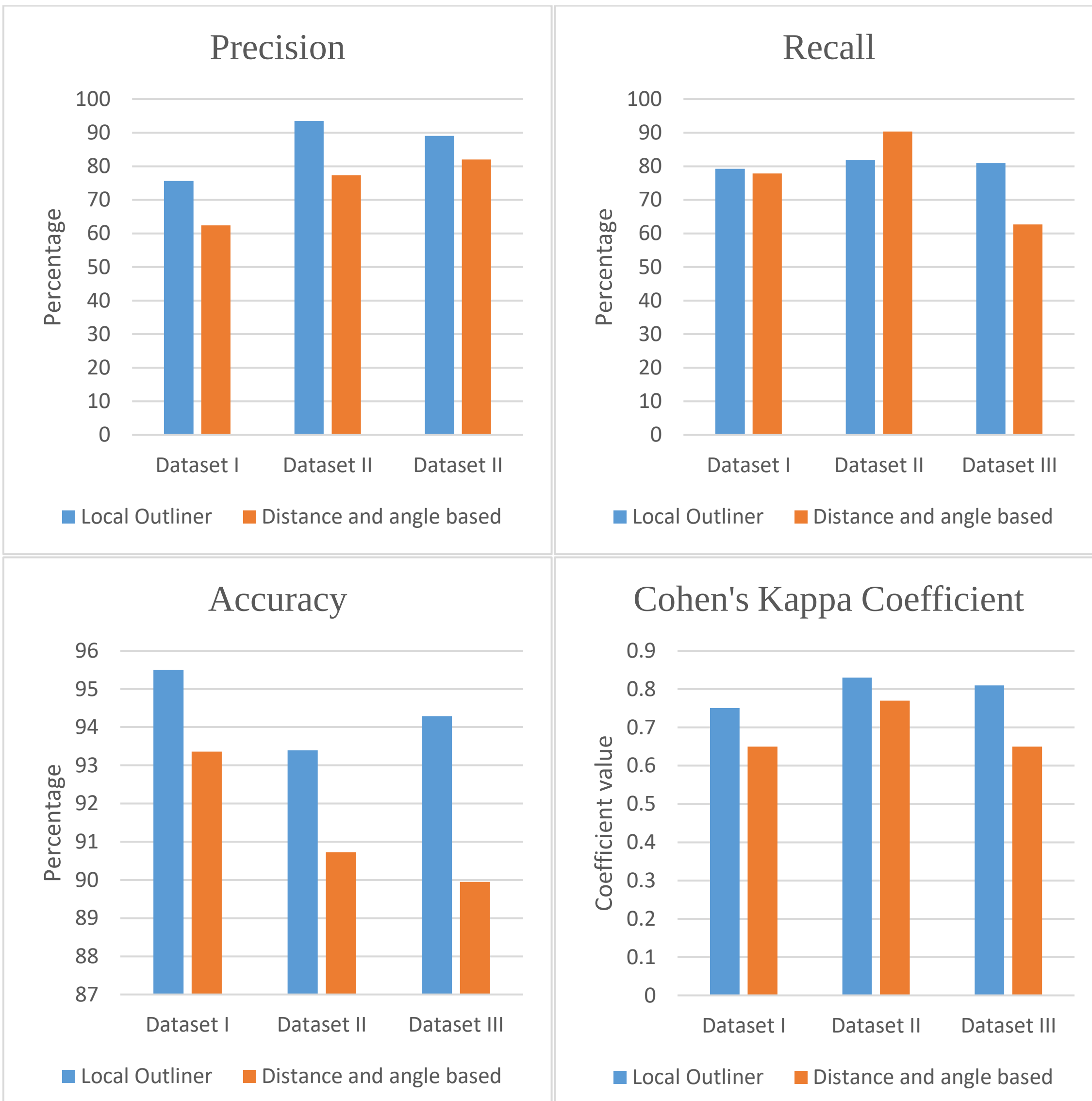

Figure 4: Evaluation Metrics of BotWalk Approach

4.7 PRECISION BASED COMPARISON OF BOT DETECTION TECHNIQUES

Precision refers to the percentage of positive results obtained by the technique. The average precision value is calculated for each technique by adding the individual precision value obtained for each Dataset. The average precision is given by,

$$\frac{Precision(Dataset\ I + Dataset\ II + Dataset\ III)}{Number\ of\ Datasets}$$

For the Content and Graph-Based approach, and Account and Tweet-Based Approach the precision values yielded by Naïve Bayes are considered, as Naïve Bayes performs better in both the approaches. The Precision values of the Isolation-based approach are considered for the BotWalk[31] approach, as it yields better results compared to the Distance and angle-based approach.

In comparison to all the other approaches, the BotWalk[31] approach yields better Precision percentage [Fig 5]. The DeBot[30] also gives similar results to BotWalk[31] approach. The distribution of Tweet Time Interval Approach and the Incremental Clustering Approach give less positive results and are not suitable for detecting the bot accounts efficiently [Fig 5].

