TWITTER FAKE TRENDS AND POPULARITY DETECTION

A Dissertation
Presented to
The Academic Faculty

by

Ahmer Latif Khan

In Partial Fulfillment
of the Requirements for the Degree
Master Of Science in the
Syed Babar Ali School Of Science & Engineering

Thesis Advisor
Dr. Fareed Zaffar

Lahore University of Management & Sciences
July 2018

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Ahmer Latif Khan

July 2018

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to Dr. Fareed Zaffar for supervising my master's thesis. I would like to thank him for encouraging my research and for providing the guidelines for the research work. His mentorship and guidance helped me in all the time of research.

I would like to pay special thanks to all the fellows and friends who have been with me throughout the duration of my degree for the friendly atmosphere and support. A special thanks to all my family members. Words cannot express how grateful I am to my mom and dad for all the sacrifices that they made for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Since the rise of social networks like Twitter the dynamics of information spread around the world has completely changed. Not only just the spread of information but many phenomena attached to social ranking and influence of individual's have all been affected and have become more dependent on social media these days. Whenever an up to date information is required most of us turn to the web to get an insight into people's views and expression about an incident, people prefer social networks like Facebook and Twitter over any other. Due to this immense trust on famous personalities on Twitter and the opinions shared people use illegal means to increase the popularity of their views known as Tweet on Twitter.

It's not only the posts some even go to the length of manipulating the count of their followers and friends which can be considered as one of the factors of popularity over others on Twitter. We found out that the most popular twitter users in Pakistan had more than 90% of fake followers. Furthermore, upon analyzing the retweeters of the most trending political topics on twitter in Pakistan in 2017 through a machine learning algorithm, we found that most of the trends had about 30% bots retweeting the trends to make it popular.

CHAPTER 1. INTRODUCTION

As we are progressing to a digital age the concept of popularity and a person's credibility are very dependent on his acceptance on various social platforms. Twitter is one of the famous social platforms where public figures try and express their views on various subjects. It is a general belief that these views are the person's own view and are unfiltered without any agenda behind. From elections to daily activities public figures express themselves on twitter which the world believes and quotes on various occasions may it be a debate or a news article.

Since the mases put a lot of trust in Twitter and the standards for popularity and credibility are so amalgamated with twitter demographics a lot of fake and malicious activity has penetrated the social network and is trying to manipulate the general public. With this study we would like to analyze the penetration of these malicious entities specifically in politics in Pakistan for the year 2017. Our work is focused around detecting and catching such fake activity and come up with numbers to have a deep understanding of the level of penetration these entities have in shaping the opinion of the public and how alarming the situation is at the present.

The rest of the document would be as follow. First twitter and its demographics would be discussed to build an understanding as to how the social platform generally works and what are the measures of popularity. Next the process for data collection be explained. Third data analysis would be discussed and finally the results would be displayed.

1.1 Twitter

Twitter is a leading social network platform of the modern world. It has millions of users using its platform to connect and share with each other. Many public figures tend to frequently use Twitter to express their opinions and have a large fan base on the network that follow them and wait for them to express their views on the network. Twitter has a lot of functions to make it easier for people to connect and share. Following a user on twitter is the most basic use of twitter. The number of followers are the basic measure of popularity on the network as well.

1.1.1 Tweet

The most basic and important thing a user can do after joining the network is to express his views. These views on twitter when expressed are called tweets. Tweets are normally in the form of pictures and texts. Tweets are not only restricted to these two forms, twitter accepts a variety of formats for a user to express his current feelings.

Figure 1: General Tweet

As you can see in figure 1 a tweet tells which user made this tweet and the time it was made. In the body of the tweet is what the user wanted to express in this case through text. The top shows the name and the user name of the user making the tweet plus his profile picture.

Considering public figures on Twitter tweets are the most important object that the society keeps their eye out for. Tweets show what their icon or idol thinks about various subjects and happenings in the society and these are the ones being frequently quoted on news channels and articles these days specially in regard to politics.

1.1.2 Retweet

The next most important measure or function on Twitter is a Retweet. As is clear by its name Retweet is the concept of resharing a tweet made by a user. There are several meanings behind resharing a tweet but the one considered the most important due to which

a retweet has become a significant measure for popularity of a topic is when someone reshares a person's tweet to show his agreement to the view and supports it to increase the reach of this view to various audience.

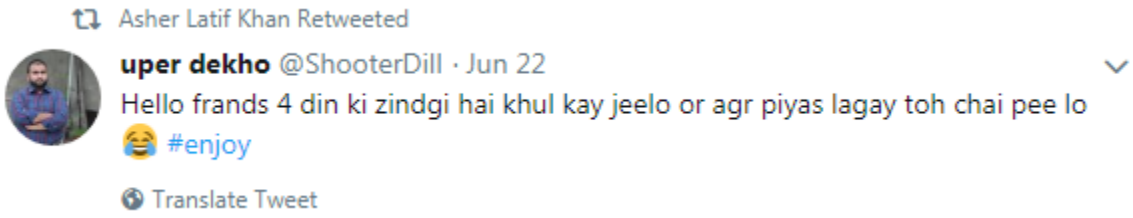

Figure 2: Retweet

As you can see in Figure 2 retweets are marked on the top making it clear that they are retweets. Furthermore, when you scrape these tweets there is an RT tag attach to the object indicating it to be a retweet.

Figure 3- Retweet count

The number of retweets a certain tweet has are also visible in the bottom of the tweet next to the recycle kind of sign. The higher the number of retweets the greater chance of it becoming a trend as explained next.

1.1.3 Trend

While sharing tweets people tend to mark the key word with an "#" to specify the genre or the intent of the tweet. When the same key word keeps being tweeted and retweeted for a certain minimum, it becomes a trend. Trend is an important measure of twitter studies as it gave a measure to the interest of masses in specific topics or genre.

