Twitter Bot Detection

Muhammad Adeel Irshad

Aneeza Fayyaz

Kashif Hussain

Supervised By

Iqra Chaudhary

Submitted for the partial fulfillment of MCS degree to the Faculty of Engineering & CS

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

September, 2021

ABSTRACT

In recent years, Big Data from various social media apps has transformed the web into a user-generated repository of information in an ever-increasing number of areas. Twitter's popularity and distinct structure have drawn a large number of bots or automated programs. Twitter bots are used for a variety of reasons, including spamming, communicating, and increasing the number of followers.

Twitter bot Detection is a web-based application that works on machine learning. It is developed for the purpose of distinguishing Twitter bots from the accounts, which are being operated by humans. The existing methods only tell us the analysis of the last twenty followers and rate them from 0-5, in which closer to a 5 score means that the account has more chances of being a bot. The problem with that approach is a person doesn't actually get the idea of how much spamming is going on his Timeline. And following them one by one is very hard by opening their profile manually. A naive person who doesn't know about Twitter bots much will never tell the difference between real and bot accounts easily. When machine learning model was trained as a bot or a human, the characteristics of Twitter accounts were used as features, instead of labelling their bot, the proposed application overcomes the idea of simply analyzing their followers and rating them. Moreover, users can also unfollow them. Bot Detection functionality is achieved through a supervised machine learning algorithms i.e. Random Forest.

Every application has its own limitations. The limitations of this project are based on Twitter API. In proposed system-standard version of Twitter API is used. The maximum number of requests that can be made is determined by a time interval or a set amount of time. The time interval which is mostly used in Twitter API is 15 minutes. If an endpoint's rate limit is 900 requests/15 minutes, then every 15-minute period can have up to 900 requests. When these limits are exceeded, user have to wait for 15 minutes time window in order to send the request again. This limit can be overcome by using a paid premium version of Twitter API.

CERTIFICATE

Dated: _____

Final Approval

It is certified that the project report titled **'Twitter Bot Detection'** submitted by **Muhammad Adeel Irshad, Aneeza Fayyaz** and **Kashif Hussain** for the partial fulfillment of the requirement of **"Master's Degree in Computer Science"** is approved.

COMMITTEE

Dr. Basit Shahzad

Dean Engineering & CS: Signature: _____

Dr. Sajjad Haider

HOD Computer Science: Signature: _____

Ms. Mehwish Sabih

Head Project Committee: Signature: _____

Mr. Asim Rehan

Project Coordinator: Signature: _____

Ms. Iqra Chaudhary

Supervisor: Signature: _____

DECLARATION

We hereby declare that our dissertation is unique and authentic. We acknowledge that if any PLAGIARISM is discovered at any point during the process, our group will receive an F (FAIL) grade, and our Master's degree may be revoked.

Group members:

Name	**Signature**
Muhammad Adeel Irshad	_____
Aneeza Fayyaz	_____
Kashif Hussain	_____

PLAGIARISM CERTIFICATE

This is to clarify that the project entitled **"Twitter Bot Detection"**, which is being submitted herewith for the award of the **"Degree of Masters"** in **"Computer Science".** This is the result of the original work by **Muhammad Adeel Irshad, Aneeza Fayyaz and Kashif Hussain** under my supervision and guidance. To the best of my knowledge and belief, the work included in this project has not been done previously for the foundation of any degree, comparable certificate, or similar tile like this for any other diploma/examining body or university.

Turnitin Originality Report

Submission date: 29-Aug-2021 06:05PM (UTC+0500)

Submission ID: 1637593152

File name: Modefied-_Twitter_Bot_Detection_-_2.docx (168.58K)

Word count: 8031

Character count: 4145

Similarity Index 18%

Similarity by Source

Internet Source: 9%

Student Papers: 15%

Publications: 4%

Date: 29-08-2021

_____

Iqra Chaudhary (Supervisor)

TURNITIN ORIGINALITY REPORT

Group based application **Twitter Bot Detection** [MCS] by **Muhammad Adeel Irshad, Aneeza Fayyaz** and **Kashif Hussain.**

From Iqra Chaudhary.

Submission date: 29-Aug-2021 04:14PM (UTC+0500)

Submission ID: 1637593152

File name: Modefied-_Twitter_Bot_Detection_-_2.docx (168.58K)

Word count: 8031

Character count: 41452

Similarity Index 18%

Similarity by Source

Internet Source: 9%

Student Papers: 15%

Publications: 4%

SOURCES:

1. 8% match form student papers

Submitted to Higher Education Commission Pakistan

2. 1% match from internet source

en.wikipedia.org

3. 1% match from student papers

Submitted to Polytechnic of Zagreb

4. 1% match from internet source

scholar.smu.edu

5. 1% match from internet source

www.personal.psu.edu

6. 1% match from internet source

www.sas.com

7. <1% from student paper

Submitted to Indian Institute of Technology, Madras

8. <1% from student papers

Submitted to Southern New Hampshire University - Continuing Education

9. <1% from student papers

Submitted to University of Bedfordshire Student Paper

10. <1% from student papers

Submitted to University of Bedfordshire Student Paper

11. <1% from student papers

Submitted to University of Keele Student Paper

12. <1% from student papers

Submitted to University of Sydney

13. <1% from Internet Source

Ceur-ws.org

14. <1% from student papers

Submitted to Birkbeck College

15. <1% from student papers

www.aclweb.org Internet

16. <1% from student papers

Submitted to Coventry University

17. <1% from Internet Source

tech.hindustantimes.com

18. <1% from Internet Source

Hidaytrahman.medium.com Internet

19. <1% from Internet Source

citeseerx.ist.psu.edu

20. <1% from Internet Source

Data-flair.training

21. <1% from Publications

Jarai Carter. "Who's Trending in Agriculture? A Look at Social Media" , Natural Sciences Education, 2013

22. <1% from Internet Source

phys.org

23. <1% from Internet Source

www.geeksforgeeks.org

24. <1% from student papers

Submitted to Siddaganga Institute of Technology

25. <1% from Internet Source

docplayer.net

26. <1% from student papers

Submitted to Campbellsville University

27. <1% from Publications

Winnie Main, Narendra Shekokhar. "Twitterati Identification System" , Procedia Computer Science, 2015

28. <1% from Internet Source

www.hpl.hp.com

29. <1% from Internet Source

www.conceptdraw.com

30. <1% from Internet Source

www.studyeducation.org

31. <1% from Publication

Dipta Voumick, Prince Deb, Sourav Sutradhar, Mohammad Monirujjaman Khan. "Development of Online Based Smart House Renting Web Application" , Journal of Software Engineering and Applications, 202

32. <1% from Publication

Hrushikesh Shukla, Nakshatra Jagtap, Balaji Patil. "Enhanced Twitter bot detection using ensemble machine learning" , 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021

33. <1% from Publication

Hrushikesh Shukla, Nakshatra Jagtap, Balaji Patil. "Enhanced Twitter bot detection using ensemble machine learning", 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021

34. <1% from Internet Sources

docs.microsoft.com

35. <1% from Internet Sources

roi4cio.com

36. <1% from Internet Sources

trap.ncirl.ie

ACKNOWLEDGMENT

All the greatness is for **ALMIGHTY ALLAH** who gave us potential, capability, and influence to complete the project. We are highly thankful to the **Faculty of Computer Science** for continuous supervision, compensation, and auspices. It gives us a huge gratification to receive the supervision, intention, intimation, judgment, and inspiration provided by the project supervisor **Ms. Iqra Chaudhary**. We are much obliged to him for careful consideration, gratitude, and supervision throughout the project. His moderate way was also offered to us all through the project. We are grateful to our parents who always motivated us and remember us in their special prayers for achieving goals in our lives and all other friends who boosted up our motivations. And special Thanks to **ALMIGHTY ALLAH** once again.