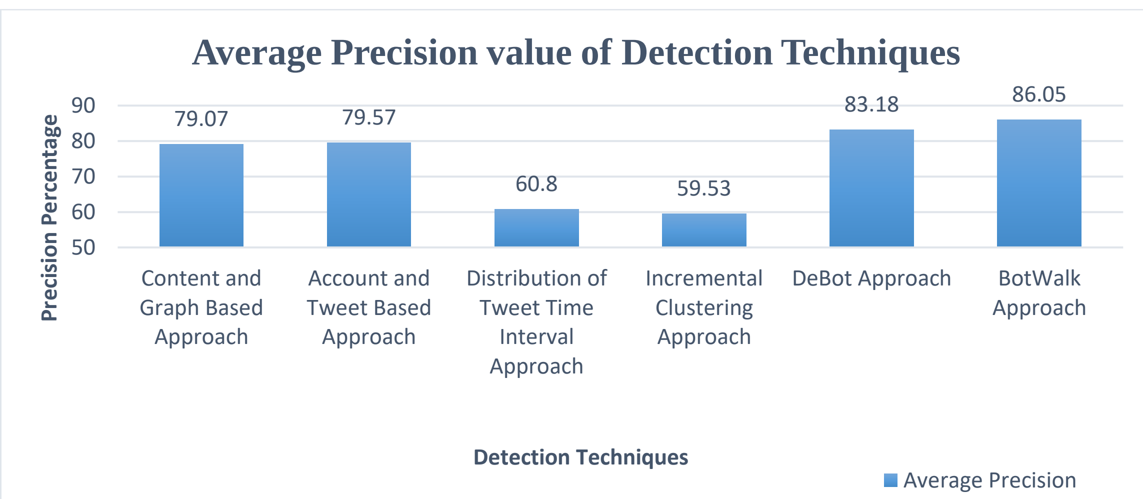

Figure 5: Average Precision values of Detection Techniques

4.8RECALL BASED COMPARISON OF BOT DETECTION TECHNIQUES

Recall refers to the percentage of correct positive results obtained by the technique. The average Recall value is calculated for each technique by adding the individual Recall value obtained for each Dataset. The average Recall is given by,

$$\frac{Recall(Dataset\ I + Dataset\ II + Dataset\ III)}{Number\ of\ Datasets}$$

For the Content and Graph-Based approach, and Account and Tweet-Based Approach, the Recall values yielded by Naïve Bayes are considered, as Naïve Bayes performs better in both the approaches. The Recall values of the Isolation-based approach are considered for the BotWalk[31] approach, as it yields better results compared to the Distance and angle-based approach.

In comparison to other bot detection techniques, DeBot[30]approach yields a very high percentage of Recall values [Fig 6]. This shows the approach is capable of detecting the highest number of correct positive results i.e. the bot accounts. The Incremental Clustering Approach gives the least number of correct positive results.

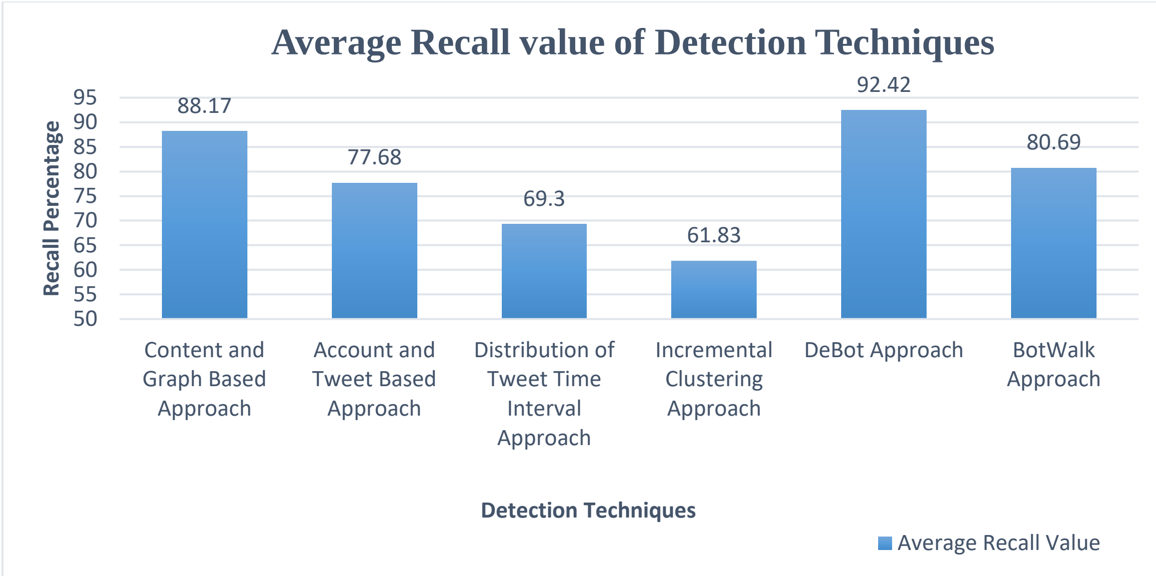

Figure 6: Average Recall values of Detection Techniques

4.9ACCURACY BASED COMPARISON OF BOT DETECTION TECHNIQUES

Accuracy refers to the percentage of total accounts classified correctly by the technique. The average Accuracy value is calculated for each technique by adding the individual Accuracy value obtained for each Dataset. The average Accuracy is given by,

$$\frac{Accuracy(Dataset\ I + Dataset\ II + Dataset\ III)}{Number\ of\ Datasets}$$

For the Content and Graph-Based approach, and Account and Tweet-Based Approach the Accuracy values yielded by Naïve Bayes are considered, as Naïve Bayes performs better in both the approaches. The Accuracyvalues of the Isolation-based approach are considered for the BotWalk[31] approach, as it yields better results compared to the Distance and angle-based approach.

