**A MINI PROJECT REPORT**

**On**

**SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORK**

*Submitted by*

|  |  |
| --- | --- |
| N.SNEHA | 20J41A05A0 |
| K.SRAVANI | 20J41A0587 |
| E.PAVAN KUMAR | 20J41A0578 |
| CH.VASU | 20J41A0572 |

*in partial fulfillment of the requirement for the award of degree*

*of*

***Bachelor of Technology***

In

**Computer Science and Engineering**

Under the guidance of

**Mr.B.D.Pavan Kumar**

Designation ,Department

****

Department of Computer Science and Engineering

**MALLA REDDY ENGINEERING COLLEGE**

(An UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Maisammaguda, Secunderabad, Telangana, India 500100

**AUGUST-2023**

**MALLA REDDY ENGINEERING COLLEGE**

Maisammaguda, Secunderabad, Telangana, India 500100



**BONAFIDE CERTIFICATE**

This is to certify that this major project work entitled **“SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS ”,** submitted by **N.SNEHA (20J41A05A0), K.SRAVANI (20J41A0587), E.PAVAN KUMAR (20J41A0578), CH.VASU (20J41A0572)**  to Malla Reddy Engineering College affiliated to JNTUH, Hyderabad in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** is a bonafide record of project work carried out under my/our supervision during the academic year 2023-2024 and that this work has not been submitted elsewhere for a degree.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| Mr. B.D.Pavan Kumar  **SUPERVISOR**  Assistant Professor  Computer Science And Engineering  Malla Reddy Engineering College  Secunderabad, 500 100 | Dr.PSRC Murthy  **HOD**  Computer Science And Engineering  Malla Reddy Engineering College  Secunderabad, 500 100 |
| **Submitted for Major Project viva-voce examination held on \_\_\_\_\_\_\_\_\_** | |
|  |  |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

**MALLA REDDY ENGINEERING COLLEGE**

Maisammaguda, Secunderabad, Telangana, India 50010

**ACKNOWLEDGEMENT**

We would like to express our sincere thanks to **Dr. A. Ramaswami Reddy, Principal, MREC(A)** for providing the working facilities in the college.Our sincere thanks and gratitude to **Dr.PSRC Murthy, Professor & Head**, **Department of CSE, MREC(A)** for all the timely support and valuable suggestions during the period of our project.We are thankful to our Project Coordinator **Mr.K V Vara Prasad, Assistant Professor,**Department of Computer Science and Engineering, for his cooperation during the project work.

We are extremely thankful and indebted to our internal guide, **Mr.B.D.Pavan Kumar, Assistant Professor, Department of CSE, MREC(A)** for her constant guidance, encouragement and moral support throughout the project.

Finally, We would also like to thank all the faculty and staff of the CSE Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

N.Sneha 20J41A05A0

K.Sravani 20J41A0587

E.Pavan Kumar 20J41A0578

CH.Vasu 20J41A0572

**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 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 spam. Fake users send undesired tweets to users to promote services or websites that not only affect legitimate users but also disrupt resource consumption. Moreover, the possibility of expanding invalid information to users through fake identities has increased those 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 paper, 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.