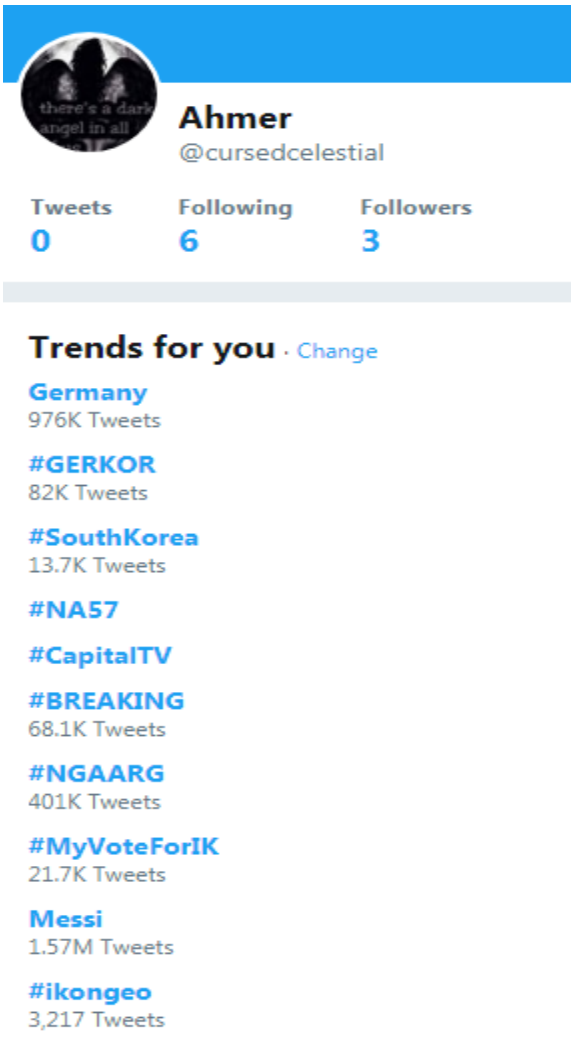

Figure 4 – Trends

Trends appear on the left side of your home screen. Trends showed to you are different based on the geo location your account is set to and your interests as seen in Figure 4.

1.1.4 Followers

One of the most important popularity measure for a user on twitter is the number of followers he has. Followers are people that choose to follow you in a manner that they receive notifications for your new tweets or can see them on their home page. More the followers a user have more he is believed to be popular because anything shared by him would be viewed by masses and has a high probability of being reshared further.

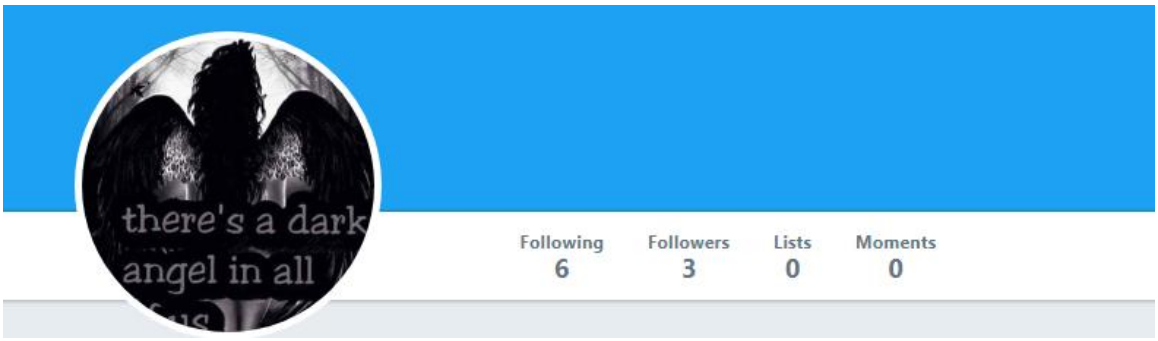

Figure 5- Profile

The number of followers a user has are visible on his profile if the profile is public. They are shown right at the top as can be seen in Figure 5.

1.2 Politics and Twitter

Politics and twitter are very closely related. Almost all the politicians are on twitter and their tweets are waited for and read by a lot of people.

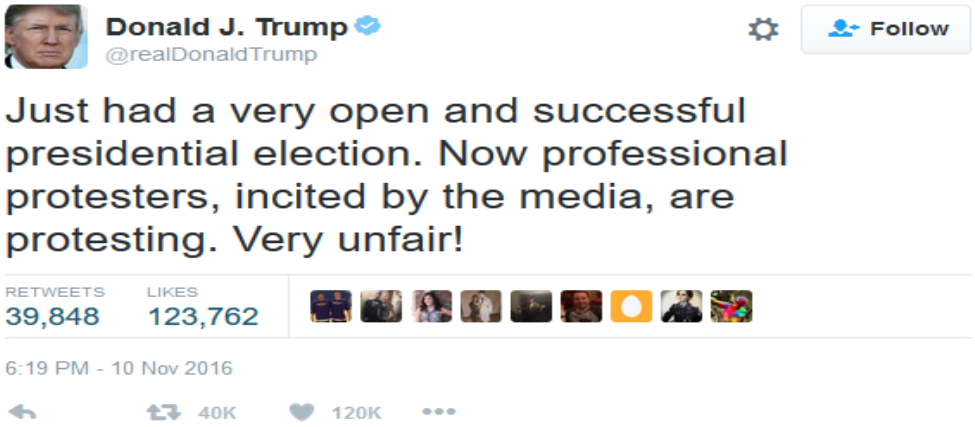

Figure 6- Tweet popularity

As can be seen from Figure 6 about 40 thousand people have retweeted this tweet by trump. This is a tweet when trump was a presidential candidate and simply by these numbers people tend to believe in the popularity of a person or his acceptance by the society.

This relation is becoming so important and integral to politics that even the probability of a candidate being elected is based on his popularity on twitter. These results are quite credible as proved in the last US presidential elections. As shown in Figure 7 at the end of the presidential election Trump was mentioned more in user's tweets than any other candidate and proved to be the final winner in the end.

Figure 7-Twitter Popularity (https://www.feicho.com/watson-alchemy-twitter-analysis-us-presidential-election/)

1.3 Twitter and Pakistani Politics

Pakistani politics is not a stranger to global trends and its politicians go through the same practices to get their views out there in the public. In fact, the social media effect in politics specially Twitter is much more intense and active in Pakistan. Many politicians take up to twitter to criticize others and speak about their agenda freely.

Figure 8- Imran's Tweet

As it can be observed in Figure 8 how openly politicians are expressing their views and criticizing other politicians. Pakistani's these days believe it to be more important to have

an upper hand on twitter my it be criticizing or explaining themselves as can be seen in Figure 9

Figure 9- Maryam Nawaz Tweet

1.4 Media and its interests in Twitter

The media in question here is both the electronic and print media, Majorly the News channels. In the modern world it has become a practice in Pakistan to quote or take reference of a politician's tweets whenever discussing political concerns or political views of a specific person.