In comparison to other detection techniques, DeBot[30]yields a higher accuracy. BotWalk[31]approach and Content and Graph-Based approach also give similar results to DeBot approach. The Incremental clustering approach provides the least accuracy among all the techniques [Fig 7].

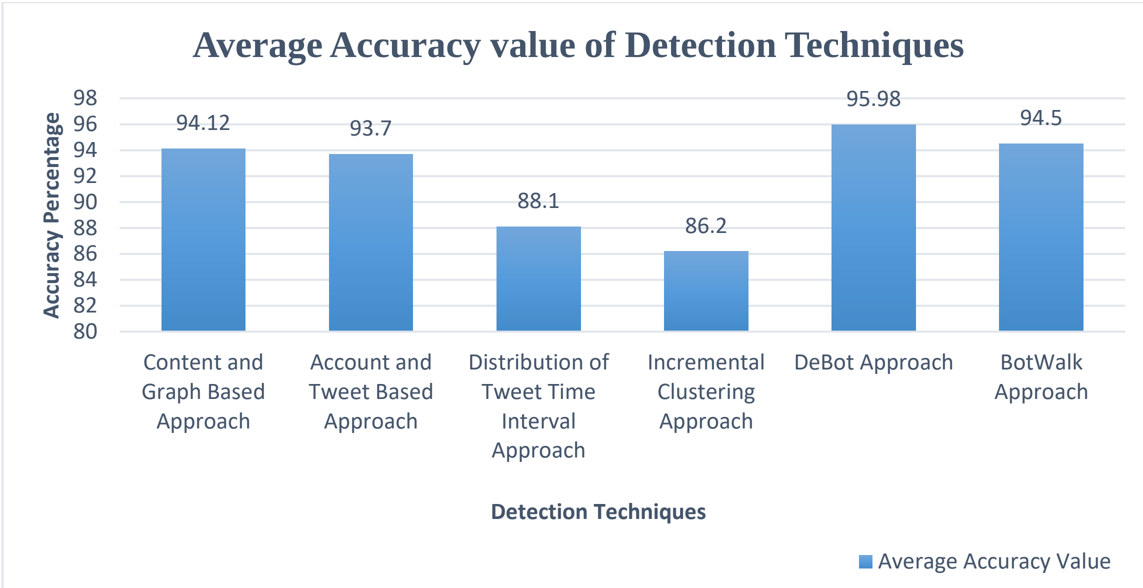

Figure 7: Average Accuracy values of Detection Techniques

4.10COHEN'S KAPPA COEFFICIENT BASED COMPARISON OF BOT DETECTION TECHNIQUES

Cohen's Kappa Coefficient provides the agreement of the detection technique results with the manual detection results. The average k value, where k is the coefficient, is calculated for each technique by adding the individual k values obtained for each Dataset. The average k value is given by,

$$\frac{K(Dataset\ I + Dataset\ II + Dataset\ III)}{Number\ of\ Datasets}$$

For the Content and Graph-Based approach, and Account and Tweet Based Approach the Coefficient values yielded by Naïve Bayes are considered, as Naïve Bayes performs better in both the approaches. The Coefficient values of the Isolation-based approach are considered for the

BotWalk[31]approach, as it yields better results compared to the Distance and angle-based approach [Figure 7].