TABLE OF CONTENTS

Chapter **Page**

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Machine learning allows the program to learn on its own without human interaction. It's a branch of artificial intelligence based on the idea that computers can learn from data, identify patterns, and make decisions with minimal human intervention.

Most people use Twitter to think and discuss emotions, news, memes and daily activities. Till 2017 Twitter allowed the users to use 140 characters but later it is changed to 280 words now. Supervised machine learning, is an artificial intelligence and machine learning subcategory. The use of labeled datasets to train algorithms that identify data or predict outcomes defines it. A bot is a piece of software that finishes robotized tasks over the Internet. On social media, the pervasiveness of bots is universal. Twitter bot Detection is a web-based application that works on machine learning. It is developed to detect Twitter bots.

The application enables the user to detect bot accounts from their following. The system uses a random forest algorithm to detect the bots. The existing systems only return the first 20 users with a rating of 1-5 and it doesn’t allow them to unfollow or block at once. The reason behind this application is to detect and remove all the spam accounts from one following. Bots usually mess up timelines by retweeting and sharing malicious links. It also allows the user to unfollow or block them also.

Resurging interest in machine learning is thanks to an equivalent factors that have created process} and theorem analysis a lot of in style than ever. Things like growing volumes and styles of on the market knowledge, machine processing that's cheaper and more powerful, and reasonable data storage. All of those things mean it is attainable to quickly and mechanically turn out models which will analyze bigger, more advanced data and deliver faster, more correct results even on a really giant scale. And by building precise models, a company contains a higher probability of characteristic profitable opportunities or avoiding unknown risks [1].

One of the major issues with social media platforms such as Twitter is that multiple social bots or Sybil accounts are controlled by automated agents and are commonly used for malicious activities. Here a large number of visitors are directed to a particular website that can be considered as spam, influence the community on a particular topic, disseminate false information, and encourage people to join illegal organizations. Includes recruiting, maneuvering people in stock market activities and blackmailing people. Through the power to disseminate personal information. This account. Therefore, the detection of social robots is very important to protect people from these harmful effects. In this study, social bot detection is treated as a map classification problem on Twitter and use machine learning algorithms after preprocessing and feature extraction of a lot of data. Tweets are analyzed,

posted by Twitter user accounts, personal profile information and time behavior, and extracted a wealth of functions.

A central goal of students is to summarize from their experience. In this case, generalization is the ability of the learning machine to accurately process new and unseen examples/tasks after experiencing a set of learning data. The training example comes from a generally unknown probability distribution (considered as a representative of the occurrence space), and the student must build a general model in this space so that he can produce sufficiently accurate predictions in new situations [2].

The computational analysis of machine learning algorithms and their performance is a branch of theoretical calculations, called computational learning theory. Because the training set is limited and the future is uncertain, learning theory usually cannot guarantee the performance of the algorithm. On the contrary, the performance probability limit is very common. Bias variance decomposition is a method to quantify the generalization error.

For best performance in the context of generalization, the complexity of the assumption must match the complexity of the database function. If neither function is assumed to be complex, the model does not fit the data. If the complexity of the model increases accordingly, the training error will decrease. But if the assumptions are too complex, the model is easy to over fit, and the generalization will become worse.

In addition to performance limitations, learning theorists also study the complexity of time and the feasibility of learning. In computational learning theory, a calculation is considered feasible if it can be performed in polynomial time. There are two types of time complexity results. The positive results indicate that certain types of functions can be learned in polynomial time. Negative results indicate that certain classes cannot be learned in polynomial time.

1.1. Project Domain

The project is based on detecting Twitter bots using machine learning algorithms. This project is developed to check the credibility of the followers one has on his Twitter account. The project is developed using Python and its framework Flask on the backend and HTML, CSS, JavaScript, and Ajax on the frontend. The developed system provides three options to anyone who is using it. The first one is to check an individual account as a bot or human. The second one is to check the total number of bots and humans. The third and the last one is to either unfollow them or block them. This is the main purpose of proposed system. Proposed system will be a web app, which allows user to perform all above tasks.

1.2. Problem Identification

Many Twitter bots are specialized for various purposes. There are many reasons to create Twitter bots but it entirely depends on what particular purpose they are being created for. Different Twitter bots are designed for different purposes.

It's a problem because you can manipulate your bot account information to disseminate false information and promote unconfirmed information. This can adversely affect public opinion on a variety of topics, such as product sales and political campaigns. Detecting bot activity is complicated because many bots try to avoid detection. Proposed system presents a new complex machine learning algorithm that takes advantage of various features including username length, reissue ratio, time pattern, emotional expression, follower friend ratio, and message variability for bot detection [3].

The popularity of Twitter has led this website to become ideal for bots and spammers. Creating a bot is very easy you simply need to approve your Twitter API approved. Famous Brands and Influencers also buy them online very easily. Some are used to increase the number of followers, some are for spamming some are for political campaigns. You don't know the next person you are going to follow is probably a bot or a person you know for years who is just liking your tweets is also a bot. They also spam one's timeline terribly by sharing the same links again and again.

1.2.1. Proposed Solution

Twitter bots always try to hide their characteristics continuously so detecting them is quite a complex process. To determine whether an account is a bot or not, checked over 18 different distinguishing attributes per case, including the amount of randomness in the Twitter handle, whether the account is verified, the number of people that account is following to followers, and the account's description. Dataset consists of 37k accounts with 18 attributes per account. The dataset is divided into training data set and test data set, the ratio is 0.8 and 0.2 respectively. Each account consists of attributes created_at, default_profile, default_profile_image, description,favourites_count,followers_count, friends_count,geo_enabled,id,lang,location,profile_background_image_url , profile_image_url,screen_name,statuses_count,verified,average_tweets_pe

r_day, account_age_days and account_type. Most Twitter bots are programmed in a way that they follow people at a very fast rate. During following, they might not get a follow back from a user which makes their following to follow ratio very high. This is one of the key signs along with many other ones like tweeting at certain topics repeatedly like favoring and retweeting other's tweets at a very fast rate.

1.2.2. Objectives

Every project which is created has some kind of aim and goal. Objectives of this project are:

i. To design a system that helps the user to detect and remove spam.
ii. To develop a system which let user detect, unfollow and block bots.
iii. To help the advertisement companies to check the authenticity of the influencer's followers.
iv. To allow users to unfollow/block them directly instead of going to their timeline manually.

