A Project Report on

SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS

Submitted in partial fulfillment for the award of

Bachelor of Technology

in

Computer Science and Engineering

By

B. DevaNandini (Y19ACS420) **K. Suma (Y19ACS482)**

L. Ajay Kumar (Y19ACS496) **A. Harsha Vardhan (Y19ACS411)**

Under the guidance of
Mr. V. Naveen Kumar, Assistant Professor

Department of Computer Science and Engineering
Bapatla Engineering College
(Autonomous)
(Affiliated to Acharya Nagarjuna University)
BAPATLA – 522102, Andhra Pradesh, INDIA

2022-2023

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitled **Spammer Detection And Fake User Identification On Social Networks** that is being submitted by **B. DevaNandini (Y19ACS420), K. Suma (Y19ACS482), L. Ajay Kumar (Y19ACS496), A. Harsha Vardhan (Y19ACS411)** in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under my guidance and supervision.

Date:

V. Naveen Kumar
Assistant Professor

Dr. P. Pardhasaradhi
Professor and HoD

Declaration

We declare that this project work is composed by ourselves, and that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

B. DevaNandini (Y19ACS420)
K. Suma (Y19ACS482)
L. Ajay Kumar (Y19ACS496)
A. Harsha Vardhan (Y19ACS411)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Mr. V. Naveen Kumar**, Assistant Professor, Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. P. Pardhasaradhi**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal, **Dr. SK. Nazeer** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to the project coordination committee and our project coordinator **Dr. N. Sudhakar,** Prof., Dept. of CSE for their helpful suggestions throughout the project work and in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

B. DevaNandini (Y19ACS420)
K. Suma (Y19ACS482)
L. Ajay Kumar (Y19ACS496)
A. Harsha Vardhan (Y19ACS411)

Abstract

Social networking sites engage millions of users around the world. The users' interactions with these social sites, such as Twitter and Facebook have a tremendous impact and occasionally undesirable repercussions for the daily life. The prominent social networking sites have turned into a target platform for the spammers to disperse a huge amount of irrelevant and deleterious information. Twitter, for example, has become one of the most extravagantly used platforms of all times and therefore allows an unreasonable amount of spamming.

Fake users send undesired tweets to users to promote services or websites that not only affect the legitimate users but also disrupt the resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased that results in the unrolling of harmful content. Recently, the detection of spammers and identification of fake users on Twitter has become a common area of research in contemporary online social Networks (OSNs).

In this project, we perform a review of techniques used for detecting spammers on Twitter. Moreover, a taxonomy of the Twitter spam detection approaches is presented that classifies the techniques based on their ability to detect: (i) fake content, (ii) spam based on URL, (iii) spam in trending topics, and (iv) fake users. The presented techniques are also compared based on various features, such as user features, content features, graph features, structure features, and time features. We are hopeful that the presented study will be a useful resource for researchers to find the highlights of recent developments in Twitter spam detection on a single platform.

Table Of Contents

List of Figures

1 Introduction

It has become quite unpretentious to obtain any kind of information from any source across the world by using the Internet. The increased demand of social sites permits users to collect abundant amount of information and data about users. Huge volumes of data available on these sites also draw the attention of fake users. Twitter has rapidly become an online source for acquiring real-time information about users. Twitter is an Online Social Network (OSN) where users can share anything and everything, such as news, opinions, and even their moods. Several arguments can be held over different topics, such as politics, current affairs, and important events.

Many people who do not have much information regarding the OSNs can easily be tricked by the fraudsters. There is also a demand to combat and place a control on the people who use OSNs only for advertisements and thus spam other people’s accounts. It is essential to recognize spams in the OSN sites to save users from various kinds of malicious attacks and to preserve their security and privacy. These hazardous maneuvers adopted by spammers cause massive destruction of the community in the real world. Twitter spammers have various objectives, such as spreading invalid information, fake news, rumors, and spontaneous messages.

1.1 Introduction

It has become quite unpretentious to obtain any kind of information from any source across the world by using the Internet. The increased demand of social sites permits users to collect abundant amount of information and data about users. Huge volumes of data available on these sites also draw the attention of fake users. Twitter has rapidly become an online source for acquiring real-time information about users. Twitter is an Online Social Network (OSN) where users can share anything and everything, such as

news, opinions, and even their moods. Several arguments can be held over different topics, such as politics, current affairs, and important events. When a user tweets something, it is instantly conveyed to his/her followers, allowing them to outspread the received information at a much broader level. With the evolution of OSNs, the need to study and analyze users’ behaviors in online social platforms has intensified. Many people who do not have much information regarding the OSNs can easily be tricked by the fraudsters. There is also a demand to combat and place a control on the people who use OSNs only for advertisements and thus spam other people’s accounts.

Recently, the detection of spam in social networking sites attracted the attention of researchers. Spam detection is a difficult task in maintaining the security of social networks. It is essential to recognize spams in the OSN sites to save users from various kinds of malicious attacks and to preserve their security and privacy. These hazardous maneuvers adopted by spammers cause massive destruction of the community in the real world. Twitter spammers have various objectives, such as spreading invalid information, fake news, rumors, and spontaneous messages. Spammers achieve their malicious objectives through advertisements and several other means where they support different mailing lists and subsequently dispatch spam messages randomly to broadcast their interests. These activities cause disturbance to the original users who are known as non-spammers. In addition, it also decreases the repute of the OSN platforms. Therefore, it is essential to design a scheme to spot spammers so that corrective efforts can be taken to counter their malicious activities.

Several research works have been carried out in the domain of Twitter spam detection. To encompass the existing state-of-the-art, a few surveys have also been carried out on fake user identification from Twitter. Tingmin et al. provide a survey of new methods and techniques to identify Twitter spam detection. The above survey presents a comparative study of the current approaches. On the other hand, the authors in conducted a survey on different behaviors exhibited by spammers on Twitter social network. The study also provides a literature review that recognizes the existence of spammers on Twitter social network. Despite all the existing studies, there is still a gap in the existing literature. Therefore, to bridge the gap, we review state-of-the-art in the spammer detection and fake user identification on Twitter. Moreover, this survey presents a taxonomy of the Twitter spam detection approaches and attempts to offer a detailed description of recent developments in the domain.

The aim of this project is to identify different approaches of spam detection on Twitter and to present a taxonomy by classifying these approaches into several categories. For classification, we have identified four means of reporting spammers that can be helpful in identifying fake identities of users. Spammers can be identified based on: (i) fake content, (ii) URL based spam detection, (iii) detecting spam in trending topics, and (iv) fake user identification.

1.2 Motivation

Social networks can provide a range of benefits to members of an organization:

1. Support for learning: Social networks can enhance informal learning and support social connections within groups of learners and with those involved in the support of learning.

2. Support for members of an organization: Social networks can potentially be used my all members of an organization, and not just those involved in working with students. Social networks can help the development of communities of practice.

3. Engaging with others: Passive use of social networks can provide valuable business intelligence and feedback on institutional services (although this may give rise to ethical concerns).

4. Ease of access to information and applications: The ease of use of many social networking services can provide benefits to users by simplifying access to other tools and applications. The Facebook Platform provides an example of how a social networking service can be used as an environment for other tools.

1.3 Problem statement

The detection of spam in social networking sites attracted the attention of researchers. Spam detection is a difficult task in maintaining the security of social networks. It is essential to recognize spams in the OSN sites to save users from various kinds of malicious attacks and to preserve their security and privacy. These hazardous maneuvers adopted by spammers cause massive destruction of the community in the real world. Twitter spammers have various objectives, such as spreading invalid information, fake news, rumors, and spontaneous messages. Spammers achieve their malicious objectives through advertisements and several other means where they support different mailing lists and subsequently dispatch spam messages randomly to broadcast their interests. These activities cause disturbance to the original users who are known as non-spammers.

1.3.1 Problem Solution

Proposed a methodology for the detection of anomalous tweets. The type of abnormality that is distributed on Twitter is the type of URL anomaly. Anomalous users use various URL links for creating spams. The proposed methodology, which is used to identify various anomalous activities from social networking sites, for example, Twitter, comprises the following features.

1. Similarity of tweets includes posting of same tweets again and again.
2. Time difference between tweets involves posting of five or more tweets during the time period of one minute.
3. Malware content consists of malware URL that can damage the system.
4. Adult content contains posts that consist of adult content.

The goal of this work is to discover several ways to spam detection on Twitter and to offer a taxonomy that categorizes these techniques into several groups. For categorization, we have found four methods for reporting spammers that can assist in detecting user impersonation. Spammers can be detected using the following methods as shown in below Figure 1.1.

1. fake content,
2. URL-based spam detection,
3. Spam detection in trending subjects, and
4. Fake user identification

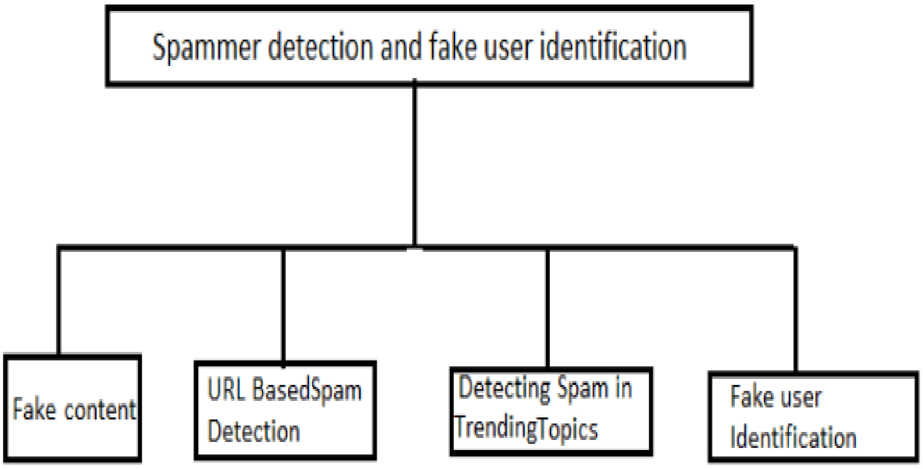

Spammer detection and fake user identification
Fake content
URL BasedSpam Detection
Detecting Spam in TrendingTopics
Fake user Identification

Figure 1.1 Methods to detect spammer and fake user

User's behavior and tweets content have been analyzed for the purpose of finding the best feature to recognize twitter spammers. The objective of the study to detect spam tweets which enhance the quantity to data that needs to be assembled by relying only on tweet-inherent features. To understand the significance of each well-defined edge in order to find the opinion leader and to perceive the weight that could permit more précised opinion based on evaluation algorithms. The goal of the study is to attain real time Twitter spam detection capabilities.