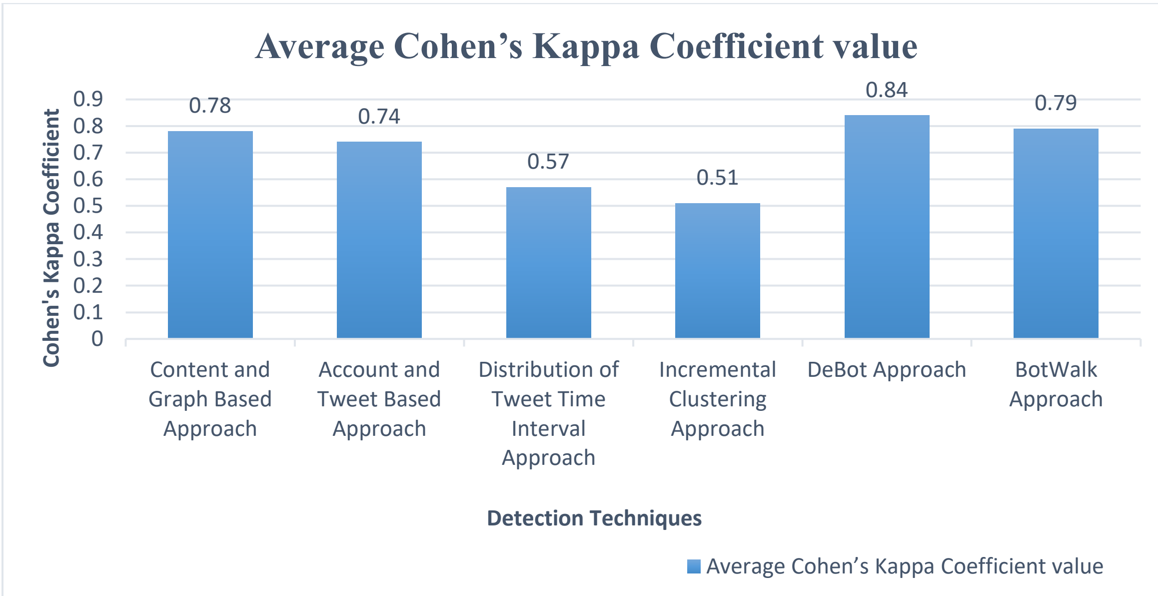

Figure 8: Average Cohen's Kappa Coefficient values of Detection Techniques

Table 18: Average Coefficient Values and the Type of Agreement of Detection Techniques

Detection Technique	**Average Coefficient Value**	**Type of Agreement**
Content and Graph Based Approach	0.78	Substantial Agreement
Account and Tweet Based Approach	0.74	Substantial Agreement
Distribution of Tweet Time Interval	0.57	Moderate Agreement
Incremental Clustering Approach	0.51	Moderate Agreement
DeBot Approach	0.84	Near Perfect
BotWalk Approach	0.79	Substantial Agreement

The DeBot[30]Approach provides the highest Coefficient value compared to the other five techniques and yields a near-perfect agreement of the results (Reference Row 6, Table 18). The Content and Graph-Based Approach, Account and Tweet Based Approach, and BotWalk[31]approach yield a substantial agreement of results compared to other techniques (Reference Row 2, 3, 7, Table 18). The Incremental Clustering Approach Distribution of Tweet Time Interval Approach and Distribution of Tweet Time Interval Approach provide the least Coefficient values compared to other techniques respectively by yielding a moderate agreement (Reference Row 5,6, Table 18).

4.11INFERENCE

In this research, the input requirements, approach, outcomes, accuracy, and the efficiency of the social bot detection techniques are analyzed. The input requirements are the features that are used for performing the detection. The outcomes are the type of output the system provides. The precision, recall, and accuracy are used to measure the efficiency of the approaches. The details of the input requirements, approach, and findings of the approaches are listed below [Table 19]. The accuracy is identified by calculating the percentage of total accounts classified correctly by the technique.

Table 19: The Input requirements, Approach, Output and Findings identified for Detection Techniques

Detection Technique	**Input Requirements**	**Approach**	**Output**	**Findings**
Content and Graph Based Approach	Number of followers, number of friends, Tweet text	Calculation of similarity of text and Follower ratio; classification (Naïve Bayes, Decision Tree, SVM, K- nearest neighbor)	List of bot and legitimate human accounts	Naïve Bayes classification works better. 94.12% accuracy
Account and Tweet Based Approach	Text of the Tweet	Identify the number of hashtags, URLs and user mentions; Classification(Random Forest, SVM, Naïve Bayes)	List of bot and legitimate human accounts	Naïve Bayes classification works better. 93.7% accuracy
Distribution of Tweet Time Interval Approach	Text of the Tweet and timestamps	Find the probability Density Function; Calculate the classification score	Scores for each account	Naïve Bayes Classification. 88.1% accuracy
Incremental Clustering Approach	Text of the Tweet	Calculate the similarity in URLs and form clusters, merge clusters with similar tweet text	Spam clusters	Clustering. 86.2% accuracy.
DeBot Approach	Time series of user activities	Hashing into buckets; Single Linage Clustering	Bot Clusters	Single Linkage clustering 95.98% accuracy
BotWalk Approach	Text of the Tweet, retweet count, privacy status, number of friends and followers, location, number of total Tweets , age of the user account	Build a feature matrix, Apply Isolation-based or Distance and Angle based techniques	Classification scores to accounts	Isolation-based yields better results. 94.5% accuracy.

The efficiency of the techniques is decided based on their precision, recall, and accuracy values. The DeBot[30] is identified as the most efficient technique with the highest Recall, Accuracy and Cohen's Kappa Coefficient values, followed by BotWalk[31], Content and Graph-Based Approach, Account and Tweet Based Features, Distribution of Twee Time Interval, and Incremental Clustering Approach respectively [Fig 9].