1.3. Scope of the Project

The project is based on Twitter which allows users to detect and block accounts that aren't operated by humans. Even if someone doesn't have a Twitter account he can still check the credibility of any account. The user doesn't need to sign in to his Twitter account in order to check the result. It will allow the public/companies/brands to check the authenticity of any user. Users can miss important tweets when bots are continually tweeting and retweeting links in order to divert traffic to those sites. It will help to remove spammy tweets from one's timeline.

A bot is a set of disparate account types that automatically post Tweets. Twitter's popularity as a means of public discussion has led to a situation where Twitter is becoming an ideal target for spammers and automated programs. About 10% of all users are bots and it is estimated that this account generates about 20-25% of all tweets posted. For research purposes, bots pose serious problems because they degrade the accuracy of data and can dramatically skew the results of analytics using social media data [2].

Bots usually spam the timelines of user by spreading fake and spammy news articles from third party websites.

Every user wants to have a clean and spam free timeline which shows tweets by other authentic users not from some spammy accounts.

1.4. Effectiveness of the System

The project will help different areas of development related to the need to understand user stories. The suggestion system provides an efficient method of discovering the user's needs based on the phrases that the user really wants. The proposed system is completely free, so people can use it without paying any fees.

1.5. Resource Requirements

This section describes the resources used to develop and implement this project. These resources may include the hardware and software requirements used in your project. Since no hardware is used, the requirements are based solely on the software.

1.5.1. Software Requirement

Below table 1.1 demonstrates the system requirements:

Table1.1: Software Requirement

Serial no	Tool	Purpose
1	Anaconda	Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. Anaconda is basically free and open-source with over 1500 + python and R data science packages. It makes the package management of any project very simpler. One main advantage is to collect data from different sources using ml and AI.
2	Jupyter Notebook	The Jupyter Notebook allow us to code live and share live code with others. Jupyter notebook is used to run cell by cell which means splitting all the logical steps helps us to interact with data in a great way. It can delete any small cell and it won't affect our code. Any change on

		the cell can be reverted without thinking too much about the consequences.
3	Python	Python is a programming language that is mostly used in this project. The use of python makes the project much simpler because of its easy syntax because it's close to natural language. Python code can be executed much faster than other programming languages.
4	Flask API	Flask is a micro framework which is used to develop web applications. Flask provides us with useful tools, libraries that allow you to build a web application. The good thing is it doesn't force its dependencies on how you should develop your project like Django.
5	JavaScript	JavaScript is a programming language used both on the client-side and server-side. It is usually used for validation purposes. It also helps to execute very complex tasks and helps them in interacting the websites with the visitors to make web pages interactive.
6	Ajax	Ajax is a set of web development techniques using many web technologies on the client-side to create asynchronous web applications. Ajax issued for making callbacks, which means it gets or saves data without posting the entire page back to the server unlike other languages. It also allows users to make asynchronous calls to a web server. In this way, users don't have to wait for all data to arrive from the server. The applications where Ajax is used are faster and user-friendly. The Ajax Control Toolkit is a suite of controls created by Microsoft that is integrated into Visual Studio and can be dragged and dropped onto web forms just like html and server controls.

7	Visual studio	The Visual Studio IDE (integrated development environment) is a software program which is used by software developer to edit, debug and build code. It is developed by Microsoft and is very fast, cross-platform, and great for working with the cloud as well as other applications

1.5.2. List of Libraries

Below table 1.2 shows libraries used in the project:

Table1.2: List of Libraries

Serial no	Libraries	Purpose
1	Pandas	Pandas is a famous data science library in python. It is mainly used to deal with comma-separated values, JSON, SQL, and Microsoft Excel files. The best thing about pandas is it allows us to manipulate, reshape, merge, select, delete, update, clean, preprocess data in your own way very easily, and while doing it manually would be one hell of a difficult thing.
2	Matplotlib	Matplotlib is a data visualization and graphical library in python and it helps us to create graphs, plots, and different shapes. It is used to draw quick inline plots mostly when using Jupiter notebook.
3	Seaborn	Seaborn is also used to visualize data only it used fascinating themes and patterns.
4	Scikit-learn	Scikit-learn is one of the most useful libraries for machine learning in Python because of its abundance of features. It contains all the major algorithms that are used in this projects.
5	AnyChart JS	AnyChart's product family consists of a collection of adaptable JavaScript (HTML5) libraries for all of your data visualization needs.

		Beautiful charts and dashboards may help you stand out with your goods, applications, and web sites.
6	Bootstrap	Bootstrap is a free framework that is used to create beautiful fronts with very little code. It includes HTML, CSS-based designs like forms, tables, buttons, navigations, Image carousels, tables, and many others. It gives us the ability to create beautiful designs without wasting time.
7	Twint	Twint is a web scraping library that is used for scrapping tweets from Twitter on any topic, or hashtags, or languages. Twint makes it easier to scrap data without much of a limit, unlike Tweepy.
8	Tweepy	Tweepy is a python library that is used to access Twitter API. It helps us in automating Twitter accounts. You can also get data from Twitter using Tweepy.

1.5.3. Data Requirements

Data is an important aspect of machine learning projects as it mainly revolves around data. Relevant and accurate data can help in the correct result. If unnecessary data is used in this project, then it will lead to irrelevant problems. The data collected is from Kaggle's “Twitter bot detection dataset”. The dataset contains 37,000 different accounts from all over the world and each account has different attributes. Out of these attributes, some of them will be dropped while preprocessing.

1.5.4. Hardware Requirements

Hardware is necessary to run programs efficiently and helps us to run the different applications properly. So the hardware is of good and high quality the applications will run fast and if not the project will be slow. Any application needs to meet the minimum requirement to make the run on it. Although recommended requirement should follow to run the software

without any lagging. The hardware project based on the developed project are shown in below Table 1.3.

Table1.3: Hardware Requirements

Serial number	Requirements
1	Computer or Laptop
2	Intel Core i3 or later processor
3	2 GB RAM or more.
4	Any web browser but recommended Edge MS or Chrome

1.6. Report Arrangement

This report is divided into six chapters. Each chapter contains specific details about the system. The first chapter introduces the project, scope and requirements. Then the second chapter focuses on the background, the existing system and the limitations of the system. The third chapter deals with system requirements and specifications. In Chapter 4, systems modeling and designs have been discussed. Chapter 5 contains details of verification and testing of the system. Finally, Chapter 6 contains the conclusion of the final work.

Chapter 2

BACKGROUND AND EXISTING SYSTEMS

The main focus of this project is the needs of users. Without understanding the problem, it is quite difficult to create a system. Need to use a quick and easy way to build the system.

Section 2.1 of this chapter gives us ideas for related literature and where the ideas for developing this project came from. Section 2.2 of this chapter introduces us to the existing systems and work related to the project, and discusses the limitations resolved in the project. Section 2.3 discusses and explains the problems found in existing systems/works. These identified issues (limitations) may be related to technical, functional requirements or non-functional requirements. At the end of this paragraph, how to improve the existing system will be discussed. Section 2.4 defines the limitations of this proposed solution. This section contains the functions that will be implemented in the proposed system (refer to section 2.3) and discusses all the functions that will not be implemented in the system for some reason.