**KEYWORDS:** regression prediction model, machine learning algorithm, naive bayes classifier, hybrid technique

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **ABSTRACT**  **LIST OF TABLES**  **LIST OF FIGURES**  **LIST OF ABBREVIATIONS** | | |  |
| iv |
| v-vii  viii  ix |
| **Chapter** | **Description** | | **Page No.** |
| **1** | **INTRODUCTION** | | **1-10** |
|  | * 1. Introduction | | 1-2 |
|  | 1.2. Spammer Detection in Twitter | | 3 |
|  | 1.2.1 | Fake Content | 3-5 |
|  | 1.2.2 | URL Based Spam Detection | 6 |
|  | 1.2.3 | Detecting Spam in Trending Topics | 7-8 |
|  | 1.2.4 | Fake User Identification | 8-9 |
|  | 1.3. Objective of the project | | 10 |
|  | 1.3.1 | Overview | 10 |
|  | 1.3.2 | Approach | 10 |
| **2** | **LITERATURE REVIEW** | | **11-13** |
|  | 2.1.Existing System | | 11 |
|  | 2.2.Existing System Disadvantages | | 12 |
|  | 2.3. Proposed System | | 12-13 |
|  | 2.4.Proposed System Advantages | | 13 |
| **3** | **SYSTEM STUDY** | | **14-17** |
|  | 3.1. Introduction | | 14 |
|  | 3.1.1. Request Clarification | | 14 |
|  | 3.1.2. Feasibility Analysis | | 14-16 |
|  | 3.1.3 | Request Approval | 16 |
|  | 3.2. Feasibility Study | | 16-17 |
| **4** | **REQUIRED RESOURCES** | | **18-22** |
|  | 4.1. Hardware Requirements | | 18 |
|  | 4.2. Software Requirements | | 18 |
|  | 4.3. Software Environment for Python | | 18-22 |
| **5** | **SYSTEM DESIGN** | | **23-25** |
|  | 5.1. Architecture Diagram | | 23 |
|  | 5.2. Input Design | | 24 |
|  | 5.3. Output Design | | 25 |
| **6** | **FLOW CHARTS AND CLASS DIAGRAM** | | 26-31 |
|  | 6.1. Data Flow Diagram | | 26 |
|  | 6.2. Flow Chart | | 27-28 |
|  | 6.2.1. User | | 27 |
|  | 6.2.2. Admin | | 28 |
|  | 6.2.3. Sequence Diagram | | 29 |
|  | 6.2.4. Use case Diagram | | 30 |
|  | 6.3. Class Diagram | | 31 |
| **7** | **IMPLEMENTATION** | | 32-34 |
|  | 7.1. Modules | | 32 |
|  | 7.1.1 Admin | | 32 |
|  | 7.1.2. User | | 32 |
| **8** | **SYSTEM TESTING** | | 35-41 |
|  | 8.1. Introduction | | 35 |
|  | 8.1.1 Unit Testing | | 35 |
|  | 8.1.2. Integration Testing | | 36 |
|  | 8.1.3. Acceptance Testing | | 36 |
|  | 8.2. Testing Methodologies | | 36-38 |
|  | 8.3. Other Testing Methodologies | | 38-41 |
| **9** | **SCREENSHOTS** | | 42-43 |
|  | 9.1 Pie Chart | | 42 |
|  | 9.2 Line Chart | | 42 |
|  | 9.3 Bar Graph | | 43 |
|  | 9.4 Pie Chart | | 43 |
| **10** | **CONCLUSION** | | 44 |
| **11** | **REFERENCES** | | 45-47 |
|  |  | |  |
|  |  | |  |
|  |  | |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1.1 | Creation of Fake Accounts | 2 |
| 1.2 | Flow Chart of Spammer Detection | 6 |
| 1.3 | Fake User Identification | 10 |
| 2.1 | Identification of Fake networks | 13 |
| 4.1 | Django Diagram | 22 |
| 4.2 | Django Framework | 22 |
| 5.1 | Architecture Diagram | 23 |
| 5.2 | Analysis of user login | 24 |
| 6.1 | Data Flow Diagram | 25 |
| 6.2 | User Flow Diagram | 26 |
| 6.3 | Admin Flow Diagram | 27 |
| 6.4 | Sequence Diagram | 28 |
| 6.5 | Use Case Diagram | 29 |
| 6.6 | Class Diagram | 30 |
| 7.1 | Registration | 33 |
| 7.2 | User Login | 33 |
| 7.3 | Profile Details | 34 |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| OSN | - | Online Social Network |
| HTTP | - | Hyper Text Transfer Protocol |
| UDI | - | Unique Device Identification |
| LDT | - | Learn From Detected Tweet |
| LHL | - | Learn from Human Labelling |
| URL | - | Uniform Resource Locator |
| LAN  HTML  SQL  JSON | -  -  - | Local Access Network  Hypertext Markup Language  Structured Query Language  JavaScript Object Notation |

**CHAPTER 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 intensity. 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 difcult 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.

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 paper 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.