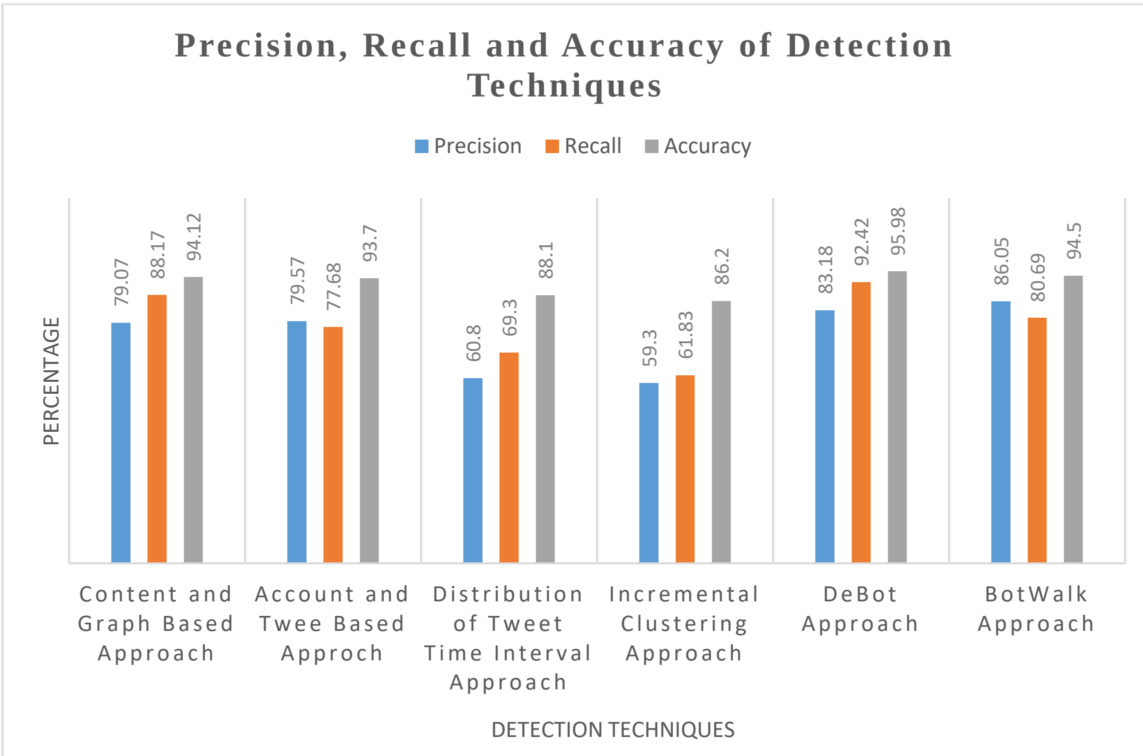

Figure 9: Evaluation Metrics of bot detection techniques

CHAPTER V

CONCLUSION

In this research, we compare the working of various social bot detection techniques. The techniques are identified and implemented. The Structure-based Techniques are not implemented in this research, as they require the complete details of the Network, which is not feasible. The implementation of the Feature-Based Bot Detection Approaches has provided an understanding of the input requirements, approach, outcomes, and efficiency comparison of the techniques. The count of the bot and legitimate human accounts aredetected using these techniques. The implementation is based on Twitter Social Network. Based on the results, the True Positive, True Negative, False Positive, and False Negative values are obtained. These values are used to calculate the evaluation metrics to compare the efficiency of the techniques.

For the Supervised Machine Learning-Based Approaches, the Naïve Bayes Classification technique yields a higher Recall, Accuracy and Cohen's Kappa Coefficient value. In the Supervised Approaches, the efficiency of Content and Graph-Based Approach is greater compared to the other two techniques. For the Unsupervised Machine Learning Based Approaches, the DeBot[30]Approach shows higher efficiency. The decreasing order of efficiency of the techniques isDeBot[30], BotWalk[31], Content and Graph-Based Approach, Account and Tweet Based Approach, Distribution of Tweet Time Interval Approach, and Incremental Clustering.

The accuracy of techniques varies based on the datasets. But the efficiency order of the technique remains the same for all the datasets. The variation of the evaluation metrics and efficiency based on the type of datasets is proposed for future work.

Table 20: Links

Serial Number	**Links**
1	Tweet Object of Tweet https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object
2	User Object of Tweet https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object
3	Precision, Recall, Accuracy Formulae https://towardsdatascience.com/precision-vs-recall-386cf9f89488
4	Cohen's Kappa Coefficient https://www.statisticshowto.datasciencecentral.com/cohens-kappa-statistic/

REFERENCES

[1] P. R. Center, "Social Media Fact Sheet," 2019.

[2] Q. 2. L. t. shareholders, 2019.

[3] O. Varol, E. Ferrara, C. A. Davis, F. Menczer and A. Flammini, "Online Human-Bot Interactions: Detection, Estimation, and Characterization," *arXiv:1703.03107v2,* 27 march 2017.

[4] Y. Roth and D. Harvey, " How Twitter is fighting spam and malicious automation," 2018.

[5] C. Snapshots, "How powerful are Social Bots? Understanding the types, purposes and impacts of bots in social media," 2018.

[6] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini and F. Menczer, "Detecting and Tracking Political Abuse in Social Media," *ICWSM,* vol. 11, pp. 297-304, 2011.

[7] A. Karataş and S. Şahin, "A Review on Social Bot Detection Techniques," in *ISCTurkey 10th International Information Security and Cryptology Conference,* Oct 2017.

[8] A. Bessi and E. Ferrara, "Social bots distort the 2016 u.s. presidential election online".