2.1. Related Literature Review

Research papers and articles associated for proposed project are explained below:

2.1.1. Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots

This paper is all about the detection of Twitter bots and their presence online. Bots account considered problematic because the reason behind them is to increase artificial following, spamming timelines, spreading misinformation, and sharing fake news which can affect the public opinion on some topics.

Bots actively try to avoid detection so it's not easy to detect them. In this paper, a complex novel machine learning algorithm using different features including names, temporal methods, expressions using tweets, follow and following ratio. This technique is effective with a 2.25% classification rate. To evaluate the method is labeled manually 500 Twitter user accounts for two classes: spam and not spam. Each user account is manually evaluated by reading the 20 most recent tweets posted by the user and checking the friends and followers of the user. The result shows that there is about 1% of spam account in the data set. The study shows that there are probably 3% of spam on Twitter [4].

These accounts spread fake news, web links and other kind of spam links to interact users to their website or any other platform.

2.1.2. Machine Learning and Bots Detection on Twitter

This paper, it is described how to eradicate spam from micro blogging site Twitter. @ Of the recent tweets have been extracted of each user and analyze. This tweet data has been used along with the other attributes of the Twitter account. The chances of the error have been increased due to the evaluation experiments.

The goal of this paper is to identify spam behavior by applying different methods.

2.1.3. Enhanced Twitter Bot Detection Using Ensemble Machine Learning

This paper is about how they can utilized metadata from different profiles to get a unique features selection which was later explored to train a robot classifier. The difficulty of detecting Twitter bots is addressed in this work. They examine a set of 8385 Twitter accounts and their tweets, which includes both people and various types of bots. This information is used to train machine learning classifiers that can tell the difference between authentic and fake accounts. They look for traits that are simple to extract and produce good outcomes. They find traits that are simple to extract and produce good results. You compare the performance of multiple feature groups based on account-specific, tweet-specific, and behavioral-specific features to existing state-of-the-art bot detection approaches.

The Twitter client was a mechanized bot account. Each time, typically around 6,000 tweets are tweeted on Twitter. Utilizing social media on a vast scale is in the present age both in benefits and in distress. Adopting Twitter accounts to spread fake data has been on the rise in recent years. Counterfeit accounts are a significant source of misinformation through web-based network media. In this paper, classification computation in machine learning is used to detect fake accounts. How to find fake accounts, in most cases, relies on features such as screen names, location guidance, and verification [5]. These accounts tweet at very high number as compared to normal account as well and the reason is because they are automated. That’s why they start spamming user’s timelines. And which is totally weird for user, as everyone wants their timeline clean and just want to show their tweets .

2.2. Related Systems/Applications

Systems and applications associated to proposed project are explained below:

2.2.1. Botometer

Botometer is a web application that looks at the feature of a user which includes friends, social network structure, account activity, language, and sentiment of accounts. Later then it gives a (0-5) score. It used several other scores that provide a likelihood that an account is a bot or not.

Most of the target users are general public. It is free but you need to buy premium version to get features. It used machine learning and artificial intelligence and is developed by professionals at Indiana University.

2.2.2. Analytics for Twitter (Android)

This app is based on how inactive the accounts are by just extracting timeline of all the accounts without the use of machine learning.

2.3. Selected Boundary for Proposed Solution

The boundaries of this system are all public accounts. The system will perform well when the account is public but if the user has made his account private then one can't predict it. The reason behind this is Twitter doesn't allow anyone to extract the data which is private. The application is not suitable for very large accounts as standard API won't allow many API calls and the system gets the per window limit. These limitations are both for POST and GET requests. It only allows us to extract 3200 tweets per user and 7-day history as well, which is very low.

Chapter 3

SYSTEM REQUIREMENTS AND SPECIFICATIONS

This chapter discusses the system requirements and project specifications. There is a discussion about the different content of the functional requirements system module, as well as the non-functional requirements of the proposed application / system. Each title is defined in detail for a better understanding.

The first part discusses the system specifications. In the second part the modules that make up the system are discussed. In the third part, the functional and non-functional requirements of the planning system are discussed.

3.1. Specifications of System

Proposed project contains various modules and technologies. At first username of the Twitter account is provided to the system. The system then uses Twitter API to get the followers of that user. After that, all the different and important attributes will be extracted through API. Random forest algorithm has been used to get the output result. The result will send back to the frontend using Flask and will be displayed there.

3.2. System Modules

System modules are different components that are combined to form a complete system. Each function is completed by different modules. This project consists of mainly three-module, number one is to preprocess and clean the dataset and train it, the second one is to extract data from Twitter and transform it to a format that proposed model can easily predict and the third one is to create a web application to display the output.

3.2.1. Extraction of followers

In this step, system will take input from the user which will be the Twitter handle. The next step will be the use of Twitter API with the help of twitter's library Tweepy and Twint which is an advance twitter scrapping library to get data of that account including the followers, following, favorite count, how often the user tweet, when the account was created, average tweets per day, whether the account has default profile picture or not, description, are the verified or not. Our machine learning will train on the basis of these features. Our dataset also contains these features. They differentiate and an account from bot and human.

3.2.2. Training the model

The first step in training the model is preprocessing. The data got from Kaggle has a lot of missing values which were figured out first. Later different algorithms have been used to get better accuracy i.e. Random forest, decision Tree Classifier.

3.2.3. Predicting the output

The data that the system got from Twitter is later converted into a format in which model can easily understand and predict. Pickle module is used to convert the already build model into a file.

3.2.4. Building the web application

In last step flask is used to build the web application also it manages the HTTP requests and rendering template.

3.3. Non Functional Requirements (NFR)

Requirements not included in the system and software functions are basically non-functional requirements. In fact, it tells us the quality of a system. Non-functional requirements are as important as functional ones. Non-functional requirements play an important and effective role in the development of the system. Non-functional requirements refer to what the system proposes and the qualities it contains. Users must be satisfied with the system when they use it, which can be achieved through functional and non-functional requirements. These requirements are divided into the following categories.

Consumers did not clearly state non-functional requirements for the duration of the previous phase. The developer carefully considered its existence and the importance of its inclusion in programming applications on the basis of many segments. Examples of these requirements include system adaptability, efficiency, scalability, etc.

3.3.1. Security of System

The system needs to be trusted for security purposes, and this project can be trusted because the administrator has all the primary privileges. In particular, some key features for administrators include uploading images and videos.

These features can only be viewed or accessed by an administrator. Unauthorized users cannot use these features or control them to use the system without registration and limited privileges. Since admin also has the authority to manage user permissions, user permissions can also be determined by admin permissions.

3.3.2. Reliability of System

Consistency, as well as reliability, is a disaster-free procedure for a planned system for specific patience, surrounded by the atmosphere measured during a specific period. If the structure recommends the structure or the structure can be used stably for a long time, we applaud the fact that the disaster rate is low.

3.3.3. System Usability

Not only is the structure / product boundary appropriate in System Usability, but also insert data to create the appropriate interface, never see the interface, and all consumers through the interface through a naive plan it is deliberate so that it can be easily interconnected.

3.3.4. Availability of System

The availability of a system is contracted through the structure or functionality of an application to ensure that the operation of the system or application runs throughout the day without errors or bugs that could cause the system or application to become inaccessible. The system or application is reachable and up and running around the clock.