Figure 10- Trump News

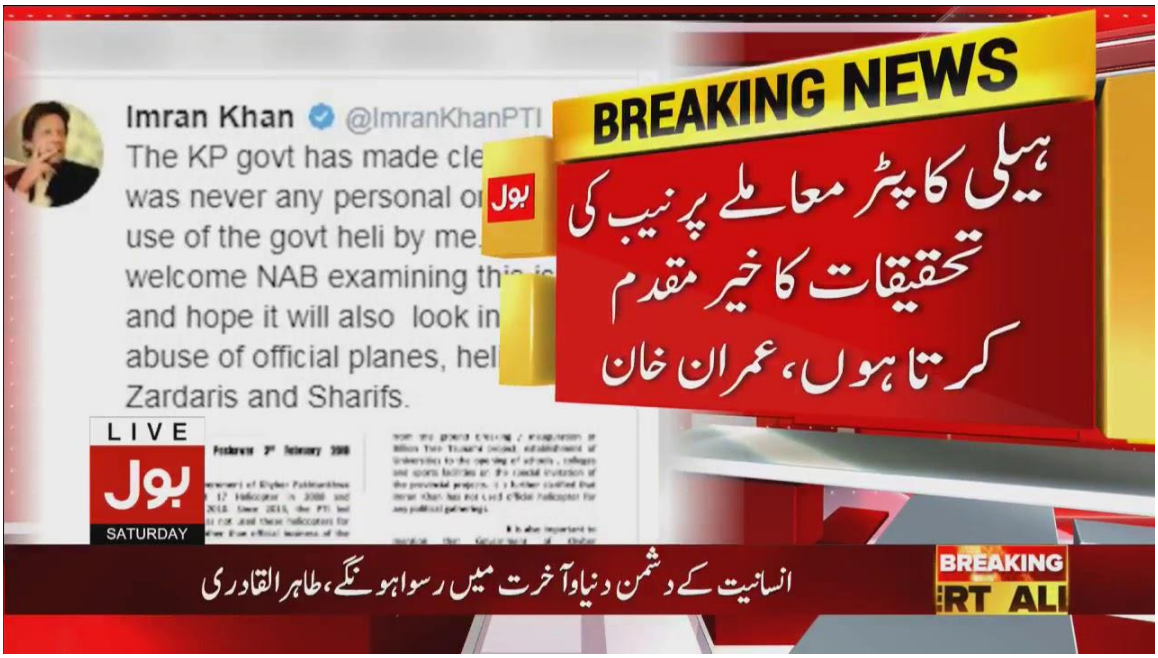

Figure 11- Imran News

As can be seen in Figure 10 and Figure 11 news channels make it their top priority to report to people what have their politicians tweeted on a certain subject. This is making more and

more people focused on twitter and it is becoming an important factor for today's affairs may it be politics or anything else.

1.5 Problem and contribution.

As described in the above sections that a lot of focus and people's trust is being put in Twitter, so there are many things to be careful off. Since people form opinions based on the statistics on twitter which could lead to a different outcome to different situations it is very important to maintain and check that all the statistics being presented are true and unaltered.

The fact is Twitter statistics are not as pure as considered. Due to immense trust of people on these statistics and these statistics being an important popularity measure specially in politics have given birth to a lot of unfair and manipulating means to trick people into someone's personal agenda.

This research tends to figure out these measures prominently in the shape of bots and bring a clearer picture of the depth of manipulation and falls figures being shown on twitter.

CHAPTER 2. LITERATURE REVIEW

2.1 CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks

Social media has gained much popularity in the recent years so much so that social media is the major source of information these days. We trust social media in every aspect of our daily lives from news to products. This popularity gathers much attention from advertisement agencies and business and the social media market is of billions of dollars now. With social media comes the problem of opinion diversity and trust issues. With so much out there it's difficult to decide who to trust. This is solved by the likes mechanism on Facebook. The more likes your page or your post has the more people trust your content and more credible it becomes. The more credible your content is the more traffic it would gather and is a potential target for advertising companies to show their ads so that majority of people watch them. With the amount of many involved people try to fake popularity with any means necessary deceiving users into trusting their content. This has become a major problem as this could lead to serious consequences as the information is most likely skewed and the highest bidder's agenda would be pushed.

Facebook has many security checks in place to essentially block this sort of activity and it is very difficult to have fake likes, due to this the adversaries are limited to using a few accounts over and over again to generate fake likes. To address this issue and to catch the adversary accounts the paper proposes CopyCatch an advance algorithm that based on the timing the relevant accounts like the same pages detect the adversaries. The idea behind the algorithm is that the malicious accounts would like the same set of pages in small differential amount of time and detecting this activity would reveal the fakers. This is known as the lockstep behaviour and CopyCatch limits the amount of likes a group of

accounts can make in a differential amount of time to stop the adversary from making greedy attacks.

2.2 A sock puppet detection algorithm on virtual spaces

As this is a digital era, online social networks have gained significant user base and are the most popular sites on the Web these days. Users with similar interests can join the same topic of discussion and other users comment on this topic to continue the discussion. The fast-paced improvement and globalization of online virtual networks have improved and enhanced our lives of gossip & entertainment, but it also gives equal opportunities for deceptions, such as users' ID theft, article counterfeit, and swindle. On virtual Networks, individuals using multiple identities (usernames) to communicate with others are usually called ''sock puppet''. These fake identities are used to support or create the illusion of support for one's view, pretending to be a different person.

In this paper, a sock puppet detection algorithm which combines authorship-identification techniques and link analysis is proposed. Firstly, an interesting social network model (SVN) is proposed in which links between two IDs are built if they have similar attitude to most topics. Then, the edges are filtered according to a hypothesis testing: (1) if two IDs are connected in the network, particular comment sets of these users are extracted from dataset to check their writing features; (2) the null hypothesis emphasizes that these two sets are from the same person; (3) the test value T is calculated; (4) in a given test level, if the null hypothesis is true, the edge is kept, otherwise, it is removed. Finally, the link-based community detection for pruned network is performed. The pruned SVN is called sock-

puppet network (SPN). Finally, the link-based community detection for SPN is performed. The algorithm is efficient that it can be applied to real dataset.

This paper technique can be applied to variety of use cases on the online community where the opinion of different users is trusted based on the endorsement by the community. Rather than simply detecting fake identities it potential marks people trying to do false business and make money.

2.3 The Follower Count Fallacy: Detecting Twitter Users with Manipulated Follower Count

Online Social Networks (OSN) are increasingly being opted as a platform for effective communication or engagement with other users. The worth of a user on an OSN is via number of likes, followers and shares. Such metrics and crowd-sourced ratings give the OSN user a sense of social reputation which the user tries to maintain and boost to be more influential. Users artificially bolster their social reputation via black-market web services. In this work, the authors have tried to identify users which manipulate their projected follower count using an unsupervised local neighborhood detection method. They identified a neighborhood of the user based on a robust set of features which reflect user similarity in terms of the expected follower count. It is further shown that follower count estimation using their method has 84.2% accuracy with a low error rate. In addition, they estimate the follower count of the user under suspicion by finding its neighborhood drawn from a large random sample of Twitter. They have shown that their method is highly tolerant to synthetic manipulation of followers. Using the deviation of predicted follower

count from the displayed count, they were also able to detect customers with a high precision of 98.62%.

2.4 Online Human-Bot Interactions: Detection, Estimation, and Characterization

Increasing evidence suggests that a growing amount of social media content is generated by autonomous entities known as social bots. This work presents a framework to detect such entities on Twitter. The authors leverage more than a thousand features extracted from public data and meta-data about users: friends, tweet content and sentiment, network patterns, and activity time series. They benchmark the classification framework by using a publicly available dataset of Twitter bots. This training data is enriched by a manually annotated collection of active Twitter users that include both humans and bots of varying sophistication. Their models yield high accuracy and agreement with each other and can detect bots of different nature. Their estimates suggest that between 9% and 15% of active Twitter accounts are bots. Characterizing ties among accounts, it was observed that simple bots tend to interact with bots that exhibit more human-like behaviors. Analysis of content flows reveals retweet and mention strategies adopted by bots to interact with different target groups. Using clustering analysis, they were able to characterize several subclasses of accounts, including spammers, self-promoters, and accounts that post content from connected applications.