[9] Alessandro, C. Shao, G. L. Ciampaglia, O. Varol, A. Flammini and F. Menczer, "“The spread of fake news by social bots”,," *arXiv,* Vols. 1707.07592,, 2017.

[10] arxivblog, "“First Evidence That Social Bots Play a Major Role in Spreading Fake News”," Emerging Technology, 2017.

[11] Y. H, M. Kaminsky, P. B. Gibbons and A. Flaxman, "Sybilguard: defending against Sybil attacks via social networks," *ACM SIGCOMM Computer Communication Review,* vol. 36, no. ACM 2006, pp. 267 - 278, 2006.

[12] H. Yu, P. B. Gibbons, M. Kaminsky and F. Xiao, "SybilLimit: a near-optimal social network defense against sybil attacks," *IEEE/ACM Transactions on Networking,* vol. 18, no. 3, June, pp. 885-898, 2010.

[13] M. Latah, "The Art of Social Bots: A Review and a Refined Taxonomy," *ArXiv,* vol. abs/1905.03240, 2019.

[14] G. &. M. P. Danezis, "SybilInfer: Detecting Sybil Nodes using Social Networks..," 2009.

[15] Q. Cao, M. Sirivianos, X. Yang and a. T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *9th USENIX conference on Networked Systems Design and Implementation*, 2012.

[16] Ma, W, S. Z. Hu, Q. Dai, T. Wang and Y. F. Huang, "Sybil-resist: A new protocol for Sybil attack defense in social network," *International Conference on Applications and Techniques in Information Security,,* pp. 219-230, 2014.

[17] C. Yang, R. Harkreader, J. Zhang, SeungwonShin and G. Gu, "Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter," *WWW '12 Proceedings of the 21st international conference on World Wide Web,* pp. 71-80, 2012.

[18] J. Jia, B. Wang, N. Z. Gong and quot, "Random walk based fake account detection in online social Networks," in *IEEE*, 2017.

[19] A. Mohaisen, A. Yun and Y. Kim, "Measuring the mixing time of social graphs," *IMC '10 Proceedings of the 10th ACM SIGCOMM conference on Internet measurement,* pp. 383-389, 2010.

[20] Leskovec, J. K. J. Lang, A. Dasgupta and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," *17th international conference on World Wide Web,* no. ACM, pp. 695-704, 2008.

[21] B. Viswanath, A. Post, K. P. Gummadi and A. Mislove, "An analysis of social network-based Sybil defenses.," in *the ACM SIGCOMM 2010 conference (SIGCOMM '10)*, NY, USA, 2010.

[22] G. Neil, Zhenqiang, F. Mario and P. Mittal, "SybilBelief: A Semi-Supervised Learning Approach for Structure-Based Sybil Detection," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 1556-6021, p. 976–987, 2014.

[23] P. Gao, N. Z. Gong, S. Kulkarni, K. Thomas and P. Mittal, "SybilFrame: A Defense-in-Depth Framework for Structure-Based Sybil Detection," *ArXiv,* 2015.

[24] Davis, C. A., O. Varol, E. Ferrara, A. Flammini and Menczer, "Botornot: A system to evaluate social bots.," *25th International Conference Companion on World Wide Web, Pp. 273–2,* no. International World Wide Web Conferences Steering Committee, p. 273–274, 2016.

[25] A. H. Wang, "Detecting Spam Bots in Online Social Networking Sites: A Machine Learning Approach," *Data and Applications Security and Privacy XXIV*, pp. 335-342, 2010.

[26] Stringhini, G. C. V. Kruegel and G., "Detecting spammers on social networks," in *the 26th Annual Computer Security Applications Conference*, New York, USA, 2010.

[27] C. Z., S. Gianvecchio, H. Wang and S. Jajodia, "Detecting automation of twitter accounts: Are you a human, bot, or cyborg?," in *IEEE Transactions on Dependable and Secure Computing*, 2012.

[28] T. G and A. Faisal, "Scaling-laws of human broadcast communication enable distinction between human, corporate and robot twitter users," 2013.

[29] Gao, H. J. Hu, C. Wilson, Z. Li, Y. Chen and B. Y. Z. 2, "Detecting and characterizing social spam campaigns.," *10th ACM SIGCOMM conference on Internet measurement*, no. ACM, pp. 35-47, 2010.

[30] C. N, H. H and M. A, "DeBot: twitter bot detection via warped correlation," *IEEE 16th International Conference on Data Mining (ICDM)*, pp. 817-822, 2016.

[31] Minnich, N. C. A., D. Koutra and A. Mueen, "Botwalk: Efficient adaptive exploration of twitter bot networks.," *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 467-474, 2017.

[32] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger and H. Zheng, "Social turing tests: Crowdsourcing Sybil detection," *ArXiv*, vol. arXiv: 1205.3856, 2012.