To detect fake accounts on Twitter by proposing classification methods and to illustrate the effect of discretization based on Naïve bayes algorithm in Twitter. Achieve higher accuracy by combining user based, content based, and graph-based features for spam profile detection.

1.4 Scope of Work

Despite the development of efficient and successful ways for spam detection and fake user identification on Twitter, there are still certain gaps in the study that need to be addressed. The following are a few of the issues: Because of the substantial ramifications of false news on an individual and communal level, false news identification on social media networks is a subject that needs to be investigated. The identification of rumor origins on social media is another related topic worth researching. Although a few studies using statistical methods to discover the origin of rumors have already been undertaken, more complex approaches, such as social network-based approaches, can be used due to their demonstrated efficiency.

2 Literature Survey

2.1 Title: Twitter fake account detection.

Authors: B. Erçahin, Ö. Aktaş, D. Kilinç, and C. Akyol

Social networking sites such as Twitter and Facebook attract millions of users across the world and their interaction with social networking has affected their life. This popularity in social networking has led to different problems including the possibility of exposing incorrect information to their users through fake accounts which results to the spread of malicious content. This situation can result to a huge damage in the real world to the society. In our study, we present a classification method for detecting the fake accounts on Twitter. We have preprocessed our dataset using a supervised discretization technique named Entropy Minimization Discretization (EMD) on numerical features and analyzed the results of the Naïve Bayes algorithm.

2.2 Title: Detecting spammers on Twitter.

Author: F. Benevento, G. Magno, T. Rodrigues, and V. Almeida.

With millions of users tweeting around the world, real time search systems and different types of mining tools are emerging to allow people tracking the repercussion of events and news on Twitter. However, although appealing as mechanisms to ease the spread of news and allow users to discuss events and post their status, these services open opportunities for new forms of spam. Trending topics, the most talked about items on Twitter at a given point in time, have been seen as an opportunity to generate traffic and revenue. Spammers post tweets containing typical words of a trending topic and URLs, usually obfuscated by URL shorteners, that lead users to completely unrelated websites. This kind of spam can contribute to de-value real time search services unless mechanisms to fight and stop spammers can be found. In this project we consider the problem of detecting spammers on

Twitter. We first collected a large dataset of Twitter that includes more than 54 million users, 1.9 billion links, and almost 1.8 billion tweets. Using tweets related to three famous trending topics from 2009, we construct a large labeled collection of users, manually classified into spammers and non-spammers. We then identify a number of characteristics related to tweet content and user social behavior, which could potentially be used to detect spammers. We used these characteristics as attributes of machine learning process for classifying users as either spammers or non-spammers. Our strategy succeeds at detecting much of the spammers while only a small percentage of non-spammers are misclassified. Approximately 70% of spammers and 96% of no spammers were correctly classified. Our results also highlight the most important attributes for spam detection on Twitter.

2.3 Title: An approach for malicious tweets detection using NLP.

Author: S. Gharge, and M. Chavan.

Many previous works have focused on detection of malicious user accounts. Detecting spams or spammers on Twitter has become a recent area of research in social network. However, we present a method based on two new aspects: the identification of spam tweets without knowing previous background of the user; and the other based on analysis of language for detecting spam on twitter in such topics that are in trending at that time. Trending topics are the topics of discussion that are popular at that time. This growing micro blogging phenomenon therefore benefits spammers. Our work tries to detect spam tweets in based on language tools. We first collected the tweets related to many trending topics, labelling them based on their content which is either malicious or safe. After a labelling process we extracted a many features based on the language models using language as a tool. We also evaluate the performance and classify tweets as spam or not spam. Thus, our system can be applied for detecting spam on Twitter, focusing mainly on analyzing of tweets instead of the user accounts.

2.4 Title: Twitter spam detection: Survey of new approaches

Author: T. Wu, S. Wen, Y. Xiang, and W. Zhou.

Twitter spam has long been a critical but difficult problem to be addressed. So far, researchers have proposed many detection and defense methods in order to protect Twitter users from spamming activities. Particularly in the last three years, many innovative methods have been developed, which have greatly improved the detection accuracy and efficiency compared to those which were proposed three years ago. Therefore, we are motivated to work out a new survey about Twitter spam detection techniques. This survey includes three parts: 1) A literature review on the state-of-art: this part provides detailed analysis (e.g. taxonomies and biases on feature selection) and discussion (e.g. pros and cons on each typical method); 2) Comparative studies: we will compare the performance of various typical methods on a universal testbed (i.e. same datasets and ground truths) to provide a quantitative understanding of current methods; 3) Open issues: the final part is to summarize the unsolved challenges in current Twitter spam detection techniques. Solutions to these open issues are of great significance to both academia and industries. Readers of this survey may include those who do or do not have expertise in this area and those who are looking for deep understanding of this field in order to develop new methods.

2.5 Title: Survey on spammers in popular social media networks.

AUTHOR: S. J. Soman.

Social networking sites have become a major factor of the Web and are playing an important role in the life of human being. People communicate with each other through social networking services (SNSs). Unfortunately, the Blogosphere has been infected by different forms of spam-like contents. The rise of social networking sites made them the targets of spammers as they lead the users to be fed up with irrelevant information while surfing. During early days, researchers were concentrating on the development of Honey pots for detecting spams. Twitter is a target platform for promoters and spammers. The authors survey the related literature that identifies the presence of spam as well as spammers in popular social media networks.

3 System Analysis

System analysis is a review of a technological system, like a software package, for troubleshooting, development, or improvement purposes. With a systems analysis, considering the goals of the system is important for solving problems and creating efficiencies. From there, dividing a system into components can make it easier to perform individual analyses that influence the complete system. In this chapter, we discuss about the existing system, proposed system, system requirements and the classification techniques.

3.1 Existing System

In the field of Twitter spam detection, several studies have been conducted. A few polls on false user identification from Twitter were also conducted to cover the current state-of-the-art. Present a review of new methodologies and techniques for detecting Twitter spam. The survey above provides a comparative analysis of existing techniques. The authors of conducted a survey on the various behaviors displayed by spammers on the Twitter social network. The research also includes a literature analysis that acknowledges the existence of spammers on Twitter. Despite all the studies that have been done, there is still a void in the literature. As a result, we examine the state-of-the-art in spammer detection and fake user identification on Twitter in order to close the gap. Furthermore, this study gives a taxonomy of Twitter spam detection methods and strives to provide a comprehensive overview of current developments in the field.

In the existing system the authors conducted a survey on different behaviors exhibited by spammers on Twitter social network. The study also provides a literature review that recognizes the existence of spammers on Twitter social network. Despite all the existing studies, there is still a gap in the existing literature.

3.1.1 Disadvantages of existing system:

1. No efficient methods used.
2. No real time data used.
3. More complex.

Therefore, to bridge the gap, we finding the spammer detection and fake user identification on Twitter by using four types: fake content, URL based spam detection, detecting spam in trending topics, and fake user identification.

3.2 Proposed System

The aim of this project is to identify different approaches of spam detection on Twitter and to present a taxonomy by classifying these approaches into several categories. For classification, we have identified four means of reporting spammers that can be helpful in identifying fake identities of users. Spammers can be identified based on:

1. fake content,
2. URL based spam detection,
3. detecting spam in trending topics, and
4. fake user identification.

Moreover, the analysis also shows that several machine learning-based techniques can be effective for identifying spams on Twitter. However, the selection of the most feasible techniques and methods is highly dependent on the available data.

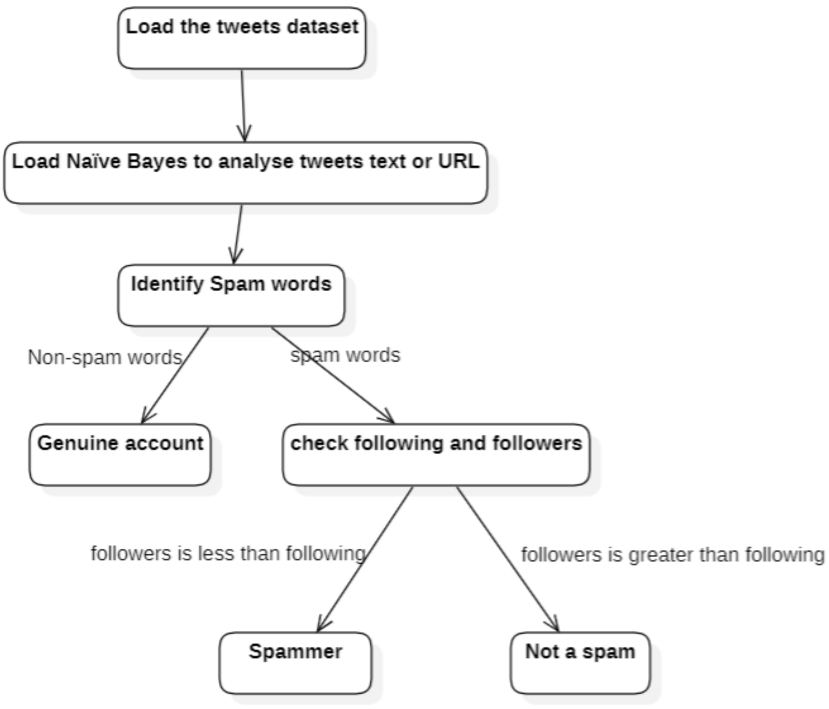

Load the tweets dataset
Load Naïve Bayes to analyse tweets text or URL
Identify Spam words
Non-spam words
spam words
Genuine account
check following and followers
followers is less than following
followers is greater than following
Spammer
Not a spam

Figure 3.1 Module Design and Functionality

At first, we have to click on 'Upload Twitter JSON Format Tweets Dataset' button and upload tweets folder. We are uploading the JSON format tweets because the posts are different and the different users must post different content in different languages, those tweets are taken as single file in the Tweets dataset. And then click on 'Load Naive Bayes to Analyze Tweet Text or URL' button to load Naïve Bayes classifier. The tweets that are loaded must be preprocessed by using naïve bayes classifier. The naïve bayes classifier preprocess the tweets and shows that Naïve Bayes is loaded. And now click on 'Detect Fake Content, Spam URL, Trending Topic & Fake Account' to analyses each tweet for fake content, spam URL and fake account using Naïve Bayes classifier and other above mention technique. The characteristics retrieved from the tweets dataset, which are then analyzed to determine if a tweet is spam or not. The detection result is shown in the last column, and each spam row has less followers and followers, indicating that this account is false and that the user is only using it to disseminate spam messages and is not establishing any friends or following anybody. And now click on ‘Detection Graph’ button to know total tweets and spam and fake account graph. The results of the study shows that random forest classifier achieves high spam detection accuracy in real-time. The random forest classifier model will be used to predict/detect fake or spam account for upcoming future tweets.