CHAPTER 3. DATA COLLECTION

To confirm the involvement of illegal and unfair stakeholders in manipulating the popularity of various users and topics on Twitter, different types of twitter data needed to be collected and in hierarchical manner. The data collections were as follow:

3.1 Trends

We started with gathering the most popular trends of 2017 in Pakistan. The trends were scraped from the website www.trendogate.com which stores all twitter trends for various geo location and dates.

```
1. intilize start_date
2. intialize end_date
3. set country_code to 23424922
4. set url to 'https://trendogate.com/placebydate/' + country_code
5. make a list of dates in range from start to end dates
6. for every date in the list
7.  print date in output file
8.  get the page at url appended with '/' + date
9.  intialize an empty list
10. parse the page for text
11. for every trend in the page
12.     write the trnd to output file
13. close outputt file
```

Figure 12- Trends code

Once we gathered all the trends for the year of 2017 specifically for Pakistan which turned out to be about 5000 in total we moved to further data collection.

3.1.1 Tweets

Once all the trends on twitter for 2017 were recorded we gathered all the tweets made about those trends in the year 2017. The tweets were saved in a text format with features like

‘tweet id’, ‘text of tweet’, ‘number of like on tweet’, ‘number of retweets of the tweet’, ‘number of likes on the tweet’, ‘number of replies on the tweet’, ‘full name of the user making the tweet’, ‘user id of the user making the tweet’ and ‘the time the tweet was made’. All these tweets were gathered using an online scrapper which can be found at https://github.com/taspinar/twitterscraper. The scrapper was tweaked to our specific needs.

3.1.2 Political Trends

Once all the tweets for every trend were gathered the next step was to filter out trends that had political inclination. From 5235 trends 1000 trends were related to Pakistani politics and were filtered out manually.

3.1.3 Filtering tweets

For every political trends a separate record was made for all the tweets that had been retweeted at least once.

```
for every trend in political trends
    load all the data from the tweets file
    for every tweet in the data
        convert tweet to a dictionary
        if count of retweets > 0
            write tweet to the output file
            close output file
```

Figure 13- Filtering Retweets

3.1.4 Retweeters id

Once we had records for tweets that were retweeted at least once for all the political trends the next step was to obtain the user ids of the twitter users that retweeted a particular tweet. These ids and any further data collection was done through Twitter’s API.

```
intialize API_key with twitter provided key
intialize API_secret with twitter provided value
intialize Access_Token with twitter provided token
intialize Acess_token_secret with twitter provided token secret
build a connection with twitter api by passing the above 4 parameters
for every trend
    load tweets
    make the tweet data into a dictionary
    extract tweet id
    intialize next_cursor to -1
    while next_cursor not equal to 0
        send a request to twitter api for retweeters id with parameters (tweet id and next_cursor)
        if request returns a rate limit error
            wait for 15 minutes
            send request again
        add the ids to the tweet dictionary
        update next_cursor
    save the dictionary with the ids to the output file
    close file
```

Figure 14- Retweeters ids

3.1.5 Unique Retweeters per trend

After all the ids of the retweeters per tweet for every trend were collected the next step was to find unique users the retweeted a trend.

```
for every trend
    load all the tweets data
    creat an empty list
    for every tweet in the data
        convert data to a dictionary
        extract all the ids
        for all ids
            if id not in the list
                add id to the list
                print id to output file
        close file
```

Figure 15- unique retweeters

3.1.6 Retweeter profile info

When we had all the unique retweeters that retweeted a particular trend the next step was to scrape the profile info of all these users.

```
intialize API_key with twitter provided key
intialize API_secret with twitter provided value
intialize Access_Token with twitter provided token
intialize Acess_token_secret with twitter provided token secret
build a connection with twitter api by passing the above 4 parameters
for every trend
    load all the unique retweeters
    for every single user
        send request to twitter api to show user with the parameter being the user id
        if request returns a rate limit error
            wait for 15 minutes
            send request again
        save the user profile info in the output file
        close file
```

Figure 16-Retweeter Profile info

3.1.7 Retweeters profile activity

After collecting the user profile info of every unique retweeter per trend, the next step was to scrape the profile activity of these retweeters. The profile activity included all the tweets and retweets made by the respective user sin 1st January 2017.

```
intialize API_key with twitter provided key
intialize API_secret with twitter provided value
intialize Access_Token with twitter provided token
intialize Acess_token_secret with twitter provided token secret
build a connection with twitter api by passing the above 4 parameters
for every trend
    load all the unique retweeters
    for every user
        extract the user name
        create an empty list by the name all_tweets
        create an empty list by the name new_tweets
        if user profile scraped before
            load all the data in new_tweets
            set oldest to 1 less the id of the last item in new_tweets
            convert the time stamp of last item to normal date and time format
        else if user not scraped before
            send request to twitter api to get user timeline with parameters username, count = 200 and include retweets
            if request returns a rate limit error
                wait for 15 minutes
                send request again
            save the data to new_tweets
            add new_tweets to all_tweets
            set oldest to 1 less than the id of the last tweet/retweet in all_tweets
            convert the time stamp of last item to normal date and time format
        set stop date to 2017-01-01
        while the last item scraped id greater than or equal to stop date and new_tweets is not empty
            delete data already in new_tweets
            send request to twitter api to get user timeline with parameters username, count = 200, include retweets and oldest id
            if request returns a rate limit error
                wait for 15 minutes
                send request again
            save the data to new_tweets
            add new_tweets to all_tweets
            set oldest to 1 less than the id of the last tweet/retweet in all_tweets
            convert the time stamp of last item to normal date and time format
        for every tweet in all_tweets
            extract id,timestamp,text,original user name
            save to output file
```

Figure 17- Retweeter timeline

3.2 Influential users

Separately the top 500 influential user's name and twitter id were scrapped. The users were ranked by the website "social bakers" on the number of followers each of them have.