3.3.5. Testability of System

The feasibility of the system runs on the system or application, as well as many steps of checking. Unit tests are implemented in each module in the structure. Black box testing is performed to test the entire system. A system as a whole, accumulates them and tests the skilled structure or application.

3.3.6. Operability of System

The mechanism of system operation and the ability to pattern apps are fine in partial operational situations, with previously segmented requirements

and conditions. Once confirmed or confirmed, it completes the complete method or application steps. A system or app is to work comprehensively when it is practical in prototyping to create a graphical user interface (GUI).

3.3.7. Resource's Requirement

With Resource's Requirement, there are no other resources needed to run the system or app. This is easy, as well as a desktop it'll that welcomes consumers. Users can simply enter the app's input data into the text area and get the appropriate results preferred for output through the desktop app or system.

3.3.8. Required Interface

Interface requirements, an interface that welcomes consumers of this application is planned. There is one button for creating a GUI with options to upload a text file, have an area where you can enter written text, or access the border.

3.3.9. Performance Requirement

It has served the purpose of welcoming many consumers to support consumers when their performance requirements create a graphical user interface for a desktop application or GUI means of a system. Various modules are created in an application or system through various methods.

3.3.10. Accuracy

There must be one of the people who are not common in all, d. H. Accuracy, but the accuracy is not automatically to the system. The accuracy is whether users can trust the results that receive users from the system or not. The accuracy achieved by the continued system testing process. The accuracy in the work system is very important for the system to be successful in the future. The accuracy is achieved in the development project due to long hours of data or image training.

3.3.11. System Speed

System-wide functionality depends on system speed. By using a set of techniques, Ajax, this system works at great speed. With Ajax, you can make

this system very fast by rendering data to a page without retrieving the entire page multiple times from the backend.

3.3.12. Efficiency

The first thing that comes to mind for users is efficiency. Most users who use the application are hoping that the app will respond quickly because the system is not responding late. The main reason behind efficiency depends on the system the application is running on. Therefore, the functionality of the system is as important as its precision. If the system is efficient, the attack of the application increases. If your system is efficient, your application will be more efficient. Applications based on training models require fast, high-performance systems to operate efficiently.

Chapter 4

SYSTEM MODELING AND DESIGN

This chapter describes system designs and diagrams. The system design and analysis are described in detail in Section 4.1, and Section 4.2 describes the Use-Case Diagram. The next section, Section 4.3 describes the Activity Diagram. In the 4.4 section describes System Sequence diagrams. In the last section 4.5 the Data Flow diagram is described.

4.1. System Design and Analysis

This section described how to design your system. The purpose of system analysis and design is to use various diagrams such as use case diagram, activity diagram, data flow diagram, system sequence diagram, sequence diagram, design class diagram, architecture diagram, interface design, component level, etc. System Diagrams are more than process flow charts. They include feedback loops and other factors that influence how decisions are made, including attitudes, perceptions, and behaviors. If you are familiar with the terms "vicious circle", "downward spiral", "the law of unintended consequences", or "the cure is worse than the disease" you are familiar with some of the basic concepts of System Dynamics. System Diagrams provide a common language to help organizations think about these complex issues. The user is notified of the system flow. If the developer adapts the design easily then it would be easy to create the environment and coding it later.

4.2. Use-Case Diagrams

Use case diagrams are used to display system request settings. It also shows the interactions that occur between the user and the system when the program runs. Each use case must provide amazing and valuable results with the actors of the system or various stakeholders. A use case diagram will summarize the main points of your system's users (also called actors) and their interactions with the system. Machine learning applications typically interact less with users than management systems." Twitter bot detection" applications using many of the features when interacting with users. In this diagram the actor is first checking whether the account is bot or human, can unfollow, and also can block these accounts. These types of interactions usually occurs when the program is running. Following figure shows how the user interact with the proposed solution and how solution satisfies the user to user’s desired output. All the interaction between user and the system is shown in below figure. User inputs some data and the machine learning model predicts the output result. The interaction with the system is shown below in the figure 4.1.

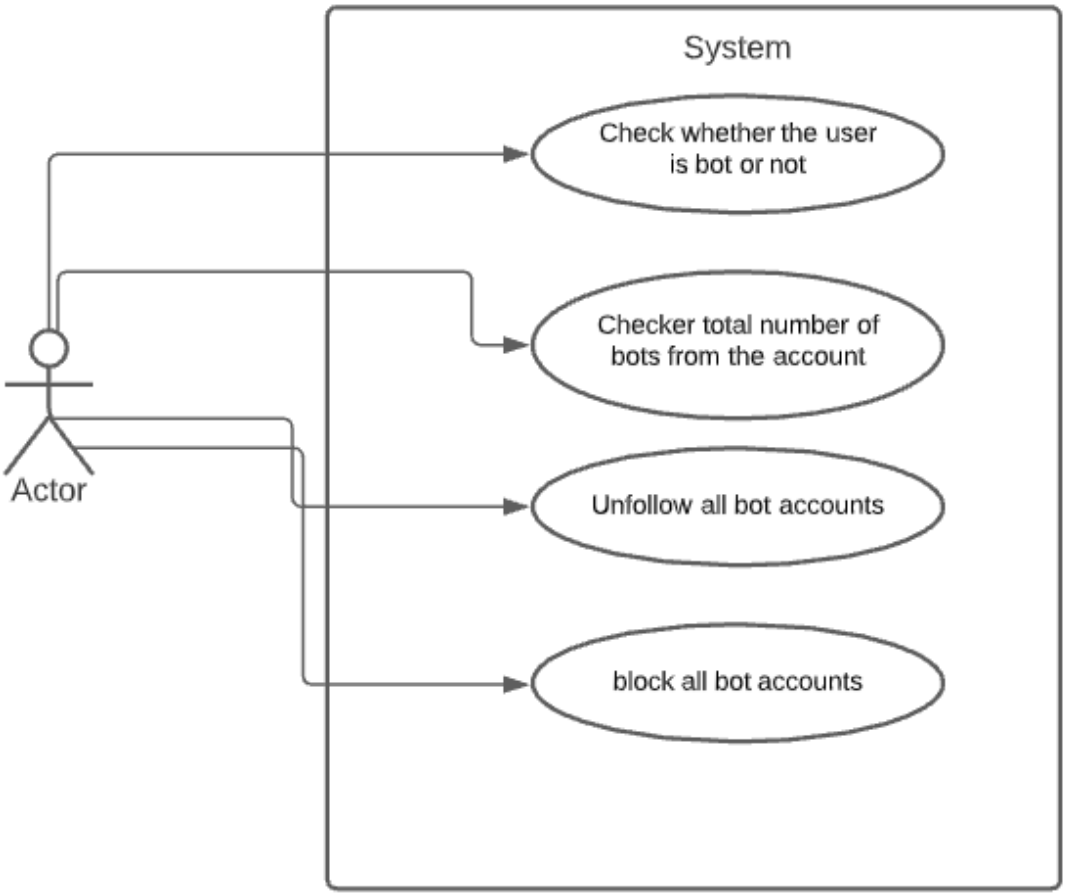

Figure 4.1: Use-Case Diagram

4.3. Activity-Diagram

Activity diagrams are important charts that explain the dynamic functioning of structures in UML (Unified Modeling Language). We use Activity Diagrams for example the flow of control during a system and talk to the steps concerned within the execution of a use case. we tend to model consecutive and cooccurring activities victimization activity diagrams. So, we tend to essentially depict workflows visually victimization Associate in Nursing activity diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. Activity diagrams show the flow of data from one task to another. Sequential and concurrent activities are being shown using this diagram. An activity diagram and the flow chart diagram are quite similar. In activity diagram, the input is taken from the user, and then it is used to get data which later used for prediction. Below figure 4.2 is activity diagram for this project:

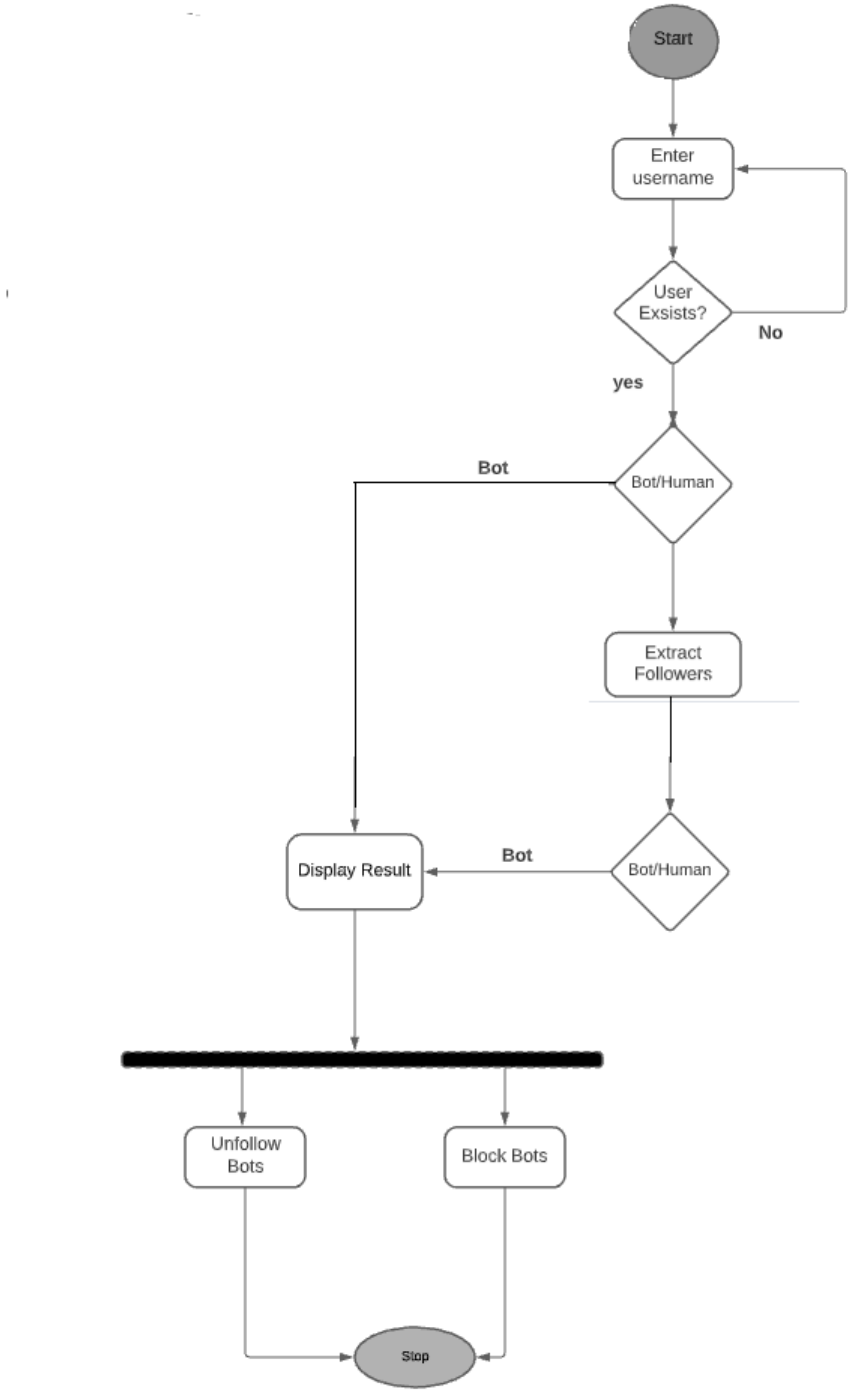

Figure 4.2: Activity-Diagram

4.4. System Sequence Diagram

Sequence diagrams focus on time and show how and when messages are sent vertically. A system sequence diagram ought to be finished the most success state

of affairs of the use case, and frequent or advanced various situations. Indicates the sequence and time of interaction. The picture given shows how the user enters a Twitter account input username. Then Twitter API is getting that particular follower's attribute back to the user. Here are two conditions if and else, if the user doesn't exist it will return an error message and if the user exists then it will return all the data. If the account is private then still won't get any data. After then that individual user's data is converted into a format that is compatible with the model. The model will then predict the probability of that individual account. Next step you can see the user is extracting all the followers and their data which is again then sent to predict the total number of bots and humans. In the last step, the user has the option to either unfollow them or block them if he wants.

The interaction between system and other systems, or between subsystems is the system or one system with another. Below figure 4.3 shows the use case or an operation of a system.