3.2.1 Fake content

Fake tweet user accounts were analyzed by the activities performed by user accounts from where the spam tweets were generated. It was observed that most of the fake tweets were shared by people with followers. Subsequently, the sources of tweet analysis were analyzed by the medium from where the tweets were posted. It was found that most of the tweets containing any information were generated through mobile devices and non-informative tweets were generated more through the Web interfaces. The role of user attributes in the identification of fake content was calculated through: (i) the average number of verified accounts that were either spam or non-spam and (ii) the number of followers of the user accounts. The fake content propagation was identified through the metrics that include: (i) social reputation, (ii) global engagement, (iii) topic engagement, (iv) likability, and (v) credibility. After that, the authors utilized regression prediction model to ensure the overall impact of people who spread the fake content at that time and to predict the fake content growth in future. The spammer identification is shown in Figure 3.1 .

3.2.2 URL based spam detection

Chen et al. played out an assessment of AI calculations to distinguish spam tweets. The creators examined the effect of different features on the exhibition of spam identification, for instance: (i) spam to non-spam proportion, (ii) size of preparing dataset, (iii) time related information, (iv) factor discretization, what is more, (v) examining of information. To assess the location, first, around 600 million open tweets were gathered and in this manner the creators applied the Trend miniaturized scale's web notoriety framework to distinguish spam tweets however much as could be expected. An aggregate of 12 lightweight highlights were likewise isolated to recognize non-spam and spam tweets from this identified dataset. classification, which are later utilized in the investigation to assess the location of spam. Four datasets are tested to repeat various situations. Since no dataset is accessible openly for the assignment, few datasets were utilized in past investigates.

After the identification of spam tweets, 12 highlights were assembled. These highlights are separated into two classes, i.e., client-based highlights and tweet-based highlights. The client-based highlights are identified through different items for example, account age and number of client top picks, records, and tweets. The identified client-based highlights are parsed from the JSON structure. Then again, the tweet-based highlights incorporate the quantity of (i) retweets, (ii) hashtags, (iii) client notices, and (iv) URLs. The consequence of assessment shows that the changing element dispersion decreased the presentation though no distinctions were seen in the preparation dataset circulation.

3.2.3 Detecting Spam in Trending topics

Gharge et al. [3] start a technique, which is classified on the premise of two new viewpoints. The first one is the acknowledgment of spam tweets with no earlier data about the clients what is more, the subsequent one is the investigation of language for spam recognition on Twitter drifting theme around then. The spammer identification is shown in Figure 3.1 .The framework structure incorporates the accompanying five steps.

Step 1: The assortment of tweets regarding drifting points on Twitter. In the wake of putting away the tweets in a specific position, the tweets are hence examined.

Step 2: Labeling of spam is performed to check through all datasets that are accessible to recognize the dangerous URL.

Step 3: Feature extraction isolates the qualities develop in view of the language model that utilizes language as apparatus and aides in deciding if the tweets are counterfeit or not.

Step 4: The classification of informational collection is performed by shortlisting the arrangement of tweets that is depicted by the arrangement of features given to the classifier to train the model and to secure the information for spam location.

Step 5: The spam location utilizes the classification system to acknowledge tweets as the info and characterize the spam and non-spam.

3.2.4 Fake User Identification

A categorized strategy is proposed by Er³ahin et al. to distinguish spam accounts on Twitter. The dataset utilized in the investigation was gathered physically. The classification is performed by examining client name, prole and foundation picture, number of companions and devotees, substance of tweets, depiction of record, and number of tweets. The dataset included 501 phony and 499 genuine records, where 16 highlights from the data that were acquired from the Twitter APIs were identified. Two examinations were performed for characterizing counterfeit records. The first test utilizes the Naïve Bayes learning calculation on the Twitter dataset including all angles without discretization, though the subsequent analysis employments the Naïve Bayes learning calculation on the Twitter dataset after the discretization. Mateen et al. proposed a half and half procedure that uses client based, content-based, and chart-based qualities for spammer recognition. A model is proposed to separate between the non-spam and spam profiles utilizing three attributes. The proposed procedure was breaking down utilizing Twitter dataset with 11K clients and around 400K tweets. The objective is to accomplish higher efficiency and accuracy by coordinating every one of these qualities. Client based highlights are set up in view of relationship and properties of client accounts. It is fundamental to add client-based highlights for the spam recognition model. As these highlights are identified with client accounts, all characteristics, which were connected to client accounts, were identified.

These properties incorporate the quantity of devotees what is more, after, age, FF proportion, and notoriety. On the other hand, content highlights are connected to the tweets that are posted by clients as spam bots that post a colossal measure of copy substance as difference to non-spammers who do not post copy tweets. These features rely upon messages or substance that clients compose. Spammers present substance on spread phony news and these substances contain noxious URL to advance their item. The substance-based highlights include: (i) the all-out number of tweets, (ii) hash tag proportion, (iii) URLs proportion, (iv) specifies proportion, and (v) frequency of tweets.

3.3 System Requirements

Functional requirements for a secure cloud storage service are straightforward: The service should be able to store the user’s data, The data should be accessible through any devices connected to the Internet, The service should be capable to synchronize the user’s data between multiple devices (notebooks, smart phones, etc.), The service should preserve all historical changes (versioning), Data should be shareable with other users, The service should support SSO and The service should be interoperable with other cloud storage services, enabling data migration from one CSP to another.

3.3.1 Software Requirements

1. **Operating System:** Windows
2. **Coding Language:** Python
3. **IDE:** Python IDE
4. **Version:** Python 3.7

3.3.2 Hardware requirements

1. **Processor:** intel CORE i3
2. **RAM:** 4 GB (min)
3. **Hard Disk:** 128 GB (min)
4. **Key Board:** Standard Keyboard

3.4 Algorithms

An algorithm is a procedure used for solving a problem or performing a computation. Algorithms act as an exact list of instructions that conduct specified actions step by step in either hardware- or software-based routines.

Algorithms are widely used throughout all areas of IT. In mathematics and computer science, an algorithm usually refers to a small procedure that solves a recurrent problem. Algorithms are also used as specifications for performing data processing and play a major role in automated systems.

An algorithm could be used for sorting sets of numbers or for more complicated tasks, like recommending user content on social media. Algorithms typically start with initial input and instructions that describe a specific computation. When the computation is executed, the process produces an output.

3.4.1 Random Forest Algorithm

In this we are using random forest which is comes under supervised learning in machine learning. Random forest algorithm which is used to classification, in this project we are going to identify the spammer and firstly we must categorize the spammer after that we are going to identify the spammer.

Steps for Random Forest algorithm

Step 1: Gather the different training data from the training dataset.

Step 2: In each data which we are gathered we must take the information.

Step 3: Finally, we have to predict the data

Random Forest is a classifier that contains a few decision trees on various subsets of given dataset and takes the average to improve the predictive accuracy of that data. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The working of the Random Forest algorithm is shown in below Figure 3.2 .

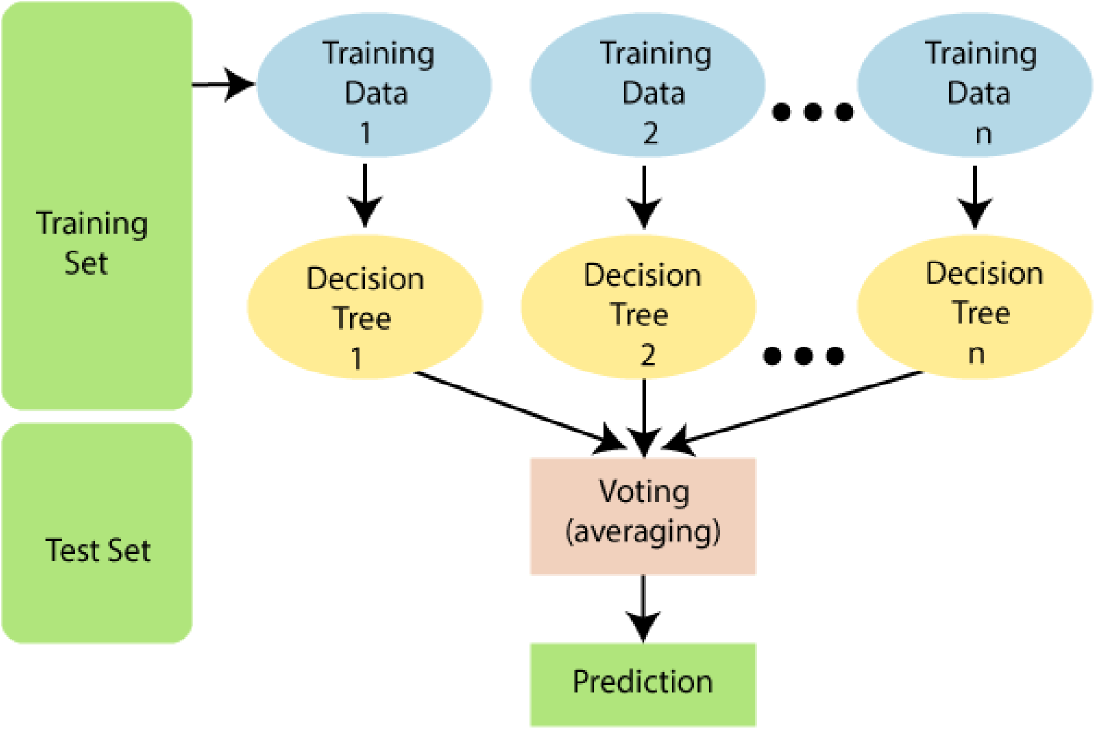


Figure 3.2 Working of random forest

3.4.2 Naïve Bayes algorithm

Naïve Bayes algorithm is used to detect the user tweets whether the user tweets information are spam or non-spam. Naïve Bayes is better for classifying the user tweets with stemming techniques and stop words. And collect dataset from tweeter and it is used for classifying the user tweets into spammer or non-spammer.