3.2.1 Followers profile info

Among the 500 influential users the top 20 were chosen. From those top 20 influential users the profile info of 30,000 followers each were collected using the Twitter API.

```
intialize API_key with twitter provided key
intialize API_secret with twitter provided value
intialize Access_Token with twitter provided token
intialize Acess_token_secret with twitter provided token secret
build a connection with twitter api by passing the above 4 parameters
load the ranked users
select the first 20
for every user in the 20
    extract user_name
    create an empty list
    set next_cursor to -1
    while the list does not hve 30000 entries
        send request to twitter api for followers list with parameters (user_name, next_cursor, count=200, skip_status = 1)
        if request returns a rate limit error
            wait for 15 minutes
            send request again
        add data received to list
        update next_cursor
    convert list to dictionary
    create a table from dictionary
    save table to output file
```

Figure 18- follower info

3.2.2 Follower timeline

When all the profile info of 30,000 followers of each influential user among the top 20 was collected, the next step was to scrape the profile activity of these 30,000 users since 1st January 2017.

```
intialize API_key with twitter provided key
intialize API_secret with twitter provided value
intialize Access_Token with twitter provided token
intialize Acess_token_secret with twitter provided token secret
build a connection with twitter api by passing the above 4 parameters
for every user in the 20
    load all the followers info
    for every follower
        extract the user name
        extract user_id
        create an empty list by the name all_tweets
        create an empty list by the name new_tweets
        if user profile scraped before
            load all the data in new_tweets
            set oldest to 1 less the id of the last item in new_tweets
            convert the time stamp of last item to normal date and time format
        else if user not scraped before
            send request to twitter api to get user timeline with parameters user_id, count = 200 and include retweets
            if request returns a rate limit error
                wait for 15 minutes
```

Figure 19- followers timeline I

```
                send request again
            save the data to new_tweets
            add new_tweets to all_tweets
            set oldest to 1 less than the id of the last tweet/retweet in all_tweets
            convert the time stamp of last item to normal date and time format
        set stop date to 2017-01-01
        while the last item scraped id greater than or equal to stop date and new_tweets is not empty
            delete data already in new_tweets
            send request to twitter api to get user timeline with parameters user_id, count = 200, include retweets and oldest id
            if request returns a rate limit error
                wait for 15 minutes
                send request again
            save the data to new_tweets
            add new_tweets to all_tweets
            set oldest to 1 less than the id of the last tweet/retweet in all_tweets
            convert the time stamp of last item to normal date and time format
        for every tweet in all_tweets
            extract id,timestamp,text,original user name
            save to output file
        close file
```

Figure 20- followers timeline II

CHAPTER 4. DATA ANALYSIS & MACHINE LEARNING MODEL

4.1 Machine Learning Model

In order to predict the percentage of bots or the level of penetration of illegal activity on the data collected a machine learning model was built. As it is a classification problem to predict whether a user is a potential Bot or not an off the shelf classifier was trained. Since we had a labelled dataset of Bots on Twitter the learning was supervised where about 27 features from a person's profile on twitter were used.

4.1.1 Train and Test Data

To build any Machine learning model a labelled dataset is required for the learning of the machine. We got a labelled dataset from Indiana University Network Science Institute & Complex Networks and Systems Research's joint project repository. The repository can be located at https://botometer.iuni.iu.edu/bot-repository/datasets.html. The dataset named "cresci-2017" was used.

4.1.2 Extracting Features

On viewing the dataset, it was in the form of text with Bots user profile info and profile activity. From the available information the following features were extracted:

statuses_count, followers_count, Friends_count, Listed_count, default_profile, default_profile_image, verified, avg time in sec between consecutive activity, avg time in sec between consecutive tweets, avg time in sec between consecutive retweets, max time

in sec between consecutive activity, max time in sec between consecutive tweets, max time in sec between consecutive retweets, min time in sec between consecutive activity, min time in sec between consecutive tweets, min time in sec between consecutive retweets, total_tweets, total_retweets, days_active, scree_name_length, digits in screen name, description length, specific words in description, specific words in screen name.

in the above features the words mentioned were a manually selected list of words believed to be commonly found in Bot profiles and the words are as follow:

```
bag_of_words_bot = r'bot|b0t|cannabis|tweet me|mishear|follow me|updates every|gorilla|yes_ofc|forget' \
                   r'expos|kill|clit|bbb|butt|fuck|XXX|sex|truthe|fake|anony|free|virus|funky|RNA|kuck|jargon' \
                   r'nerd|swag|jack|bang|bonsai|chick|prison|paper|pokem|xx|freak|ffd|dunia|clone|genie|bbb' \
                   r'ffd|onlyman|emoji|joke|troll|droop|free|every|wow|cheese|yeah|bio|magic|wizard|face'
```

Figure 21- bag of words

Figure 21 is source code snippet. The code was in python hence the formatting. All the features had values in binary format for ease of training the model. As the features were extracted in two parts first the profile info and them the profile activity for each user the id of users was kept till last for merging the data to correct records.

The pseudocode of the two stages is as shown in Figure 22 & Figure 23. All the values of the features are binary for the ease of training a learning model. If a feature is true its value is 1 else its 0.

```
load labled Bot and genuine account profile info data
merge all the data to single data object
extract the decided features from the data object for every record
convert the timestamp to readable date
create empty lists for days lengths digits description words and screen name words
define the words expected to be found in Bot profiles
for every record in the data object
    subtract the creation date from a set date
    add the difference of days active in days list
    intialize count to 0
    add the count of number of characters in screen_name to length list
    for every character in the screen name
        if its an integer
            increse count by one
        else
            continue to next character
    add the count to digits list
    if the user has some description
        count the number of characers in description
        add the count to description list
    else
        add 0 to description list
    if words present in description
        add 1 to words list
    else
        add 0 to word list
    if words present in screen name
        add 1 to screen name words list
    else
        add 0 to screen name words list
add all of the lists to data objects as profile information features
save data to output file
```

Figure 22- Profile Info Features

```
load all the tweets data for Bots and Real users
merge all the data
drop any null records
extract data for (text,create_at,user_id) from the data object for all records
extract unique id values in the data object
for every unique id
    for every record in data objectect
        if id matches
            if text has RT
                increase retweet_count by 1
                subtracted retweet time from last retweet time
                save time difference in a list
            else
                increase tweet count by 1
                subtract tweet time from previous tweet time
                save time differencce in a list
            subtrct time from last activity time
            save data in a list
        else
            continue to next record
    delete all the records for this id
    take the maximum, minimum and average of each list
    add these features to a new record with user_id as identifier
save the data object with features of every record to out put file
```

Figure 23- profile activity features

4.1.3 Training model and Testing for Accuracy

Once all the data was saved with the featured decided for training and testing the data, the next step was to train a machine learning model with different classifiers and to select the one with the best accuracy. All classifiers like Logistic regression, Random Forest, Decision Tree, Linear Discriminant Analysis, Naïve Bayes and support Vector were trained and tested.

The Best classifier turned out to be Linear Discriminant Analysis (LDA) with an accuracy of 87%. The second best was Random Forrest with 84% accuracy.

```
1  Load the data with all the required features
2  drop any null values
3  split the data in to train test data sets
4  train different classifiers with train data
5  check the accuracy of predictin for every classifier though the test data
6  print accuracy
```

Figure 24- Machine Learning Model

4.2 Data Analysis

Once the machine model was prepared and decided, the next step was to extract the same features from our collected data and prepare it for prediction.