**Fig-1.1** Creation of Fake Accounts

**1.2 SPAMMER DETECTION IN TWITTER**

In this article, we elaborate a classification of spammer detection techniques. Fig. 1 shows the proposed taxonomy for identification of spammers on Twitter. The proposed taxonomy is categorized into four main classes, namely, (i) fake content, (ii) URL based spam detection, (iii) detecting spam in trending topics, and (iv) fake user identification. Each category of identification methods relies on a specific model, technique, and detection algorithm. The first category (fake content) includes various techniques, such as regression prediction model, malware alerting system, and Lfun scheme approach. In the second category (URL based spam detection), the spammer is identified in URL through different machine learning algorithms. The third category (spam in trending topics) is identified through Naïve Bayes classifier and language model divergence. The last category (fake user identification) is based on detecting fake users through hybrid techniques. Techniques related to each of the spammer identification categories are discussed in the following subsections.

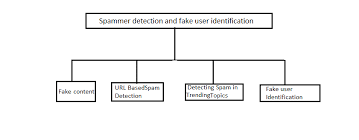
**1.2.1 FAKE CONTENT**

Gupta et al. performed an in-depth characterization of the components that are affected by the rapidly growing malicious content. It was observed that a large number of people with high social profiles were responsible for circulating fake news. To recognize the fake accounts, the authors selected the accounts that were built immediately after the Boston blast and were later banned by Twitter due to violation of terms and conditions. About 7.9 million distinctive tweets were collected by 3.7 million distinctive users. This dataset is known as the largest dataset of Boston blast. The authors performed the fake content categorization through temporal analysis where temporal distribution of tweets is calculated based on the number of tweets posted per hour. 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 also to predict the fake content growth in future.

Concone et al. presented a methodology that provides malignant alerting by using a specified set of tweets in real-time conquered through the Twitter API. Afterwards,the batch of tweets considering the same topic is sum up to generate an alert. The proposed architecture is used to evaluate Twitter posting, recognizing the advancement of admissible event, and reporting of that event itself. The proposed approach utilizes the information contained in the tweets when a spam or malware is recognized by the users or the report of security has been released by the certified authorities. The proposed alerting system comprises of the following components: (i) real time data extraction of both tweets and users, (ii) filtering system based on a preprocessing schedule and on Naïve Bayes algorithm to discard the tweets containing inaccurate information, (iii) data analysis for spammer detection where the detection windows are rigorously abolished according to the Sigmoid function or when the window size reaches the maximum, (iv) alert subsystem that is used when the event is established, the system groups up the tweets that are relevant to the same topic where tweets are distinguished with the cluster barycenter and the one that is nearest to the cluster center is chosen as the representative of the whole system cluster, and (v) feedback analysis. The approach is claimed to be efficient and effective for the detection of some invasive and admirable malignant activities in circulation

Moreover, Eshraqi et al. determined different features to detect the spam and then with the help of a den stream based clustering algorithm, recognize the spam tweets. Some user accounts were selected from various datasets and afterwards random tweets were selected from these accounts. The tweets are subsequently categorized as spam and nonspam. The authors claimed that the algorithm can divide the data into spam and non-spam with high accuracy and fake tweets maybe recognized with high accuracy and precision.

Various features can be used to determine the spams. For example, feature based on the graph is a state in which Twitter is shaped as a social model of a graph. If the number of followers is low in comparison with the number of followings, the credibility of an account is low and the possibility that the account is spam is relatively high. Likewise, feature based on content includes tweets reputation, HTTP links, mentions and replies, and trending topics. For the time feature, if many tweets are sent by a user account in a certain time interval, then it is a spam account. The dataset of the study comprised 50,000 user accounts. The approach identified the spammers and fake tweets with high accuracy. A Lfun (learning for unlabeled tweets) scheme, which is used to handle various problems in the detection of Twitter spam, has been presented by Chen et al. Their framework comprises two components, i.e., learn from detected tweets (LDT) and learn from human labelling (LHL). The two components are used to automatically generate spam tweets from the given set of unmarked tweets that are easily collected from the Twitter network side. Once the labelled spam tweets are obtained, random forest algorithm is used to perform classification. The performance of the scheme is evaluated while detecting drifted spam tweets. The experiments were performed on the real-world data of ten continuous days with each day having 100K tweets each for the spam and non-spam. The F-measure and the detection rate were used to evaluate the performance of the presented scheme. The results of the proposed approach showed that the methodology improves the accuracy of spam detection significantly in the real-world situations. Furthermore, Buntain et al. introduced a method for detecting fake news on Twitter automatically by predicting accurate assessment in two credibility-focused datasets. The method was applied on the Twitter fake news dataset and the model was trained against a crowd sourced worker based on the assessment of journalists. The two Twitter datasets were used to study the integrity in OSNs. The first dataset CREDBANK, a crowd-sourced dataset, was used to evaluate the accuracy of events in Twitter whereas the second dataset called PHEME is a journalist-labelled dataset of possible rumors in Twitter and journalistic evaluation of their accuracy. A total of 45 features were described that fall into four categories: structural feature, user feature, content feature, and temporal features. Aligning labels in PHEME and BUZZFEED contain classes that describe whether a story is fake or true. Results of the analysis are helpful in studying information on social media to know whether such stories support similar pattern.