APPENDICES

1. **Extraction of user details, recent twenty tweets with time stamp**

```
def collect_tweets(name):
accountinformation = AccessRequest.get_user(name)
    friends = accountinformation.friends_count
    followers = accountinformation.followers_count
DataList = [name, friends, followers]
while count < 1:
        count = count + 1
        tweets = AccessRequest.user_timeline(screen_name=name, count=20)
ListofTweets.extend(tweets)
with open('TestingData.csv', 'a', newline='') as g:
        add = csv.writer(g)
add.writerows([DataList])
pass
    with open('%s.csv' % name, 'w', newline='') as f:
        writer = csv.writer(f)
writer.writerow(["User_id", "creation_time", "tweet_text"])
writer.writerows([tweet.id_str, tweet.created_at, tweet.text.encode("utf-8")] for
tweet in ListofTweets)
pass
if __name__ == '__main__':
CKey = "0FNx1TAeeJrQOMOLaRflZsI6o"
CSecret = "KsXwgnmNhpZipxr50s8dZHF8gYfHfY2XRviGGCHDR7QlPOl67t"
```

```
ATSecret = "Dwf6JimKAbrtTrqm1DvX5iULG70GnkRHjQNKZnxQLzGZR"

AToken = "706139316997599233-mJqfE4xEDMBepXsokmYRB4ebIp65sA0"

AccessRequest = tweepy.OAuthHandler(CKey, CSecret)

AccessRequest.set_access_token(AToken, ATSecret)

AccessRequest = tweepy.API(AccessRequest)

with open('TestingData.csv', 'a') as g:

        add = csv.writer(g)

add.writerow(["name","followers","friends"])

pass

    with open('tester.csv') as csvfile:

readCSV = csv.reader(csvfile, delimiter=',')

for row in readCSV:

collect_tweets(row[0])
```

2. User object of Tweet

"user": {

"utc_offset": null,

"friends_count": 420,

"profile_image_url_https":

"listed_count": 0,

"profile_background_image_url": "",

"default_profile_image": false,

"favourites_count": 52538,

"description": "Retired teacher, proud American, wife, and

animal owner. I am very supportive of

President Donald Trump. We need to put the
American people first.",
"created_at": "Mon Jul 02 19:58:00 +0000 2018",
"is_translator": false,
"profile_background_image_url_https": "",
"protected": false,
"screen_name": "SandraC42595084",
"id_str": "1013874449840345088",
"profile_link_color": "1DA1F2",
"translator_type": "none",
"id": 1013874449840345088,
"geo_enabled": true,
"profile_background_color": "F5F8FA",
"lang": "en",
"profile_sidebar_border_color": "C0DEED",
"profile_text_color": "333333",
"verified": false,
"profile_image_url":
"time_zone": null,
"url": null,
"contributors_enabled": false,
"profile_background_tile": false,
"profile_banner_url": ,
"statuses_count": 47791,
"follow_request_sent": null,
"followers_count": 395,

"profile_use_background_image": true,

"default_profile": true,

"following": null,

"name": "Sandra Cooper",

"location": null,

"profile_sidebar_fill_color": "DDEEF6",

"notifications": null

}

3. **Tweet Object of the Tweet**

{

"retweet_count": 0,

"retweeted": false,

"geo": null,

"filter_level": "low",

"in_reply_to_screen_name": null,

"is_quote_status": false,

"id_str": "1079980539036094464",

"in_reply_to_user_id": null,

"favorite_count": 0,

"id": 1079980539036094464,

"text": "RT @charliekirk11: For my first tweet of 2019 I just want to remind all the liberals Donald Trump is still your President and Brett Kavanau…",

"place": null,

"lang": "en",

"quote_count": 0,

"favorited": false,

"coordinates": null,

"truncated": false,

"timestamp_ms": "1546322400231",

"reply_count": 0,

"entities": {

"urls": [],

"hashtags": [],

"user_mentions": [

{

"indices": [3,17]

"screen_name": "charliekirk11",

"id_str": "292929271",

"name": "Charlie Kirk",

"id": 292929271

}

],

"symbols": []

}

4. Similarity Index Calculation

```
list= pd.read_csv('realDonaldTrump.csv')

list.head(5)

list1=[]

list1=df['tweet_text']

list['tweet_text'].count()

final=[]
```

```
count=0
for each in list1:
if each!=each:
    count+=1
final.append(count)
final
list['Repeated_count'] = final
list.head()
list.to_csv (r'realDonaldTrump_solution.csv', index = None, header=True)
```

5. **Implementation ofApproaches**

#Content and Graph Based Approach

```
def FeatureExtraction(name, followers, friends):
FollowingRatio = followers/friends
check = 0
df = pd.read_csv('Datafile.csv')
tweet_list = []
tweet_list.append(df['tweet_text'])
similarity: int = 0
count = 0
for each in list1:
if each != each:
      count += 1
  similarity = similarity + count
  similarity
  value = '@'
```

```
MentionCount = 0
for each in tweet_list:
for iin each:
for j in i:
if j == value:
                    print(j)
MentionCount += 1
    value1 = '#'
HashTagCount = 0
for each1 in tweet_list:
for a in each1:
for b in a:
if b == value1:
HashTagCount += 1
DataList = [name, FollowingRatio, HashTagCount,
MentionCount, similarity]
with open('%TrainingData.csv', 'a') as f:
        writer = csv.writer(f)
while check == 0:
writer.writerow(["name", "Follower Ratio", "Number of HashTags",
"Number of UserMentions", "Similarity"])
            check = check + 1
writer.writerows([DataList])
pass
if __name__ == '__main__':
with open('PK.csv') as csvfile:
```

```
readCSV = csv.reader(csvfile, delimiter=',')

for row in readCSV:

FeatureExtraction(row[0], row[1], row[2])
```