With Bayes' Rule, we want to find the probability an account is spammer, given it contains certain words. We do this by finding the probability that each word in the account is spam, and then multiply these probabilities together to get the spam metric to be used in classification.

Naïve Bayes algorithm can be used for classification problems. Naïve Bayes algorithm is used to detect the user tweets whether the user tweets information are rumor or non-rumor. Naïve Bayes is better for classifying the user tweets with stemming techniques and stop words. And collect dataset from tweeter and it is used for classifying the user tweets into spammer or non-spammer. Naïve bayes algorithm working can be shown in Figure 3.3.

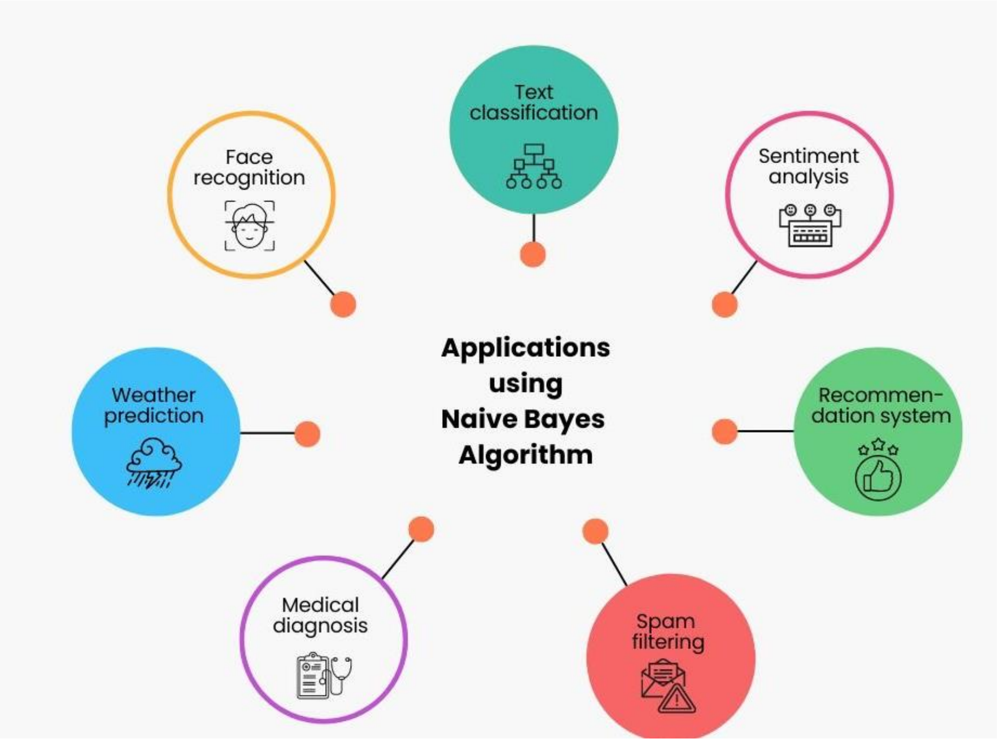


Figure 3.3 Naive bayes uses

3.4.3 Machine Learning Algorithm

A machine learning-based approach to Twitter detection necessitates the creation of a framework in which tweets are represented by a feature space. Similarly, each tweet is ultimately capacity y = f(x) models the link between the information space and the category labels, such as spammer and known spammer. Finally, empirical learning of the capacity f(x) is based on a preparation method that employs a dataset, D, including N patterns (samples); each pattern comprises a that is not part of the preparation set and assigns each test sample to a predicted category, y. The classifier shows the extracted data weather it is spam or non-spammer as shown below Figure 3.4 .

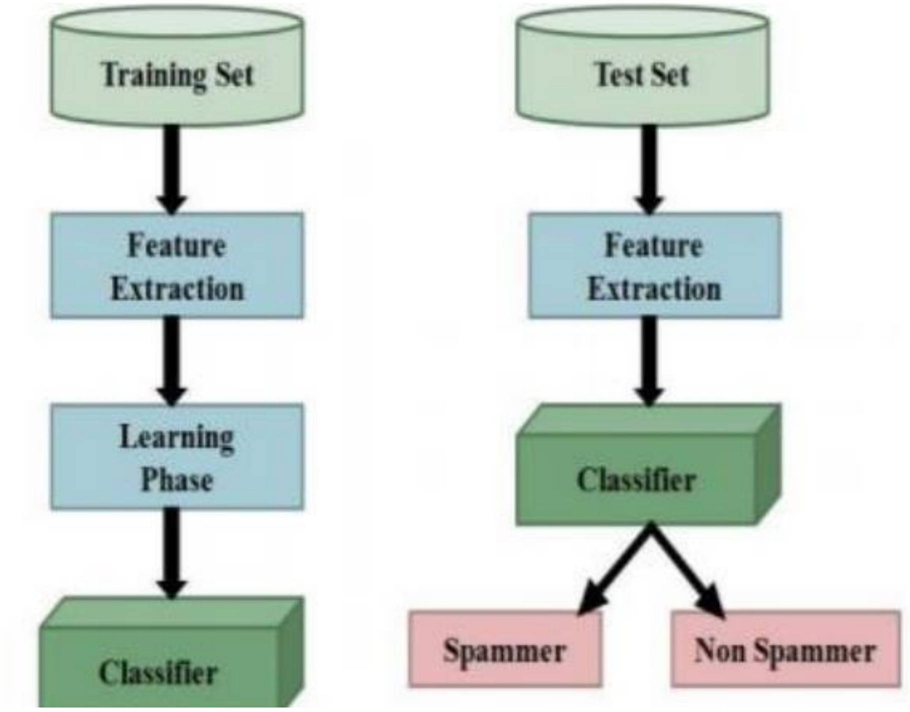


Figure 3.4 Machine Learning based classification system

4 Design

Design part is illustrated with UML Diagrams. A Diagram is the graphical presentation of asset of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). For this reason, and the UML includes nine such diagrams. The Unified Modeling Language (UML) is probably the most widely known and used notation for object-oriented analysis and design. It is the result of the merger of several early contributions to object-oriented methods. The Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts. A Modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of a system. Modeling is the designing of software applications before coding.

4.1 UML Diagrams

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we can better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational_Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates.

Goals:

1. The Primary goals in the design of the UML are as follows:
2. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
3. Provide extendibility and specialization mechanisms to extend the core concepts.
4. Be independent of programming languages and development process.
5. Provide a formal basis for understanding the modeling language.
6. Encourage the growth of OO tools market.
7. Support higher level development concepts such as collaborations, frameworks, patterns, and components.
8. Integrate best practices.

4.1.1 Use Case Diagram of spam detection

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The working of the use case is shown in below Figure 4.1.

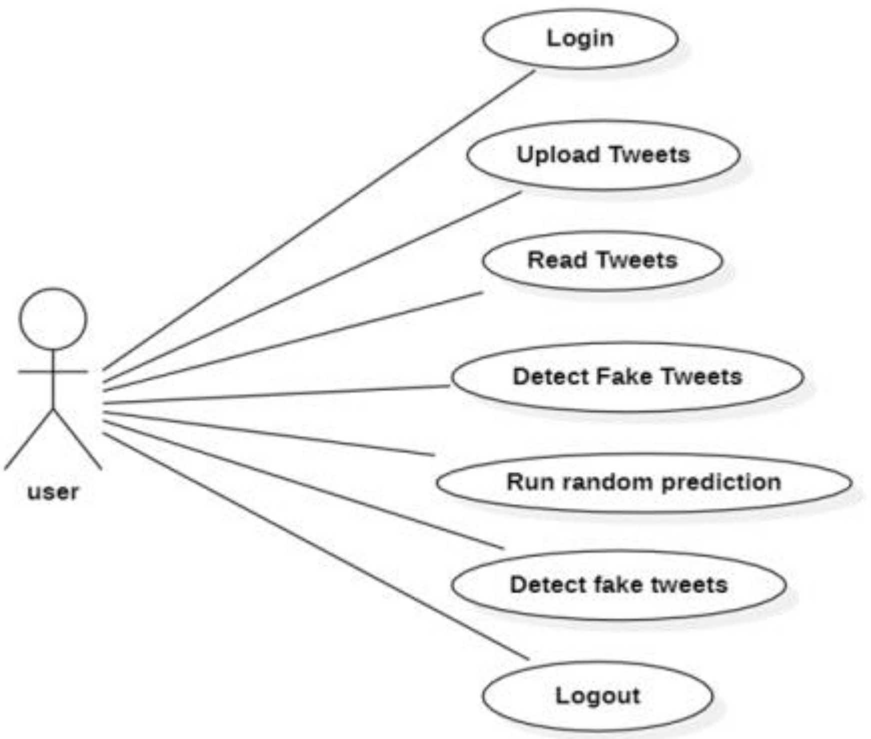


Figure 4.1 Use case diagram of spam detection

4.1.2 Data Flow diagram of spam detection

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used in the following Figure 4.2.

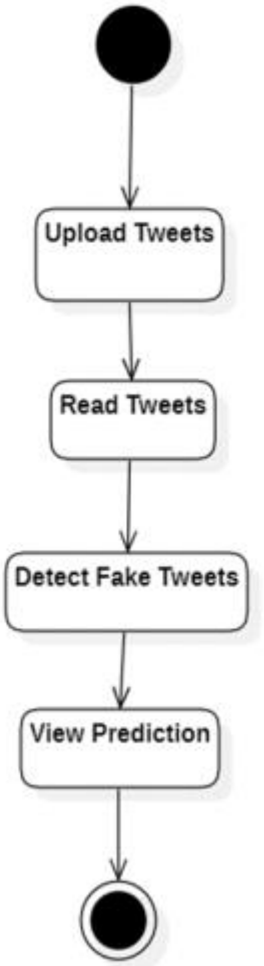


Figure 4.2 Data Flow Diagram of spam detection

4.1.3 Sequence Diagram of spam detection

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. The sequence diagram determines the spammers as shown below Figure 4.3.