**Fig-1.2** Flow chart of Spammer Detection

**1.2.2 URL Based Spam Detection**

Chen et al. performed an evaluation of machine learning algorithms to detect spam tweets. The authors analyzed the impact of various features on the performance of spam detection, for example: (i) spam to non-spam ratio, (ii) size of training dataset, (iii) time related data, (iv) factor discretization, and (v) sampling of data. To evaluate the detection, first,around 600 million public tweets were collected and subsequently the authors applied the Trend micro’s web reputation system to identify spam tweets as much as possible. A total of 12 lightweight features were also separated to distinguish non-spam and spam tweets from this identified dataset. The characteristics of identified features were represented by cdf figures. These features are grasped to machine learning based spam classification, which are later used in the experiment to evaluate the detection of spam. Four datasets are sampled to reproduce different scenarios. Since no dataset is available publicly for the task, few datasets were used in previous researches. After the identification of spam tweets, 12 features were gathered. These features are divided into two classes, i.e., user-based features and tweet-based features. The user-based features are identified through various objects such as account age and number of user favorites, lists, and tweets. The identified user-based features are parsed from the JSON structure. On the other hand, the tweet-based features include the number of (i) retweets, (ii) hashtags, (iii) user mentions, and (iv) URLs. The result of evaluation shows that the changing feature distribution reduced the performance whereas no differences were observed in the training dataset distribution.

**1.2.3 Detecting Spam in Trending Topics**

Gharge et al. initiate a method, which is classified on the basis of two new aspects. The first one is the recognition of spam tweets without any prior information about the users and the second one is the exploration of language for spam detection on Twitter trending topic at that time. The system framework includes the following five steps. • The collection of tweets with respect to trending topics on Twitter. After storing the tweets in a particular file format, the tweets are subsequently analyzed. • Labelling of spam is performed to check through all datasets that are available to detect the malignant URL. • Feature extraction separates the characteristics construct based on the language model that uses language as a tool and helps in determining whether the tweets are fake or not. • The classification of data set is performed by shortlisting the set of tweets that is described by the set of features provided to the classifier to instruct the model and to acquire the knowledge for spam detection. • The spam detection uses the classification technique to accept tweets as the input and classify the spam and nonspam. The experimental setup was prepared for determining the accuracy of the system. For this purpose, a random sample set of 1,000 tweets was collected from which 60% were legal and the rest were defected. Stafford et al. examined the degree to which the trending affairs in Twitter are exploited by spammers. Although numerous methods to detect the spam have been proposed, the research on determining the effects of spam on Twitter trending topics has attained only limited attention of the researchers. The authors in presented a technique to cooperate with Twitter public API. The aim of the implemented program was to find 10 trending topics from all over the world having a language code within one hour and open the filtered connection related to those topics to acquire a data stream. In the next hour, the authors obtained as much of the tweets and linked metadata as permitted by the Twitter API. Once the data has been collected, the collected tweets were classified into two categories, i.e., spam and non-spam tweets, which can be utilized to instruct classifiers. To develop such a collection of manual labelling, another program was suggested to sample random tweets, where the idea is based upon URL filtering by Hussain et al.. After the completion of labelling tweets, they move toward the next phase of analysis method. Analysis method has two separate phases, where the first phase was to select and evaluate the attribute through information retrieval metrics, while the second phase was to evaluate the effect of spam filtering on the trending topics through statistical test. The result of the evaluation concludes that spammer does not acquire the trending topic in Twitter but alternatively adopts target topics with required qualities. The results signify well for the sustainability of the Twitter and provide a way for improvement.