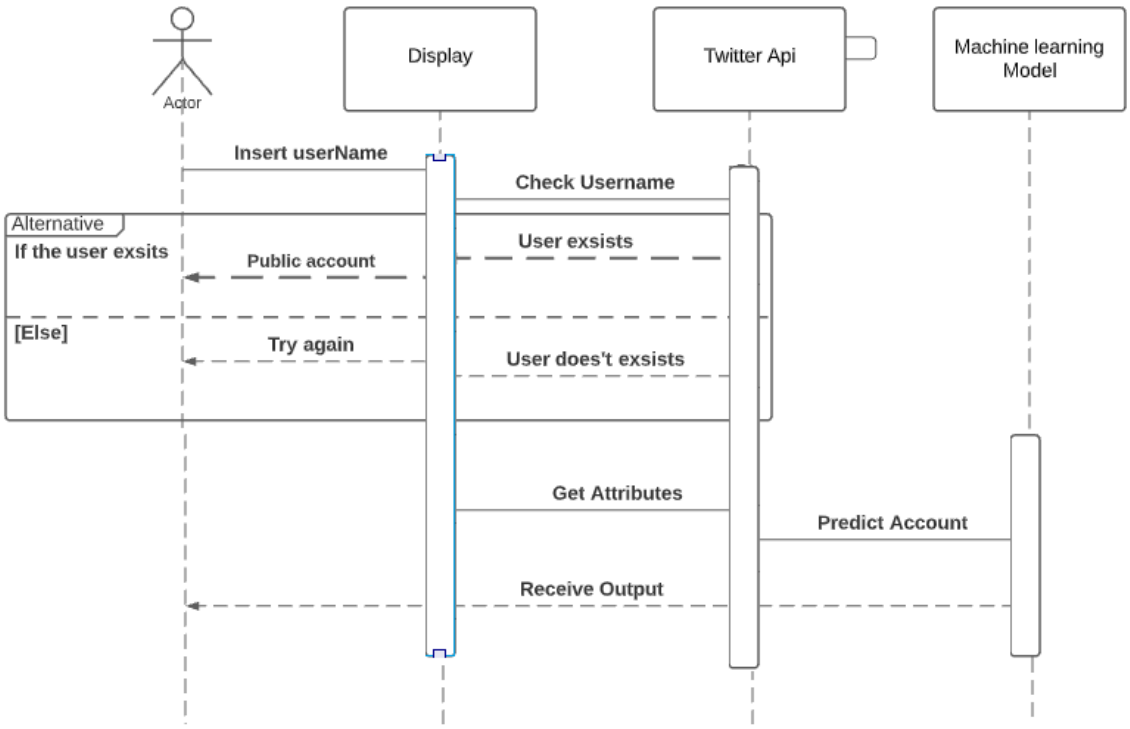

Figure 4.3: System Sequence Diagram

4.5. Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a processor or a system. The DFD also provides the information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no

decision rules and no loops. Data flow diagrams are used to represent the system at different levels. Higher levels of DFDS have more information and elements about the system as compared to the lower. DFDS start from 0-level-DFD, 1-level-DFD, and 2-level-DFD.

4.5.1. 0-Level-DFD

DFD level 0 is also known as Context Diagram. It is a very basic display of the system. It shows a system as a single high-level process. If someone is not a developer he can easily understand it so it mostly uses to facilitate a wide audience. In 0-Level-DFD diagram, the user is simply asking about the bots and the result is returning after processing from the model. Below diagram 4.5.1 shows DFD level 0.

Figure 4.5.1: 0-Level-DFD

4.5.2. 1-level-DFD

In 1-level-DFD the 0-level-DFD is decomposed into multiple processes. The main functions are breakdown into multiple other processing detail. One can also say the level-1-DFD diagram is a more exploded version of the context diagram. the context diagram is decomposed into multiple bubbles/processes. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses In 0-Level-DFD diagram, the 0-Level-DFD is explained in detail that how the user's input requirements are sent to the model and the detailed process from the model are returned back. Below is figure 4.5.2 is 1-Level-DFD:

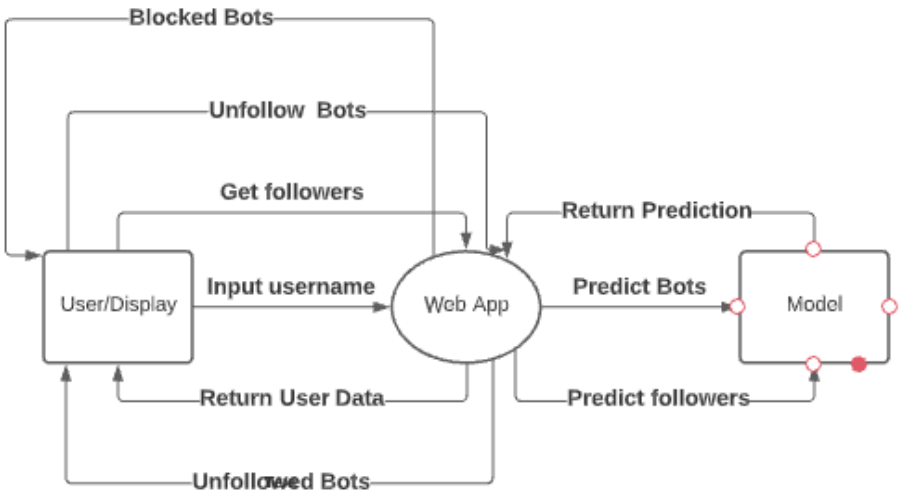

Figure 4.5.2: 1-Level-DFD

Chapter 5

TESTING AND VALIDATION

In this chapter, it'll be discussed about testing and the second most vital thing which is validation. Here many diverse methods of testing are used to ensure that there is no error, glitch, or bug. Basically, testing is an important part of software development. No bugs or errors in the application/framework as a result of testing and evaluation. The software is unacceptable to the customer, expecting not only validation but also testing procedures to be accepted and executed. Several tests are tested during the period to determine whether the application/system coder encoding system meets the requirements.

5.1. System Testing

A system is a form of testing that covers an entire set of fully integrated software products. The main purpose of unit tests is to identify and measure end-to-end requirements. Finally, software interfaces with other applications or systems. System testing is a way of demonstrating a system's good operational performance. The system should be free of any kind of errors or bugs. As main goal of this project is to deliver a very reliable and accurate system. There are many methods that can be used to test this system but all methods cannot be used for all systems. Every system needs to be tested by different methods but most of the methods which are used are based on two major categories i.e. Manual Testing and automatic testing. In order to test your system manually, the tester doesn't need to have high-level skills or advance knowledge of the system and how it is working. System testing is performed in the context of a System Requirement Specification (SRS) and/or a Functional Requirement Specifications (FRS). It is the final test to verify that the product to be delivered meets the specifications mentioned in the requirement document. Testing is a great way of helping a developer correct the uncertainties in their system. Automatic testing needs a highly skilled professional who knows everything about the project and system. It is designed to include representatives of boundary values. It is a type of testing to verify that a product performs and functions correctly according to user specifications. Automatic testing is usually preferable over manual testing and is considered more accurate. That's why testing and validation are very important for proposed system because it won't be a good sign if it meets any kind of failure after deployment [6]. Testing is used to check the performance of app under high intensity conditions. There are different types of testing that are featured below in this section.

5.2. Machine Learning

This application has been developed around machine learning, the process of learning yourself on a computer. There are several steps in this process. The first step is to split the data into training and testing.

Machine learning is divided into three main types that are most popularly used. These three types are reinforcement learning and reinforcement learning by teachers. What is used in this project is supervised learning.

5.3. Supervised Learning

Supervised learning means the use of labeled data in training the model. It can either be textual data, images, videos, or numerical data. The data is passed to different algorithms and then train in the model. Testing data is not labeled which is given to the model to predict the output. The algorithm can be used on the basis of the below categorization:

5.3.1. Classification

Classification is the process of combining data with other entities within the same group according to some general similarity. These similarities are classified as distinct from the criteria on which they are based. The classification of developed systems is based on robots and humans.

5.3.2. Dataset

Like all other machine learning systems, testing is included in map learning. The test is completely data set dependent. The data under test can be divided into three main types. These categories are described in Table 5.1:

Table 5.1: Dataset

Serial no	**System**	**Working**
1	Training data	The idea of using training data in machine learning is a very common and simple concept, but it is also base on the way these technologies are working. The training data is the initial set of data that helps in building neural networks or training the model to produce very sophisticated 3d results. It also follows by two more types

		of data, The first one is testing data and the second one is validation data. Training data in this project is labeled. Now, there are target values of bots and humans in this dataset.
2	Testing data	The test data is a group of results that are used to check the performance of the system. IT is very important that no observation from testing data should be part of training data. Suppose if a program memorizes the testing data by memorizing the output then that model can predict the output very accurately but it will fail when new test data is given to the model which has not already been in the data set. It's called over fitting of the mode in machine learning terminologies. Regularization is must in the models to reduce over fitting.
3	Validation data	Validation data is data that is held back from training the model so that it can be used to estimate the model which tuning the model's hyper parameters. The low validation loss means more chances of model accuracy.