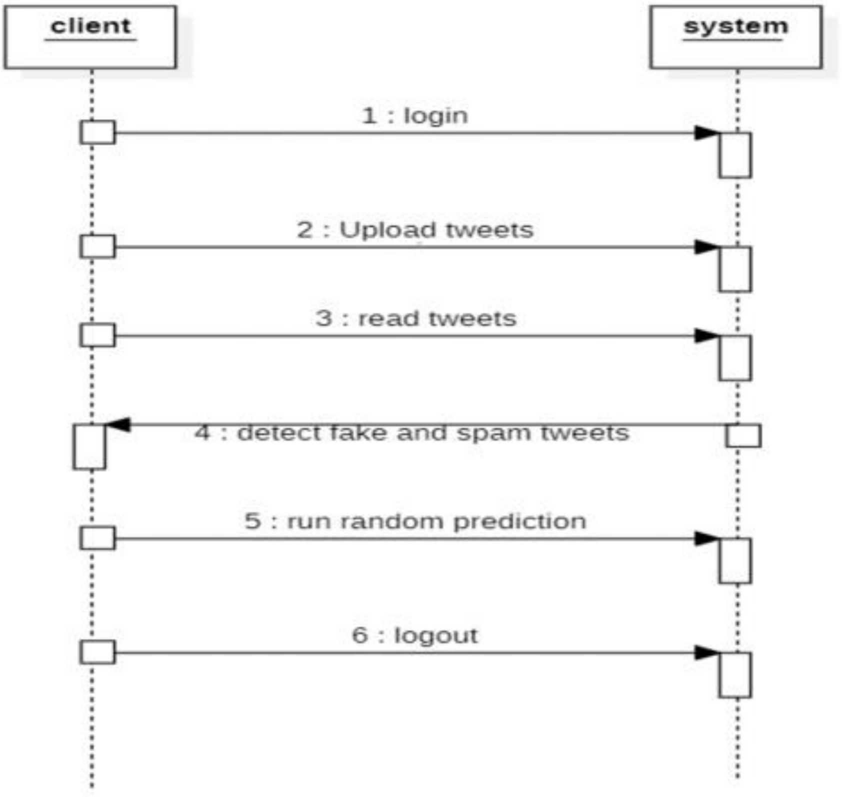


Figure 4.3 Sequence Diagram of spam detection

5 Implementation

In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal, Java, .Net are used for coding. With respect to the type of application, the right programming language is chosen.

5.1 Libraries

A **library** is an umbrella term that comprises a reusable set of Python code/instructions. A Python library is typically a collection of similar modules grouped together under a single name. Developers commonly utilize it to share reusable code with the community. This eliminates the need to write Python code from scratch.

Developers and community researchers can construct their own set of useful functions in the same domain. When programmers and developers install the Python interpreter on their machines, standard libraries are included. Python libraries include matplotlib, Pygame, Pytorch, Requests, Beautiful soup, and others.

5.1.1 NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

1. A powerful N-dimensional array object

2. Sophisticated (broadcasting) functions

3. Tools for integrating C/C++ and Fortran code
4. Useful linear algebra, Fourier transform, and random number capabilities
5. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to integrate with a wide variety of databases seamlessly and speedily.

5.1.2 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

5.1.3 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object-oriented interface or via a set of functions familiar to MATLAB users.

5.1.4 Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use Python.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.

Python is Interactive − you can sit at a Python prompt and interact with the interpreter directly to write your programs.

5.1.5 tkinter

The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the _tkinter module for more information about supported versions.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

5.2 Modules

Modules are collections of related code that are packaged together in a Python program. Within a module, programmers can define functions, classes, or variables. It's also great to accommodate runnable codes within modules. They are, in other words, Python files that contain valid Python definitions and statements. When these files are created, the suffix.py is appended to them. By grouping related code into modules, the code becomes easier to understand and implement. It also organizes the code in a logical manner.

1. Collecting Datasets

Data collection is one of the most important parts of building machine learning models. Because no matter how well designed our model is, it will not learn anything useful if the training data is invalid.

Dataset is collected from the Kaggle repository named as Tweets Dataset. The Tweets is in the form of JSON files dataset contains spam tweets and the account is spammer or not. This dataset contains more than 148,517 records.

2. Pre-Processing the Dataset

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this we use data pre-processing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

As a part of the Data Pre-processing, we check for any null or missing values in the dataset. Noisy Data and Unnecessary Data is removed from the dataset.

3. **Model Building**

   Building a model in machine learning is creating a mathematical representation by generalizing and learning from training data. Then, the built machine learning model is applied to new data to make predictions and obtain results. Here we build the model using supervised machine learning algorithms – Random Forest Algorithm and Naïve Bayes algorithm.

4. **Train the model**

   After creating a model, we divide the data into training and testing data sets then we train the model using the proposed algorithms and then check the accuracy.

5. **User Input**

   Here we upload the tweets data as the input to find the whether the user is spammer or not.

6. **Final Prediction**

   Here we make final prediction using Random Forest and Naïve Bayes algorithms and classify whether the user is spammer or non-spammer.

5.3 Source Code

We build the model using supervised machine learning algorithms – Random Forest Algorithm and Naïve Bayes algorithm. After creating a model, we divide the data into training and testing data sets then we train the model using the proposed algorithms and then check the accuracy. We upload the tweets data as the input to find the whether the user is spammer or not. Here we make final prediction using Random Forest and Naïve Bayes algorithms and classify whether the user is spammer or non-spammer.

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
import json
import os
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
import pickle as cpickle

main = tkinter.Tk()
main.title("Spammer Detection") #designing main screen
main.geometry("1300x1200")

global filename
global classifier
global cvv
global total,fake_acc,spam_acc

def process_text(text):
    nopunc = [char for char in text if char not in
string.punctuation]
    nopunc = ''.join(nopunc)
    clean_words = [word for word in nopunc.split() if
word.lower() not in stopwords.words('english')]
    return clean_words
```

```
def upload(): #function to upload tweeter profile
    global filename
    filename = filedialog.askdirectory(initialdir=".")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");



def naiveBayes():
    global classifier
    global cvv
    text.delete('1.0', END)
    classifier = cpickle.load(open('model/naiveBayes.pkl',
'rb'))
    cv =
CountVectorizer(decode_error="replace",vocabulary=cpickle.loa
d(open("model/feature.pkl", "rb")))
    cvv =
CountVectorizer(vocabulary=cv.get_feature_names(),stop_words
= "english", lowercase = True)
    text.insert(END,"Naive Bayes Classifier loaded\n");



def fakeDetection(): #extract features from tweets
    global total,fake_acc,spam_acc
    total = 0
    fake_acc = 0
    spam_acc = 0
    text.delete('1.0', END)
    dataset =
'Favourites,Retweets,Following,Followers,Reputation,Hashtag,F
ake,class\n'
    for root, dirs, files in os.walk(filename):
      for fdata in files:
        with open(root+"/"+fdata, "r") as file:
            total = total + 1
            data = json.load(file)
            textdata = data['text'].strip('\n')
            textdata = textdata.replace("\n"," ")
            textdata = re.sub('\W+',' ', textdata)
            retweet = data['retweet_count']
            followers = data['user']['followers_count']
            density = data['user']['listed_count']
            following = data['user']['friends_count']
            replies = data['user']['favourites_count']
            hashtag = data['user']['statuses_count']
            username = data['user']['screen_name']
            words = textdata.split(" ")
            text.insert(END,"Username : "+username+"\n");
            text.insert(END,"Tweet Text : "+textdata+"\n");
            text.insert(END,"Retweet Count :
"+str(retweet)+"\n")
            text.insert(END,"Following :
"+str(following)+"\n")
            text.insert(END,"Followers :
"+str(followers)+"\n")
            text.insert(END,"Reputation :
"+str(density)+"\n")
            text.insert(END,"Hashtag : "+str(hashtag)+"\n")
            text.insert(END,"Tweet Words Length :
"+str(len(words))+"\n")
            test = cvv.fit_transform([textdata])
            spam = classifier.predict(test)
            cname = 0
            fake = 0
            if spam == 0:
                text.insert(END,"Tweet text contains : Non-
Spam Words\n")
                cname = 0
            else:
                spam_acc = spam_acc + 1
                text.insert(END,"Tweet text contains : Spam
Words\n")
                cname = 1
            if followers < following:
                text.insert(END,"Twitter Account is Fake\n")
                fake = 1
                fake_acc = fake_acc + 1
            else:
                text.insert(END,"Twiiter Account is 
Genuine\n")
                fake = 0
            text.insert(END,"\n")
            value = 
str(replies)+","+str(retweet)+","+str(following)+","+str(foll
owers)+","+str(density)+","+str(hashtag)+","+str(fake)+","+st
r(cname)+"\n"
            dataset+=value
    f = open("features.txt", "w")
    f.write(dataset)
    f.close()

def prediction(X_test, cls):  #prediction done here
    y_pred = cls.predict(X_test)
    for i in range(len(X_test)):
        print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
    return y_pred