**1.2.4 Fake User Identification**

A categorization method is proposed by Erşahin et al. to detect spam accounts on Twitter. The dataset used in the study was collected manually. The classification is performed by analyzing user-name, profile and background image, number of friends and followers, content of tweets, description of account, and number of tweets. The dataset comprised 501 fake and 499 real accounts, where 16 features from the information that were obtained from the Twitter APIs were identified. Two experiments were performed for classifying fake accounts. The first experiment uses the Naïve Bayes learning algorithm on the Twitter dataset including all aspects without discretization, whereas the second experiment uses the Naïve Bayes learning algorithm on the Twitter dataset after the discretization. Mateen et al. proposed a hybrid technique that utilizes user-based, content-based, and graph-based characteristics for spammer profiles detection. A model is proposed to differentiate between the non-spam and spam profiles using three characteristics. The proposed technique was analyzed using Twitter dataset with 11K users and approximately 400K tweets. The goal is to attain higher efficiency and preciseness by integrating all these characteristics. User-based features are established because of relationship and properties of user accounts. It is essential to append user-based features for the spam detection model. As these features are related to user accounts, all attributes, which were linked to user accounts, were identified. These attributes include the number of followers and following, age, FF ratio, and reputation. Alternatively, content features are linked to the tweets that are posted by users as spam bots that post a huge amount of duplicate contents as contrast to non-spammers who do not post duplicate tweets. These features depend on messages or content that users write. Spammers post contents to spread fake news and these contents contain malicious URL to promote their product. The content-based features include: (i) the total number of tweets, (ii) hashtag ratio, (iii) URLs ratio, (iv) mentions ratio, and (v) frequencyof tweets. The graph-based feature is used to control the evasion strategies that are conducted by spammers. Spammers use different techniques to avoid being detected. They can buy fake followers from different thirdparty websites and exchange their followers to another user to look like a legal user. Graph-based features include in/out degree and betweenness. The evaluation of the approach is done by using the dataset of previous techniques as, due to the Twitter policy, no data is available publicly. The results are evaluated by integrating three most common approaches, namely Decorate, Naïve Bayes, and J48. The result of the experiment shows that the detection rate of the approach is much accurate and higher than any of the existing techniques. Gupta et al. present a policy for the detection of spammers in Twitter and use the popular techniques, i.e., Naïve Bayes, clustering, and decision trees. The algorithms classify an account as spam or non-spam. The dataset comprises 1064 Twitter users that contain 62 features, which are either user-specific or tweet-specific information. The spammer account contains almost 36% of the used dataset. As the behavior of spammers is different from non-spammers, some attributes or features are recognized in which both categories are different from one another. Feature identification is based on the number of features at user and tweet level such as followers or following, spam keywords, replies, hashtags, and URLs [30], [32]. After the identification of features, pre-processor step transforms all continuous features into discrete. Subsequently, the authors developed a technique using clustering, decision trees, and Naïve Bayes algorithms. With Naïve Bayes, the accounts were identified by estimating the possibility of certain account as the spammer or non-spammer. In clustering-based algorithm, the entire set of accounts is classified into two classes as the spam and non-spam. In decision tree algorithm, structure of tree was designed, and the decisions were made at every level of the tree. The result of the proposed approach shows that the clustering algorithm’s performance to detect the non-spam accounts is better as compared to detection of spam accounts. Results of these integrated algorithm demonstrate the overall accuracy and detection of non-spammer with high effectiveness.



**Fig-1.3** Fake User Identification

**1.3 OBJECTIVE OF THE PROJECT**

**Overview**

The objective of this project is to develop a data-driven approach for detecting fake users and spammers in social networks using machine learning techniques. The goal is to identify and remove these users in order to improve the overall user experience and reduce the spread of misinformation and spam.