5.4. Testing Techniques

The main purpose of testing is to identify software errors and bugs, identify defects, and allow them to be modified. Testing cannot enable the system to function normally under all conditions. Identify only those that are not functioning properly under certain conditions. The unit test and black box test are used to test the system and validate this software / application [7].

5.4.1. Unit Testing

In this stage of testing the system has been tested one by one as a separate segment by the tester. Unit testing is usually used to check the accuracy of each module in the system. If a bug is found then the developer is informed and it is then removed. AS if the bug is not removed in the initial phase it is very difficult to remove them later. Each system in this system has been checked before the completion.

5.4.2. Integration Testing

In integration testing, different units are first combined together and then tested. The main reason behind this type of testing is to check whether proposed system works fine when it communicates with other modules or not. It is mainly used to detect any faults or bugs between integrated Systems or not. System Integration testing may be performed after or in parallel with System Testing.

5.5. Non–Functional Requirements

The non-functional requirement (NFR) specifies a software system's features attribute. It tells us on which bases the software system is working like, Reliability, Usability, Security, Availability, Testability, Operability, Performance Requirement, and System Speed. Some of the most common nonfunctional requirements are mentioned below:

Table 5.2: NFR

Property	Measure
System Security	Proposed system is secure and it removes all the problems very quickly.
System Availability	All the features that are introduced in the system work fine 24/7 without any issues.
System Testability	Unit and integration testing has been performed on system to make it more accurate.
Resource's Requirement	System Requires PC/Laptop/Mobile and Web Browser.is quite responsive and reliable.

Chapter 6

CONCLUSION

This chapter is all about the results got in previous chapters and how these are determined. Moreover, the future work that can be done on this system to improve the performance and the accuracy of the system. Additional research and study can take this project to a better version of itself. The conclusion is as important as an introduction to any system. It is based on the overall working of the system in a summarized way. Making the conclusion in a very simple and understandable language is very important so that the people who aren't really expert in that field should also get the main purpose of the project.

6.1. Conclusion

The developed application “Twitter bot detection is web-based”. The dataset which is used to train the model is downloaded from Kaggle. It consists of 37,000 accounts with each account having different attributes. Later the system gets the input username in the form of text from the user. The attributes are both numerical and textual in nature. The data is then preprocessed to make it easier to get trained by the model. These attributes contain the total number of followers, the total number of followers, the total number of tweets, the overall number of favorites, Average of tweets, description, number of days past the account has been created how often the user is tweeting. The random forest algorithm and decision tree algorithm are used to train the model. But the accuracy of random first was better than decision tree so it’s selected for the final call.

The username which was submitted from the web page is sent back to the Flask backend to get all the major attributes from that person's account using Twitter API. The data from API is later cleaned and processed to make it easier to send it to the model for prediction of that account as a bot or not. The same process has been repeated again and again for all of his followers. After then the result has been sent to the web frontend to display the total number of bots and real accounts. The system lets the user unfollow them or block them later. Users don't need to sign in to check anyone's followers' stats but if the followers are needed to be blocked or unfollowed then the login part is mandatory. As twitter won't let anyone make changes to the account without login.

The system will be a lot different from existing systems by letting the user unfollow or block the bots as well as get the total number instead of just getting the result for

an individual account. One more existing system is just rating them from 1-5 where closer to 1 men's more chances of being a bot.

6.2. Limitations

No system is perfect. Every system has its own limitations for different reasons. Some are slow, some are not accurate some are not secure. There are two limitations to this system. The first one is as free version of Twitter Standard API is used so system is not getting rate limit very occasionally. It makes it harder for us to get more great features of accounts that can play a critical part to make this system much better. System is getting a rate limit very often whenever it makes API calls. Using the premium version of API is not possible right now because it's a very high price. The second one is if the person's account is private system can’t check anything about that account as Twitter won't let anyone access the data of a private account. So even checking the total number of bots of anyone the account who are private will automatically be skipped. That accounts won’t affect the accuracy much as private accounts are very rare because of the public environment of twitter.

6.3. Future Work

Future work is to get premium Twitter API which could potentially get a huge amount of data without any limit. After having that API you can make this machine learning model much more accurate by analyzing the retweets as well as the way bots have conversations with other users.

REFERENCES

[1] Mücahit Kantepe; Murat Can Ganiz (2017) .Preprocessing framework for Twitter bot detection.
[Online] Available : https://ieeexplore.ieee.org/abstract/document/8093483 [Accessed: 25 November ,2020]

[2] Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots by Efthimion, Phillip George; Payne, Scott; and Proferes, Nicholas (2018) "Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots," SMU Data Science Review: Vol. 1 : No. 2 , Article 5.
[Online] Available : https://scholar.smu.edu/datasciencereview/vol1/iss2/5/ [Accessed: 03 December ,2020]

[3] Towards a language independent Twitter bot detector by Jonas Lundberg1, Jonas Nordqvist2, and Mikko Laitinen3 1 Department of Computer Science, Linnaeus University, V¨axj¨o, Sweden 2 Department of Mathematics, Linnaeus University, V¨axj¨o, Sweden 3 School of Humanities, University of Eastern Finland, Joensuu, Finland
[Online] Availablehttp://ceur-ws.org/Vol-2364/28_paper.pdf [Accessed: 29 December ,2020]

[4] Machine Learning and Bots detection on Twitter by Norberto Almeida de Andrade (Universidade Anhembi Morumbi) Giuliano Carlo Rainatto, (Senac São Paulo), Fonttamara Lima (Centro Universitário das Faculdades Metrop), Genésio Renovato da Silva Neto (Universidade Nove de Julho).
[Online] Available : https://www.researchgate.net/publication/342093248_Machine_Learning_and_Bots_Detection_on_Twitter [Accessed: 05 January ,2021]

[5] Twitter bot detection using machine learnining Vidyadhar S Shelke , Government Engineering College of Auranagabad,Maharashtra 431005; Dr.Avinash K.Gulve, Government Engineering College of Auranagabad,Maharashtra 431005; Dr. Praveen C. Shetiye, Government Engineering College of Auranagabad,Maharashtra 431005
[Online] Available : http://ijsrd.com/Article.php?manuscript=IJSRDV8I50085 [Accessed: 22 January ,2021]

[6] M. Al-Fayoumi, J. Alwidian, M. Abusaif, and I. M. East, “Intelligent Association Classification Technique for Phishing Website Detection,” International Arab Journal of Information Technology, vol. 17, no. 4, 2020.
[Online] Available : https://iajit.org/PDF/July%202020,%20No.%204/18225.pdf [Accessed: 27 January, 2021]

[7] M. Al-Fawa'reh and M. Al-Fayoumiy, “Detecting stealth-based attacks in large campus networks,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 4, pp. 4262–4277, 2020
[Online] Available : https://dl.acm.org/doi/abs/10.1145/3460620.3460739 [Accessed : 09 January, 2021]