# Function to calculate accuracy
def cal_accuracy(y_test, y_pred, details):
    accuracy = 30 + (accuracy_score (y_test, y_pred) *100)
    text.insert(END, details+"\n\n")
    text.insert(END, “Accuracy : "+str(accuracy)+"\n\n")
    return accuracy




def machineLearning():
    text.delete('1.0', END)
    train = pd.read_csv("features.txt")
    X = train.values[:, 0:7]
    Y = train.values[:, 7]
    X_train, X_test, y_train, y_test = train_test_split(X, Y, 
test_size = 0.2, random_state = 0)
    cls = 
RandomForestClassifier(n_estimators=10,max_depth=10,random_st
ate=None)
    cls.fit(X_train, y_train)
    text.insert(END, “Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    random_acc = cal_accuracy(y_test, prediction_data, 
‘Random Forest Algorithm Accuracy')

def graph():
    height = [total,fake_acc,spam_acc]
    bars = ('Total Twitter Accounts', 'Fake Accounts’, ‘Spam 
Content Tweets')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

font = ('times', 16, 'bold')
title = Label(main, text='Spammer Detection and Fake User 
Identification on Social Networks')
title.config(bg='brown', fg='white')
title.config(font=font)
title.config (height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Upload Twitter JSON Format 
Tweets Dataset", command=upload)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='brown', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=470,y=100)

fakeButton = Button(main, text="Load Naive Bayes To Analise
Tweet Text or URL", command=naiveBayes)
fakeButton.place(x=50,y=150)
fakeButton.config(font=font1)

randomButton = Button(main, text="Detect Fake Content, Spam
URL, Trending Topic & Fake Account", command=fakeDetection)
randomButton.place(x=520,y=150)
randomButton.config(font=font1)

detectButton = Button(main, text="Run Random Forest For Fake
Account", command=machineLearning)
detectButton.place(x=50,y=200)
detectButton.config(font=font1)

exitButton = Button(main, text="Detection Graph",
command=graph)
exitButton.place(x=520,y=200)
exitButton.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=30,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)

main.config(bg='brown')
main.mainloop()
```

6 Results

In this, we develop the Online Social Networking (OSN) system module. We build up the system with the feature of Online Social Networking system, Twitter. We will be using a Python Library to connect to the Twitter API and collect the data. We download tweets containing certain key words, to incorporate the words or hash tags that contain relevant keyword related to fake users.

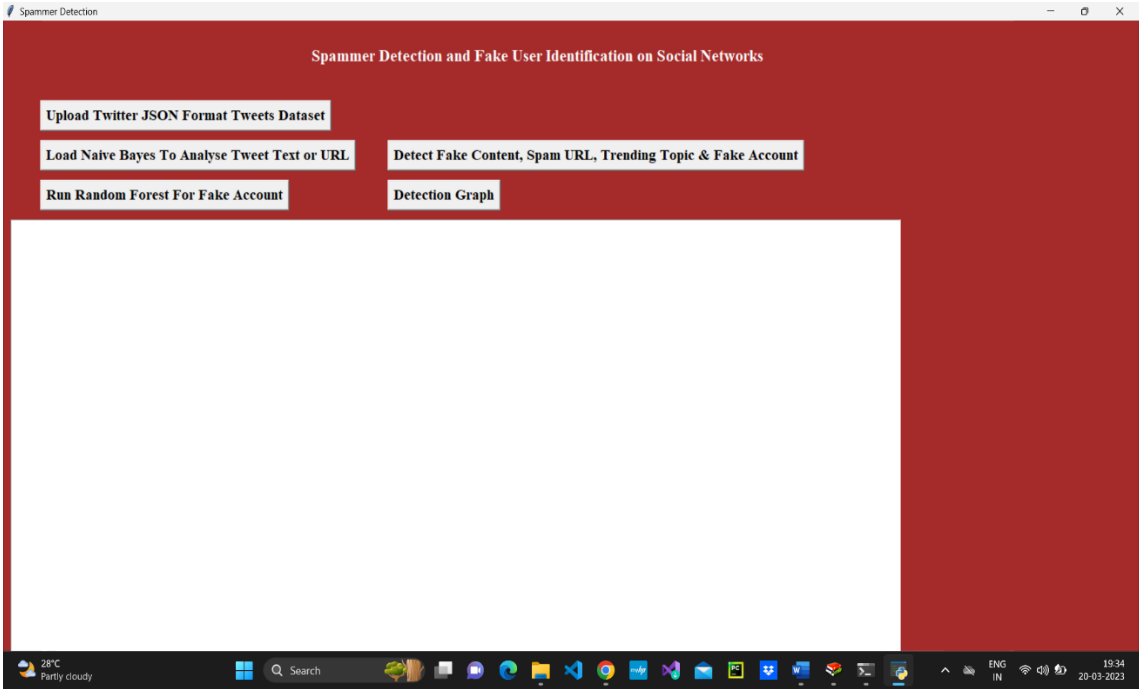
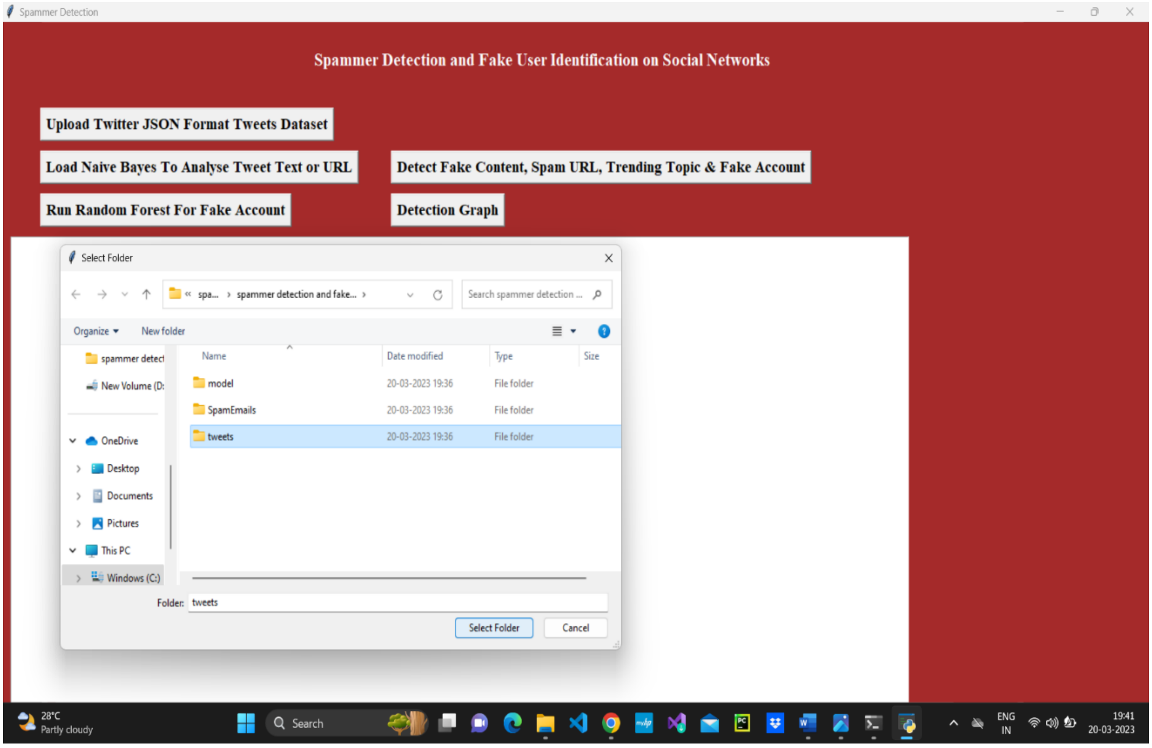

Spammer Detection
Spammer Detection and Fake User Identification on Social Networks
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
28°C
Partly cloudy
Search
ENG
IN
19:34
20-03-2023

Figure 6.1 Output screen of spam detection

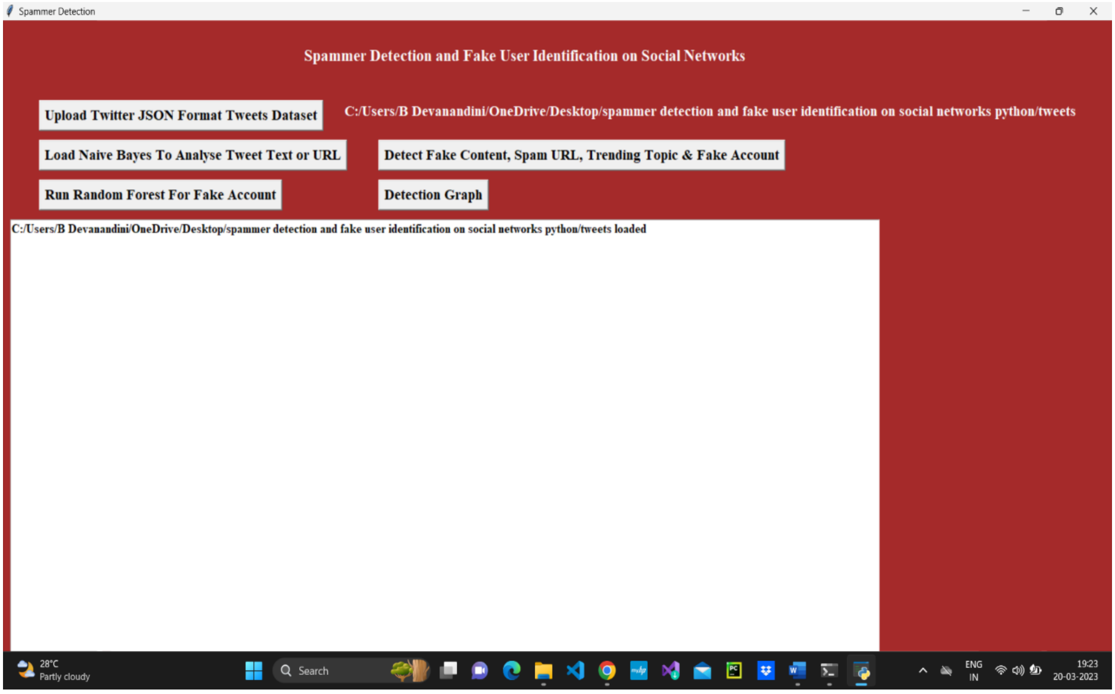
In Figure 6.1 as shown above screen click on 'Upload Twitter JSON Format Tweets Dataset' button and upload tweets folder. We are uploading the JSON format tweets because the posts are different and the different users has to post different content in different languages, those tweets are taken as single file in the Tweets dataset.


Spammer Detection
Spammer Detection and Fake User Identification on Social Networks
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
Select Folder
« spa... › spammer detection and fake... ›
Search spammer detection ...
Organize ▾
New folder
spammer detect
New Volume (D:
OneDrive
Desktop
Documents
Pictures
This PC
Windows (C:)
Name
Date modified
Type
Size
model 20-03-2023 19:36 File folder
SpamEmails 20-03-2023 19:36 File folder
tweets 20-03-2023 19:36 File folder
Select Folder
Cancel
28°C
Partly cloudy
Search
ENG
IN
19:41
20-03-2023

Folder: tweets

Figure 6.2 Upload Tweets of spam detection

In Figure 6.2 as shown above screen I am uploading ‘tweets’ folder which contains tweets from various users in JSON format. Now click open button to start loading the tweets. We can view all tweets from all users. The first column provides the user's id, while the second column includes the user's tweets. To begin reading tweets, click the open button, then the tweets are loaded.

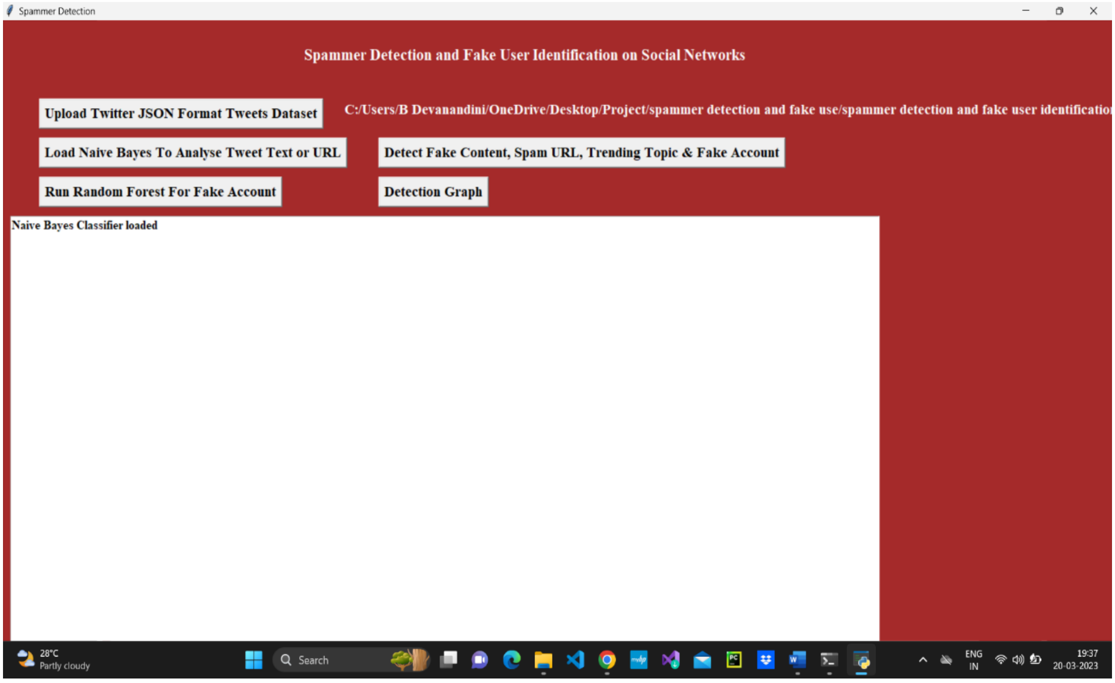

Spammer Detection
Spammer Detection and Fake User Identification on Social Networks
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
28°C
Partly cloudy
Search
ENG
IN
19:23
20-03-2023

C:/Users/B Devanandini/OneDrive/Desktop/spammer detection and fake user identification on social networks python/tweets