4.2.1 Trends

The same features were extracted for all the retweeters of the top 30 political tends of 2017 on twitter in Pakistan. The source code can be seen in the preceding figures.

```
import os
import pandas as pd
import time
import ast
import json
from datetime import datetime,date
trends = []
for (path,dir,file) in os.walk('top_30') :
    trends.extend(dir)
    break
for trend in trends:
    last_final = pd.DataFrame()
    if os.path.isfile(path + '/' + trend + '/retweeters_final.csv'):
        continue
    print("-------------------------------------------------")
    print('working on retweeters of:', trend)
    retweeters = []
    for(add,fol,files) in os.walk(path + '/' + trend + '/retweeters'):
        retweeters.extend(fol)
        break
    i = 1
    retweeters_list = []
    for retweeter in retweeters:
        print("*****************************************")
        print('processing retweeter: ' + str(i) + '/' + str(len(retweeters)))
        print(retweeter)
        i += 1
        if os.path.isfile(add + '/' + retweeter + '/final.csv'):
            ser = pd.Series.from_csv(add + '/' + retweeter + '/final.csv',sep=',')
            frame = ser.to_frame()
            frame = frame.transpose()
            last_final = last_final.append(frame)
            continue
        final_dict = {}
```

Figure 25- Trends Features I

```
    try:
        with open(add + '/' + retweeter + '/user_profile_info.txt','r') as f:
            for line in f:
                line = line.replace('\n','')
                line = line.replace('"','')
                line = line.replace("{" , "")
                line = line.replace("}" , "")
                line = line.replace(" ", "")
                temp = line.split(',')
                for item in temp:
                    index = item.find(":")
                    final_dict[item[0:index]] = item[index+1:]
            final_dict['timestamp'] = time.strftime('%Y-%m-%d %H:%M:%S', time.strptime(final_dict['created_at'],'%a%b%d%H:%M:%S+0000%Y'))
    except FileNotFoundError:
        continue
    user_dict = {}
    user_dict[final_dict['id']] = {}
    user_dict[final_dict['id']]['profile_activity'] = []
    user_dict[final_dict['id']]['retweet_activity'] = []
    user_dict[final_dict['id']]['tweet_activity'] = []
    user_dict[final_dict['id']]['total_tweets_user'] = 0
    user_dict[final_dict['id']]['total_retweets_user'] = 0
    last_activity = 0
    previous_retweet = 0
    previous_tweet = 0
    try:
        temp = pd.read_csv(add + '/' + retweeter + '/user_timeline.csv',sep=',',usecols=['text','created_at'],engine='python')
        temp['created_at'] = pd.to_datetime(temp['created_at'])
    except ValueError:
        temp = pd.read_csv(add + '/' + retweeter + '/user_timeline.csv',sep=',',header=None,names=['id','created_at','text','original_user'],
            engine='python')
        temp = temp[['created_at','text']]
        temp['created_at'] = pd.to_datetime(temp['created_at'])
    except FileNotFoundError:
        continue
```

Figure 26- Tends Features II

```
    except:
        try:
            temp = pd.read_csv(add + '/' + retweeter + '/user_timeline.csv',sep=',',engine='python')
            temp = temp.rename(columns={ temp.columns[1]: "created_at",temp.columns[2]:"text" })
            temp = temp[:,1:3]
            temp['created_at'] = pd.to_datetime(temp['created_at'])
        except:
            print('error')
            continue
    if not temp.empty:
        for number,line in temp.iterrows():
            if last_activity == 0:
                last_activity = line['created_at']
            else:
                time_diff_activity = last_activity - line["created_at"]
                last_activity = line["created_at"]
                user_dict[final_dict['id']]['profile_activity'].append(time_diff_activity.total_seconds())
            if "RT" in str(line["text"]):
                user_dict[final_dict['id']]['total_retweets_user'] += 1
                if previous_retweet == 0:
                    previous_retweet = line['created_at']
                else:
                    time_diff_retweets = previous_retweet - line['created_at']
                    previous_retweet = line['created_at']
                    user_dict[final_dict['id']]['retweet_activity'].append(time_diff_retweets.total_seconds())
            else:
                user_dict[final_dict['id']]['total_tweets_user'] += 1
                if previous_tweet == 0:
                    previous_tweet = line["created_at"]
                else:
                    time_diff_tweets = previous_tweet - line["created_at"]
                    previous_tweet = line["created_at"]
                    user_dict[final_dict['id']]['tweet_activity'].append(time_diff_tweets.total_seconds())
    for key,value in user_dict.items():
        final_dict["total_tweets"] = user_dict[key]['total_tweets_user']
```

Figure 27- Trends Features III

```
            final_dict["total_retweets"] = user_dict[key]['total_retweets_user']
            if user_dict[key]['profile_activity'] != []:
                final_dict["avg_profile_activity(sec)"] = sum(user_dict[key]['profile_activity'])/len(user_dict[key]['profile_activity'])
                final_dict['min_profile_activity(sec)'] = min(user_dict[key]['profile_activity'])
                final_dict['max_profile_activity(sec)'] = max(user_dict[key]['profile_activity'])
            else:
                final_dict["avg_profile_activity(sec)"] = 0
                final_dict['min_profile_activity(sec)'] = 0
                final_dict['max_profile_activity(sec)'] = 0
            if user_dict[key]['tweet_activity'] != []:
                final_dict["avg_tweet_activity(sec)"] = sum(user_dict[key]['tweet_activity'])/len(user_dict[key]['tweet_activity'])
                final_dict['min_tweet_activity(sec)'] = min(user_dict[key]['tweet_activity'])
                final_dict['max_tweet_activity(sec)'] = max(user_dict[key]['tweet_activity'])
            else:
                final_dict["avg_tweet_activity(sec)"] = 0
                final_dict['min_tweet_activity(sec)'] = 0
                final_dict['max_tweet_activity(sec)'] = 0
            if user_dict[key]['retweet_activity'] != []:
                final_dict["avg_retweet_activity(sec)"] = sum(user_dict[key]['retweet_activity'])/len(user_dict[key]['retweet_activity'])
                final_dict['min_retweet_activity(sec)'] = min(user_dict[key]['retweet_activity'])
                final_dict['max_retweet_activity(sec)'] = max(user_dict[key]['retweet_activity'])
            else:
                final_dict["avg_retweet_activity(sec)"] = 0
                final_dict['min_retweet_activity(sec)'] = 0
                final_dict['min_retweet_activity(sec)'] = 0
        retweeter_dataframe = pd.Series(final_dict)
        retweeter_dataframe.to_csv(add + '/' + retweeter + '/final.csv')
        retweeters_list.append(final_dict)
    last = pd.DataFrame.from_dict(retweeters_list)
    last_final = last_final.append(last)
    last_final = last_final.replace(to_replace=['True','False'],value=[1,0])
    last_final = last_final.replace(to_replace=['true','false'],value=[1,0])
    last_final = last_final.replace(to_replace=[True,False],value=[1,0])
    last_final = last_final.fillna(0)
```