### **Approach**

The approach will involve collecting and analyzing large amounts of data from social networks, including user behavior, network structure, and content. Machine learning algorithms will be used to identify patterns and anomalies that are indicative of fake users and spammers. These algorithms will be trained on labeled data sets to improve their accuracy and effectiveness.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 EXISTING SYSTEM**

* Shen *et al.* [29] investigated issues of detecting spammers on Twitter. The proposed method combines characteristics withdrawal from text content and information of social networks. The authors used matrix factorization to determine the underline feature matrix or the tweets and then came up with a social regularization with interaction coefficient to teach the factorization of the underline matrix. Subsequently, the authors combined knowledge with social regularization and factorization matrix processes, and performed experiments on the real-world Twitter dataset, i.e., UDI Twitter dataset.
* Washha *et al.* [31] described the Hidden Markov Model for filtering the spam related to recent time. The method supports the accessible and obtainable information in the tweet object to recognize spam tweets and the tweets that are handled previously related to the same topic.
* Jeong *et al.* [17] analyzed the follow spam on Twitter as an alternative of dispersion of provoking public messages, spammers follow authorized users, and followed by authorized users. Categorization techniques were proposed that are used for the detection of follow spammers. The focus of the social relation is cascaded and formulated into two mechanism, i.e., social status filtering and trade significance
* profile filtering, where each of which uses two-hop sub networks that are centered at each other. Assemble techniques and cascading filtering are also proposed for combining the properties of both trade significance profile and social status. To check whether a user is fake or not, a two-hop social network for each user is focused to gather social information from social networks.
* Meda *et al.* [21] presented a technique that utilizes a sampling of non-uniform features inside a machine learning system by the adaptation of random forest algorithm to recognize spammer insiders. The proposed framework focuses on the random forest and non-uniform feature sampling techniques. The random forest is a learning algorithm for the categorization and regression that works by assembling several decision trees at preparation time and selecting the one with the majority votes by individual trees. The scheme integrates bootstrap aggregating technique with the un-planned selection of features.

**2.2 EXISTING SYSTEM DISADVANTAGES**

* + There is no filtering system based on a preprocessing schedule and on Naïve Bayes algorithm to discard the tweets containing inaccurate information,.
  + Less security due No URL Based Spam Detection.

**2.3 PROPOSED SYSTEM**

* In the proposed system, the system elaborates a classification of spammer detection techniques. The system shows the proposed taxonomy for identification of spammers on Twitter. The proposed taxonomy is categorized into four main classes, namely, (i) fake content, (ii) URL based spam detection, (iii) detecting spam in trending topics, and (iv) fake user identification. Each category of identification methods relies on a specific model, technique, and detection algorithm.
* The first category (fake content) includes various techniques, such as regression prediction model, malware alerting system, and Lfun scheme approach. In the second category (URL based spam detection), the spammer is identified in URL through different machine learning algorithms. The third category (spam in trending topics) is identified through Naïve Bayes classifier and language model divergence. The last category (fake user identification) is based on detecting fake users through hybrid techniques.



**Fig-2.1** Identification Of Fake Accounts

**2.4 PROPOSED SYSTEM ADVANTANGES**

* The average numbers 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 also to predict the fake content growth in future.

**CHAPTER 3**

**SYSTEM STUDY**

**3.1 INTRODUCTION**

The first and foremost strategy for development of a project starts from the thought of designing a mail enabled platform for a small firm in which it is easy and convenient of sending and receiving messages, there is a search engine ,address book and also including some entertaining games. When it is approved by the organization and our project guide the first activity, ie. preliminary investigation begins. The activity has three parts:

* **Request Clarification**
* **Feasibility Study**
* **Request Approval**

**3.1.1 REQUEST CLARIFICATION**

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant for users within the company whose systems can be interconnected by the Local Area Network(LAN). In today’s busy schedule man need everything should be provided in a readymade manner. So taking into consideration of the vastly use of the net in day to day life, the corresponding development of the portal came into existence.