```
C:/Users/B Devanandini/OneDrive/Desktop/spammer detection and fake user identification on social networks python/tweets loaded
```

Figure 6.3 Tweets Uploaded screen of spam detection

In Figure 6.3 as shown above screen we can see all tweets from all users loaded. Now click on 'Load Naive Bayes to Analyze Tweet Text or URL' button to load Naïve Bayes classifier. The tweets that are loaded must be preprocessed by using naïve bayes classifier. The naïve bayes classifier preprocess the tweets and shows that Naïve Bayes is loaded.


Spammer Detection
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
28°C
Partly cloudy
Search
ENG
IN
19:37
20-03-2023

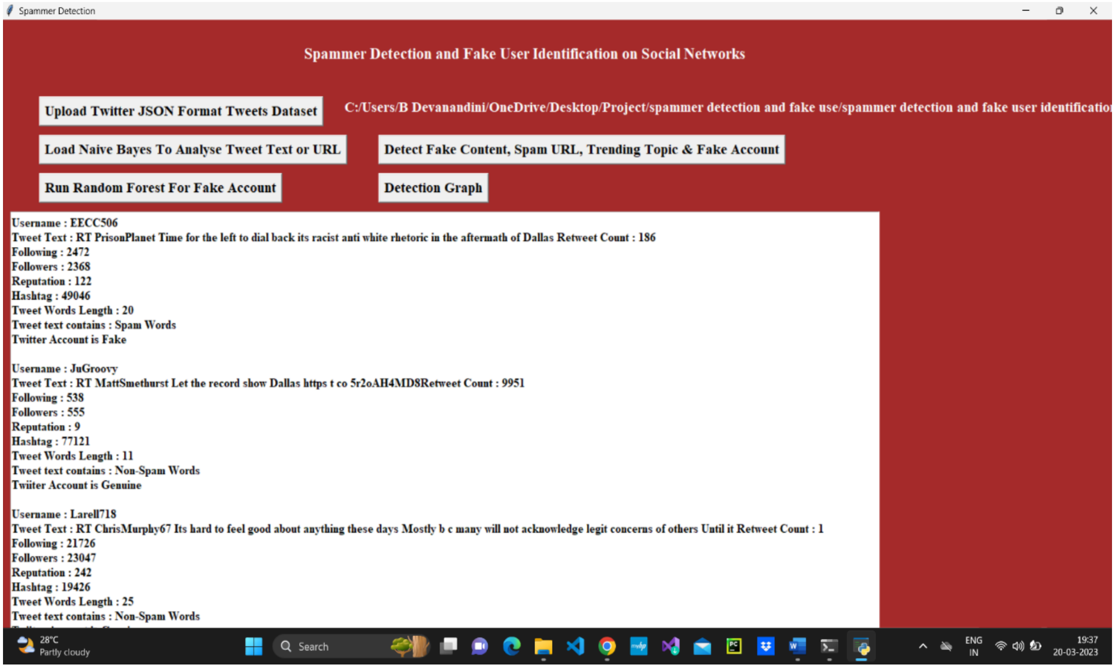
Spammer Detection and Fake User Identification on Social Networks

C:/Users/B Devanandini/OneDrive/Desktop/Project/spammer detection and fake use/spammer detection and fake user identificatio

Naive Bayes Classifier loaded

Figure 6.4 Naive Bayes Loaded screen of spam detection

In Figure 6.4 as shown above screen naïve bayes classifier loaded. The naïve bayes classifier preprocess the text tweets and shows that Naïve Bayes is loaded and now click on 'Detect Fake Content, Spam URL, Trending Topic & Fake Account' to analyses each tweet for fake content, spam URL and fake account using Naïve Bayes classifier and other above mention technique. The characteristics retrieved from the tweets dataset, which are then analyzed to determine if a tweet is spam or not. The detection result is shown in the last column, and each spam row has less followers and followers, indicating that this account is false and that the user is only using it to disseminate spam messages and is not establishing any friends or following anybody as shown below Figure 6.5 .


Spammer Detection
Spammer Detection and Fake User Identification on Social Networks
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
28°C
Partly cloudy
Search
ENG
IN
19:37
20-03-2023

C:/Users/B Devanandini/OneDrive/Desktop/Project/spammer detection and fake use/spammer detection and fake user identificatio

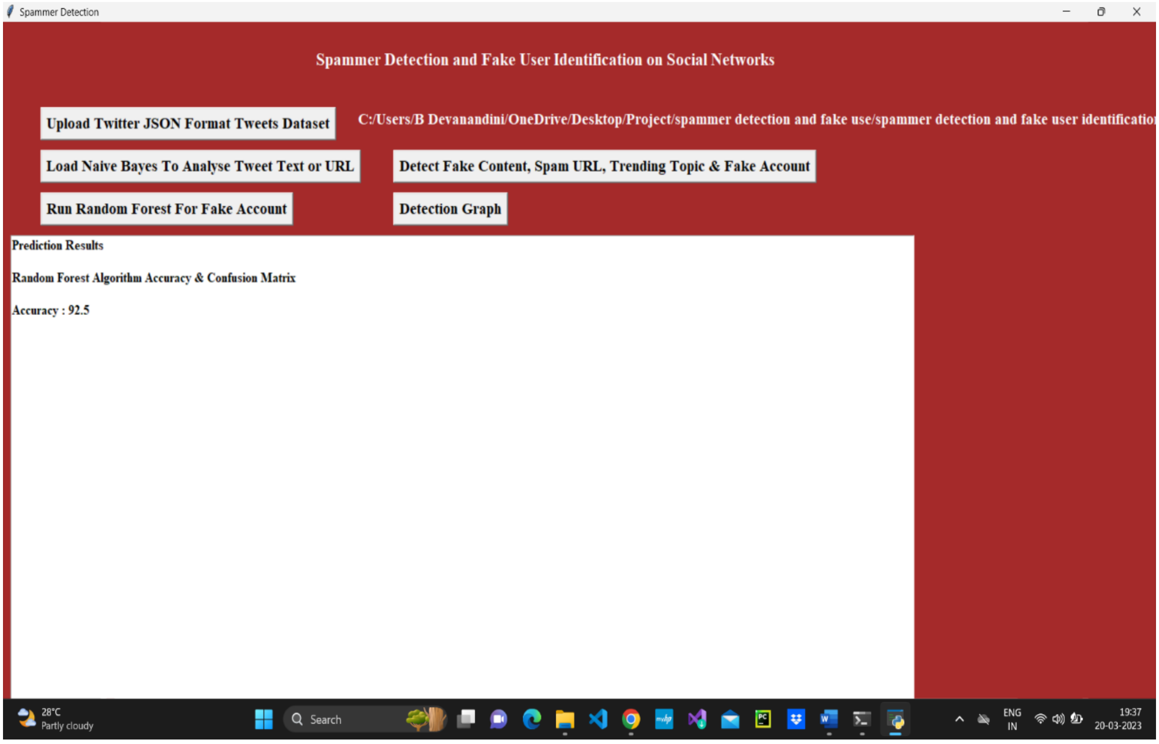
```
Username : EECC506
Tweet Text : RT PrisonPlanet Time for the left to dial back its racist anti white rhetoric in the aftermath of Dallas Retweet Count : 186
Following : 2472
Followers : 2368
Reputation : 122
Hashtag : 49046
Tweet Words Length : 20
Tweet text contains : Spam Words
Twitter Account is Fake