Figure 28- Trends Features IV

```
    features_profile = ['id','screen_name','name', 'statuses_count', 'followers_count','friends_count', 'favourites_count', 'listed_count',
                        'default_profile', 'default_profile_image','verified','description','timestamp','avg_profile_activity(sec)',
                        'avg_retweet_activity(sec)','avg_tweet_activity(sec)','max_profile_activity(sec)','max_retweet_activity(sec)',
                        'max_tweet_activity(sec)','min_profile_activity(sec)','min_retweet_activity(sec)','min_tweet_activity(sec)','total_retweets',
                        'total_tweets',]
    last_final = last_final[features_profile]
    bag_of_words_bot = r'bot|b0t|cannabis|tweet me|mishear|follow me|updates every|gorilla|yes_ofc|forget' \
                           r'expos|kill|clit|bbb|butt|fuck|XXX|sex|truthe|fake|anony|free|virus|funky|RNA|kuck|jargon' \
                           r'nerd|swag|jack|bang|bonsai|chick|prison|paper|pokem|xx|freak|ffd|dunia|clone|genie|bbb' \
                           r'ffd|onlyman|emoji|joke|troll|droop|free|every|wow|cheese|yeah|bio|magic|wizard|face'
    user_dict = {}
    i = 1
    days = []
    lengths = []
    digits = []
    lengths_des = []
    words = []
    for index,row in last_final.iterrows():
        diff = datetime.today() - datetime.strptime(row['timestamp'],'%Y-%m-%d %H:%M:%S')
        days.append(diff.days)
        count = 0
        length = len(row['screen_name'])
        lengths.append(length)
        for character in row["screen_name"]:
            try:
                int(character)
                count += 1
            except:
                continue
        digits.append(count)
        if row["description"] != 0:
            length = len(row["description"])
            if bag_of_words_bot in row["description"]:
                words.append(1)
            else:
                words.append(0)
        else:
            length = 0
```

Figure 29- Trends Features V

```
                words.append(0)
            lengths_des.append(length)
        last_final['days_active'] = days
        del last_final['timestamp']
        last_final['Screen_name_length'] = lengths
        last_final['digits_in_name'] = digits
        last_final["description_length"] = lengths_des
        last_final["contains_words"] = words
        last_final.to_csv(path + '/' + trend + '/retweeters_final.csv',index=False)
print("completed")
```

Figure 30- Trends Features VI

4.2.2 Influential Users Followers

As shown in figures from Figure 25-Figure 30 the same code with a little alteration were used to extract the features of the followers of influential users. They were than saved to a file so that the model could be used to predict the number of fake followers present.

CHAPTER 5. RESULTS

5.1 Bots in Followers

The trained model was ran on the followers of each influential user and the results are showed in Table 1.

Table 1- Bots in Followers

	total followers	total Bots	%age of Bots Present
AsadUmar	29990	29871	99.60320106702234
BilawalBhuttoZardari	29996	29780	99.2799039871983
HamidMir	29993	29823	99.43320108025205
ImranKhan	29992	29862	99.56655108028808
JahangirKhanTareen	29995	29837	99.47324554092349
KamranKhan	29997	29779	99.27326065939927
KashifAbbasi	29999	29839	99.46664888829628
MaryamNawazSharif	29989	29797	99.35976524725733
MubasherLucman	29991	29784	99.30979293788137
NajamSethi	29994	29851	99.52323798092952
PTI	27115	26887	99.15913700903559
ShahzebKhanzada	29994	29826	99.43988797759552
ShehbazSharif	29991	29854	99.543196292221
SheikhRashidAhmad	29997	29851	99.51328466179952
WaseemBadami	29994	29847	99.50990198039608
fatimabhutto	29994	29729	99.11648996465959

Further it was tested to see how many of the bots are common among all these influential users. These results are displayed in Table 2.

Table 2- Common Bots

intersection between	common_bots
AsadUmar&HamidMir	20633
AsadUmar&MaryamNawazSharif	17037
AsadUmar&MubasherLucman	290
AsadUmar&PTI	10142
AsadUmar&ShehbazSharif	18708
AsadUmar&SheikhRashidAhmad	17270
BilawalBhuttoZardari&AsadUmar	13715
BilawalBhuttoZardari&HamidMir	14424

BilawalBhuttoZardari&ImranKhan	10747
BilawalBhuttoZardari&JahangirKhanTareen	15807
BilawalBhuttoZardari&KamranKhan	10140
BilawalBhuttoZardari&MaryamNawazSharif	13970
BilawalBhuttoZardari&MubasherLucman	764
BilawalBhuttoZardari&NajamSethi	18263
BilawalBhuttoZardari&PTI	8821
BilawalBhuttoZardari&ShahzebKhanzada	11553
BilawalBhuttoZardari&ShehbazSharif	14836
BilawalBhuttoZardari&SheikhRashidAhmad	14054
BilawalBhuttoZardari&WaseemBadami	16832
BilawalBhuttoZardari&fatimabhutto	7335
GenAsimBajwa&AsadUmar	225
GenAsimBajwa&BilawalBhuttoZardari	398
GenAsimBajwa&HamidMir	308
GenAsimBajwa&ImranKhan	218
GenAsimBajwa&JahangirKhanTareen	333
GenAsimBajwa&KamranKhan	599
GenAsimBajwa&KashifAbbasi	252
GenAsimBajwa&MaryamNawazSharif	286
GenAsimBajwa&MubasherLucman	489
GenAsimBajwa&NajamSethi	304
GenAsimBajwa&PTI	521
GenAsimBajwa&ShahzebKhanzada	482
GenAsimBajwa&ShehbazSharif	290
GenAsimBajwa&SheikhRashidAhmad	373
GenAsimBajwa&WaseemBadami	324
GenAsimBajwa&fatimabhutto	433
HamidMir&MaryamNawazSharif	20111
HamidMir&MubasherLucman	404
HamidMir&PTI	10788
HamidMir&ShehbazSharif	17531
HamidMir&SheikhRashidAhmad	19214
ImranKhan&AsadUmar	19919
ImranKhan&HamidMir	16311
ImranKhan&KamranKhan	5270
ImranKhan&MaryamNawazSharif	13525
ImranKhan&MubasherLucman	249
ImranKhan&PTI	8561
ImranKhan&ShehbazSharif	15015
ImranKhan&SheikhRashidAhmad	13681
JahangirKhanTareen&AsadUmar	15948
JahangirKhanTareen&HamidMir	13271
JahangirKhanTareen&ImranKhan	11702
JahangirKhanTareen&KamranKhan	8927
JahangirKhanTareen&MaryamNawazSharif	11980
JahangirKhanTareen&MubasherLucman	506
JahangirKhanTareen&PTI	7909
JahangirKhanTareen&ShahzebKhanzada	6926
JahangirKhanTareen&ShehbazSharif	17774
JahangirKhanTareen&SheikhRashidAhmad	13031
JahangirKhanTareen&WaseemBadami	15473