#Account and Tweet Based Approach

```
def FeatureExtraction(name):
    df = pd.read_csv('%s.csv' % name)
tweet_list = [df['tweet_text']]
mentionlist = []
hashlist = []
    value = '@'
    for each in tweet_list:
for iin each:
for j in i:
if j == value:
                    Count += 1
mentionlist.append(j)
    mentions = Count
for iin mentionlist:
for kin each:
if i == k:
inc += 1
average_number_of_mention = total_mention/tweet_count
total_number_of_mentions = mentions
unique_number_of_mentions = inc
    initial = "hhtp"
```

```
    for each1 in tweet_list:
for a in each1:
for b in a:
if b == value1:
HashTagCount += 1
hashlist.append(j)
average_number_of_links = total_mention / tweet_count
total_number_of_links = mentions
unique_number_of_links = hashlist.unique()
    value1 = '#'
HashTagCount = 0
for each1 in tweet_list:
for a in each1:
for b in a:
if b == value1:
HashTagCount += 1
hashlist.append(j)
    hashtags = HashTagcount
for iin hashlist:
for kin each:
if i == k:
inc += 1
average_number_of_hashtags = total_mention / tweet_count
total_number_of_hashtags = mentions
unique_number_of_hashtags = inc
DataList = [name, average_number_of_mention, total_number_of_mentions,
```

```
unique_number_of_mentions, average_number_of_hashtags,
total_number_of_hashtags, unique_number_of_hashtags,
average_number_of_links,
total_number_of_links, unique_number_of_links,
average_number_of_hashtags]
with open('%TrainingData.csv', 'a') as f:
        writer = csv.writer(f)
writer.writerows([DataList])
pass
```

#Distribution of Tweet Time Interval Approach

```
import datetime
def timeintervalcalculation(name):
    df = pd.read_csv('%s.csv' % name)
datelist = []
datelist = df['creation_time']
for each in datelist:
for iin each:
            timestamp = datetime(i) \
                   - datetime(i + 1)
            list = []
list.append(timestamp.seconds)
with open('%TrainingData.csv', 'a') as f:
        writer = csv.writer(f)
while check == 0:
writer.writerow(["name","minimum time" , " maximum time" ,
```

```
" total timegap" ])
        check = check + 1
writer.writerows(name, min(timestamp), max(timestamp), sum(timestamp))
pass
if __name__ == '__main__':
with open('screennames.csv') as csvfile:
readCSV = csv.reader(csvfile, delimiter=',')
for row in readCSV:
timeintervalcalculation(row[1])
```

#Incremental Clustering

```
import pandas as pd
import urllib2
def processing(data_set):
tweet_text_list = data_set['tweet_text']
data_set = []
for each in tweet:
    var = "@"
    for each1 in tweet_list:
for a in each1:
if a == var:
url = a.compile(r'@([^\s:]+)')
expanded_url = url.geturl()
return expanded_url
pass
```

```
if __name__ == '__main__':
    data = pd.read_csv('%s.csv' % name)
    processing(data)
with open('Data.csv', 'a') as f:
        writer = csv.writer(f)
while check == 0:
            check = check + 1
writer.writerows(name, expanded_url, text)
```

#BotWalk Approach

```
import pandas as pd
df = pd.read_csv('%Features.csv' % name)
tweet_list = [df['tweet_text']]
similarity: int = 0
count = 0
for each in list1:
if each != each:
        count += 1
similarity = similarity + count
similarity
list=[df['privacy' , ' location', 'statuses_count', 'creation_time']]
hashlist = []
initial = "hhtp"
for each1 in tweet_list:
for a in each1:
for b in a:
```

```
if b == value1:
HashTagCount += 1
hashlist.append(j)
average_number_of_links = total_mention / tweet_count
total_number_of_links = mentions
value1 = '#'
HashTagCount = 0
for each1 in tweet_list:
for a in each1:
for b in a:
if b == value1:
HashTagCount += 1
hashlist.append(j)
hashtags = HashTagcount
for iin hashlist:
for kin each:
if i == k:
inc += 1
average_number_of_hashtags = total_mention / tweet_count
total_number_of_hashtags = mentions
DataList = [name,
average_number_of_hashtags, total_number_of_hashtags,
average_number_of_links,total_number_of_links]
with open('%Data.csv', 'a') as f:
    writer = csv.writer(f)
writer.writerows([DataList][list])
```

VITA

BHAGYASRI VALLABHANENI

COMPUTER SCIENCE

Master of Science

Thesis: A COMPARISON OF SOCIAL BOT DETECTION TECHNIQUES

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in December 2019.

Completed the requirements for the Bachelor of Technology in Computer Science at GITAM University, Hyderabad, India in 2017.