Username : JuGroovy
Tweet Text : RT MattSmethurst Let the record show Dallas https t co 5r2oAH4MD8Retweet Count : 9951
Following : 538
Followers : 555
Reputation : 9
Hashtag : 77121
Tweet Words Length : 11
Tweet text contains : Non-Spam Words
Twiiter Account is Genuine

Username : Larell718
Tweet Text : RT ChrisMurphy67 Its hard to feel good about anything these days Mostly b c many will not acknowledge legit concerns of others Until it Retweet Count : 1
Following : 21726
Followers : 23047
Reputation : 242
Hashtag : 19426
Tweet Words Length : 25
Tweet text contains : Non-Spam Words
```

Figure 6.5 Detecting Spammers of spam detection

In Figure 6.5 as shown above screen all features extracted from tweets dataset and then analyses those features to identify tweets is no spam or spam. In above text area each records values are separated with empty line and each tweet record display values as Tweet Text, Followers, Following etc. with account is fake or genuine and tweet text contains spam or non-spam words. Now click on ‘Run Random Forest Prediction’ button to train random forest classifier with extracted tweets features and this random forest classifier model will be used to predict/detect fake or spam account for upcoming future tweets. Scroll down above text area to view details of each tweet.


Spammer Detection
Spammer Detection and Fake User Identification on Social Networks
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
28°C
Partly cloudy
Search
ENG
IN
19:37
20-03-2023

C:/Users/B Devanandini/OneDrive/Desktop/Project/spammer detection and fake use/spammer detection and fake user identificatio

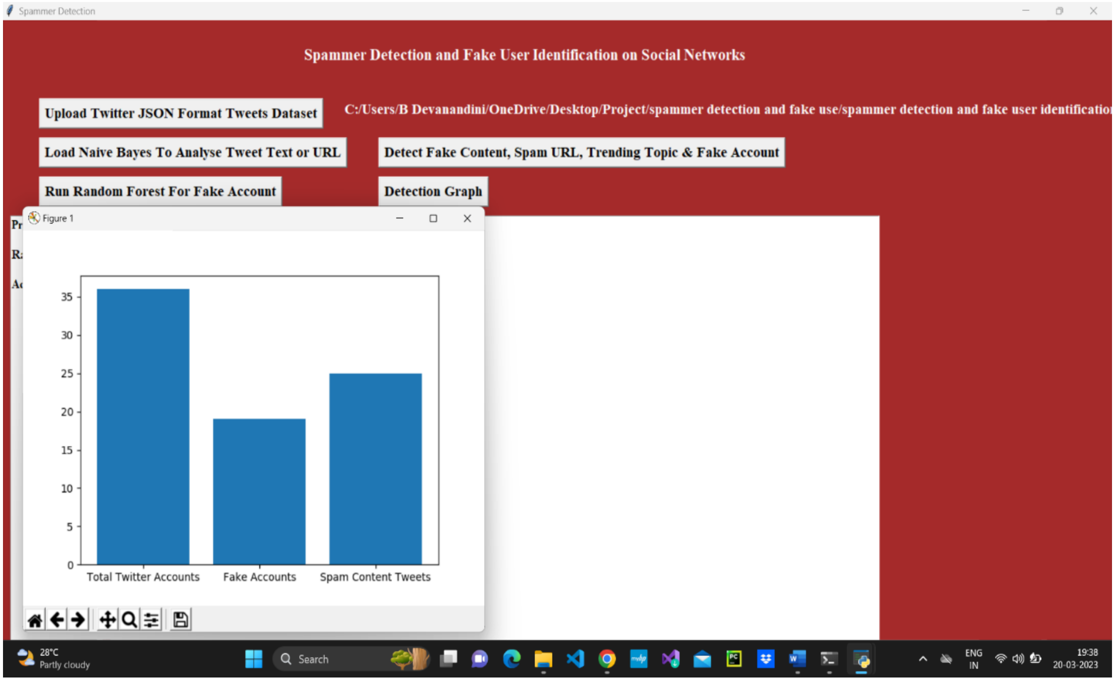
```
Prediction Results

Random Forest Algorithm Accuracy & Confusion Matrix

Accuracy : 92.5
```

Figure 6.6 Accuracy Prediction of spam detection

In Figure 6.6 as shown above screen we got random forest prediction accuracy as 92%, now click on ‘Detection Graph’ button to know total tweets and spam and fake account graph. The results of the study shows that random forest classifier achieves high spam detection accuracy in real-time. The random forest classifier model will be used to predict/detect fake or spam account for upcoming future tweets.


Spammer Detection
Spammer Detection and Fake User Identification on Social Networks
Upload Twitter JSON Format Tweets Dataset
Load Naive Bayes To Analyse Tweet Text or URL
Detect Fake Content, Spam URL, Trending Topic & Fake Account
Run Random Forest For Fake Account
Detection Graph
Figure 1
35
30
25
20
15
10
5
0
Total Twitter Accounts
Fake Accounts
Spam Content Tweets
28°C
Partly cloudy
Search
ENG
IN
19:38
20-03-2023

C:/Users/B Devanandini/OneDrive/Desktop/Project/spammer detection and fake use/spammer detection and fake user identificatio

Figure 6.7 Graph Comparison of spam detection

In Figure 6.7 as shown above graph x-axis represents total tweets, fake account and spam words content tweets and y-axis represents count of them. On the previous screen, we saw that the random forest prediction accuracy was 92%. Now, click on the View Prediction Results i.e., Detection Graph button to see the amount of projected spam and non-spam tweets. The overall graph shows that the number of fake accounts and the number of the Spam content tweets as compared with the total twitter accounts. The number of spam accounts predicted records is 18 and the number of spams content anticipated records is 25.

7 Conclusion and Future Work

Here the project is an implementation of analysis method utilized on behalf of distinguishing spammers on Twitter. We additionally exhibited taxonomy of Twitter spam identification method are considered as false contented recognition, URL built spam identification, spam location in inclining points, and phony client recognition strategies.

7.1 Conclusion

The fake users play a significant role in social media. The Spammer's messages are like non-spam messages. There are several ways to distinguish fake and genuine online media users. However, detecting malicious messages are still a challenging problem. This machine learning based approach to distinguish fake users that could delude individuals. The dataset has been collected and preprocessed. The preprocessing is carried out by the python libraries. The examination model is obtained from the preprocessing. The distinguish matrix is designed by using the examination model. The machine learning is employed in the process. The machine learning algorithm includes Random Forest, Naïve Bayes are utilized for the identification of fake users. The results are recorded and analyzed. The presented method is yielded 92% accuracy in detecting the fake users for the defined environment.

7.2 Future Work

We likewise analyzed the introduced strategies dependent on a few features, for example, client features, content features, chart features, structure features, and time features. Besides, the procedures were likewise looked at regarding their predefined objectives and datasets utilized. It is foreseen that the introduced audit will assist scientists with finding the data on best-in-class Twitter spam discovery procedures in a united structure. Notwithstanding the improvement of proficient and viable methodologies for the spam discovery and phony client distinguishing proof on Twitter, there are yet certain open zones that need extensive consideration by the analysts.

Despite the development of efficient and effective approaches for the spam detection and fake user identification on Twitter, there are still certain open areas that require considerable attention by the researchers. The issues are briefly highlighted as under: False news identification on social media networks is an issue that needs to be explored because of the serious repercussions of such news at individual as well as collective level. Another associated topic that is worth investigating is the identification of rumor sources on social media. Although a few studies based on statistical methods have already been conducted to detect the sources of rumors, more sophisticated approaches, e.g., social network-based approaches, can be applied because of their proven effectiveness.

8 References

[1] B. Ercahin, O. Aktas, D. Kilinc and C. Akyol, “Twitter fake account detection,” *in Proc. Int. Conf. Comput. Sci. Eng,* pp. 388-392, oct 2017.

[2] F. Benevento, G. Magno, T. Rodrigues and V. Almeida, “Detecting spammers on Twitter,” *in Proc. Collaboration,Electron.Messaging,Anti Abuse spam Conf.(CEAS),* vol. 6, p. 12, jUL 2010.

[3] S. Gharge and M. Chavan, “An integrated approach for malicious tweets detection using NLP,” *in Proc. Int. Conf. Inventive Commun. Comput. Technol.(ICICCT),* pp. 435-438, Mar 2017.

[4] T. Wu, S. Wen, Y. Xiang and W. Zhou, “Twitter spam detection: Survey of new approaches and comparative study,” *Comput. Secur,* vol. 76, pp. 265-284, Jul 2018.

[5] S. J.Soman, “A survey on behavior exhibited by spammes in popular social media networks,” *in Proc. Int. Conf. Circuit. Power Comput. Technol.(ICCPCT),* pp. 1-6, Mar 2016.

[6] A. Gupta, H. Lamba and P. Kumaraguru, “1.00 per RT #BostonMarthon #prayforboston:Analyzing fake content on Twitter,” *in Proc. eCrime Researchers Summit (eCRS),* pp. -12, 2013.

[7] F. Concone, A. De Paola, G. Lo Re and M. Morana, “Twitter analysis for real-time malware discovery,” *in Proc. AEIT Int. Annu. Conf.,* pp. 1-6, Sep 2017.

[8] N. Eshrai, M. Jalali and M. H.Moattar, “Detecting spam tweets in Twitter using a data stream clustering algorithm,” *in Proc. Int. Congr. Technol., Commun. Knowl.(ICTCK),* pp. 347-351, Nov 2015.

[9] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou and G. Min, “Statistical features-based real-time detection of drifted Twitter spam,” *IEEE Trans. Inf. Forensics Security,* vol. 12, no. 4, pp. 914-925, Apr 2017.

[10] C. Buntain and J. Golbeck, “Automatically identifying fake news in popular Twitter threads,” *in Proc. IEEE Int. Conf. Smart Cloud(SmartCloud),* pp. 208-215, Nov 2017.