JahangirKhanTareen&fatimabhutto	5631
KamranKhan&AsadUmar	6783
KamranKhan&HamidMir	8329
KamranKhan&MaryamNawazSharif	7039
KamranKhan&MubasherLucman	8081
KamranKhan&PTI	8796
KamranKhan&ShehbazSharif	8404
KamranKhan&SheikhRashidAhmad	8405
KashifAbbasi&AsadUmar	13889
KashifAbbasi&BilawalBhuttoZardari	10335
KashifAbbasi&HamidMir	14110
KashifAbbasi&ImranKhan	10473
KashifAbbasi&JahangirKhanTareen	10789
KashifAbbasi&KamranKhan	8443
KashifAbbasi&MaryamNawazSharif	11110
KashifAbbasi&MubasherLucman	430
KashifAbbasi&NajamSethi	10587
KashifAbbasi&PTI	8399
KashifAbbasi&ShahzebKhanzada	7145
KashifAbbasi&ShehbazSharif	11425
KashifAbbasi&SheikhRashidAhmad	12439
KashifAbbasi&WaseemBadami	12932
KashifAbbasi&fatimabhutto	5110
MaryamNawazSharif&MubasherLucman	373
MaryamNawazSharif&PTI	10130
MaryamNawazSharif&ShehbazSharif	17842
MaryamNawazSharif&SheikhRashidAhmad	18510
MubasherLucman&PTI	5021
MubasherLucman&ShehbazSharif	368
MubasherLucman&SheikhRashidAhmad	527
NajamSethi&AsadUmar	15981
NajamSethi&HamidMir	15567
NajamSethi&ImranKhan	12258
NajamSethi&JahangirKhanTareen	17741
NajamSethi&KamranKhan	9625
NajamSethi&MaryamNawazSharif	15036
NajamSethi&MubasherLucman	430
NajamSethi&PTI	8438
NajamSethi&ShahzebKhanzada	10324
NajamSethi&ShehbazSharif	17950
NajamSethi&SheikhRashidAhmad	15054
NajamSethi&WaseemBadami	18313
NajamSethi&fatimabhutto	6801
ShahzebKhanzada&AsadUmar	7778
ShahzebKhanzada&HamidMir	9395
ShahzebKhanzada&ImranKhan	6068
ShahzebKhanzada&KamranKhan	15385
ShahzebKhanzada&MaryamNawazSharif	8399
ShahzebKhanzada&MubasherLucman	10038
ShahzebKhanzada&PTI	9114
ShahzebKhanzada&ShehbazSharif	6865
ShahzebKhanzada&SheikhRashidAhmad	10178

ShahzebKhanzada&WaseemBadami	10523
ShahzebKhanzada&fatimabhutto	9222
ShehbazSharif&PTI	8233
ShehbazSharif&SheikhRashidAhmad	14464
SheikhRashidAhmad&PTI	13470
WaseemBadami&AsadUmar	15761
WaseemBadami&HamidMir	16177
WaseemBadami&ImranKhan	12363
WaseemBadami&KamranKhan	9580
WaseemBadami&MaryamNawazSharif	14902
WaseemBadami&MubasherLucman	460
WaseemBadami&PTI	7522
WaseemBadami&ShehbazSharif	15937
WaseemBadami&SheikhRashidAhmad	15608
WaseemBadami&fatimabhutto	6857
fatimabhutto&AsadUmar	4705
fatimabhutto&HamidMir	5254
fatimabhutto&ImranKhan	3732
fatimabhutto&KamranKhan	9559
fatimabhutto&MaryamNawazSharif	4864
fatimabhutto&MubasherLucman	5539
fatimabhutto&PTI	5314
fatimabhutto&ShehbazSharif	5517
fatimabhutto&SheikhRashidAhmad	5480

5.2 Bots among Retweeters of Trends

Next, we checked for the number of Bots among the retweeters of a specific trend. The results are displayed in Table 3.

Table 3- Bots in Trends

Trends	total retweeters	total Bots	%age
#3 Judges	3889	589	15.145281563383902
#AyeshaGulalai	2637	357	13.538111490329921
#BaniGala	1286	436	33.903576982892695
#BenazirBhutto	3660	813	22.213114754098363
#Brexit	3320	324	9.759036144578314
#CPEC	3733	501	13.42084114653094
#CaptainSwan	2259	418	18.503762726870296
#GoNawazGo	3075	624	20.29268292682927
#IshaqDar	2883	505	17.51647589316684
#MQMPakistan	1250	321	25.679999999999996
#MuslimBan	358	28	7.82122905027933
#Mustafakamal	1608	393	24.44029850746269
#NawazSharif	3234	545	16.852195423623996
#Neverforget	528	132	25.0

#NoBanNoWall	5840	674	11.54109589041096
#PPP Sindh	3282	840	25.59414990859232
#PanamaPapers	4364	470	10.76993583868011
#Trump	5656	321	5.675388967468176
#WomensMarch	1806	301	16.666666666666664
#Zardari	2794	361	12.920544022906228
#ayeshagulalai	2631	366	13.911060433295324
#immunity	2974	668	22.46133154001345
#ishaqdar	2974	555	18.661735036987224
#maryamnawaz	3187	559	17.540006275494196
#na-4	1816	243	13.381057268722468
#panamacase	2988	483	16.164658634538153
#pppp	2394	646	26.984126984126984
#sharifs	3300	625	18.939393939393938
#zardari	2694	374	13.882702301410543

5.3 Conclusion

As can be seen in the result for the followers a lot of Bots are found which is a clear indication of the amount of penetration fake activity has in shaping the opinions of the masses. These results are convincing enough to dig deeper into this research and come up with a definite conclusion of such activity and to unmask them to the world.

For the activity of Bots in Trends, it can be seen that an ample amount is present to push fake agendas and mislead the people.

REFERENCES

Varol O, Ferrara E, Davis CA, Menczer F, Flammini A. Online human-bot interactions: Detection, estimation, and characterization. arXiv preprint arXiv:1703.03107. 2017 Mar 9.

Aggarwal A, Kumar S, Bhargava K, Kumaraguru P. The Follower Count Fallacy: Detecting Twitter Users with Manipulated Follower Count.

Bu Z, Xia Z, Wang J. A sock puppet detection algorithm on virtual spaces. Knowledge-Based Systems. 2013 Jan 1;37:366-77.

Beutel A, Xu W, Guruswami V, Palow C, Faloutsos C. Copycatch: stopping group attacks by spotting lockstep behavior in social networks. InProceedings of the 22nd international conference on World Wide Web 2013 May 13 (pp. 119-130). ACM.