**3.1.2 FEASIBILITY ANALYSIS**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

* **Operational Feasibility**
* **Economic Feasibility**
* **Technical Feasibility**

###### Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

**Economic Feasibility**

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

###### Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

**3.1.3 REQUEST APPROVAL**

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

**3.2 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER 4**

**REQUIRED RESOURCES**

**SYSTEM REQUIREMENTS**

**4.1 H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**4.2 Software Requirements:**

* Operating System - Windows XP
* Coding Language - Python with Django
* Back End - MySQL

**4.3 SOFTWARE ENVIRONMENT FOR PYTHON**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language), Python has a design philosophy that emphasizes code [readability](https://en.wikipedia.org/wiki/Readability" \o "Readability) (notably using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character" \o "Whitespace character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block" \o "Code block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code" \o "Source lines of code) than might be used in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++)or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)" \o "Java (programming language)). It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system" \o "Operating system). [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation" \o "Reference implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source" \o "Open source) software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation" \o "Python Software Foundation). Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type" \o "Dynamic type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management" \o "Memory management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm" \o "Programming paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming" \o "Imperative programming), [functional](https://en.wikipedia.org/wiki/Functional_programming" \o "Functional programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming" \o "Procedural programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library" \o "Standard library).

**Interactive Mode Programming**

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

$ python

Python 2.4.3 (#1, Nov 11 2010, 13:34:43)

[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>

Type the following text at the Python prompt and press the Enter −

>>> print "Hello, Python!"

If you are running new version of Python, then you would need to use print statement with parenthesis as in print ("Hello, Python!");. However in Python version 2.4.3, this produces the following result −

Hello, Python!

**Script Mode Programming**

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file −

Live Demo

print "Hello, Python!"

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

$ python test.py

This produces the following result −

Hello, Python!

Let us try another way to execute a Python script. Here is the modified test.py file −

Live Demo

#!/usr/bin/python

print "Hello, Python!"

We assume that you have Python interpreter available in /usr/bin directory. Now, try to run this program as follows −

$ chmod +x test.py # This is to make file executable

$./test.py

This produces the following result −

Hello, Python!

**Python Identifiers**

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (\_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in Python.

Here are naming conventions for Python identifiers −

Class names start with an uppercase letter. All other identifiers start with a lowercase letter.

Starting an identifier with a single leading underscore indicates that the identifier is private.

Starting an identifier with two leading underscores indicates a strongly private identifier.

If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

**Reserved Words**

The following list shows the Python keywords. These are reserved words and you cannot use them as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

**Django**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability" \o "Reusability) and "pluggability" of components, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself" \o "Don't repeat yourself). Python is used throughout, even for settings files and data models.



**Fig-4.1** Django

Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete" \o "Create, read, update and delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)" \o "Introspection (computer science)) and configured via admin models



**Fig-4.2** Django Framework

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 ARCHITECTURE DIAGRAM**

Tweet Server

View All Tweets

,View All Spam Details on Reviews

,View All Spam Analysis on Tweets

,View Dislikes Results

,View Likes Results

,View Remote Users

,View All User Reviews

,View All Spam Users

,View All Fake Users

Process all user queries

**Store and retrievals**

**WEB Database**

**Remote User**

Registering the User

Post Your Tweet

,View all Tweet Details and like,dislike and review

,View Trending News

,View Your profile

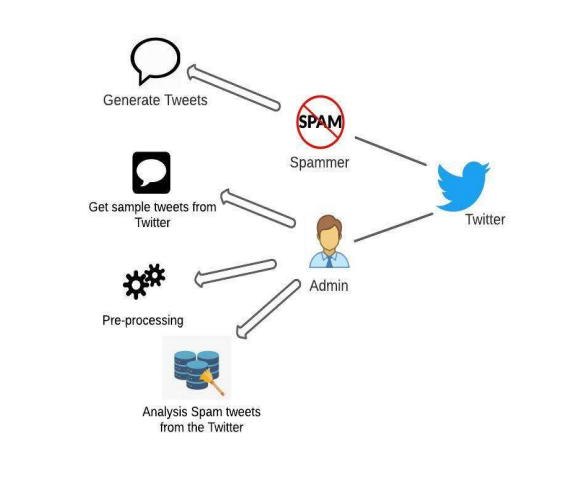
**5.2 INPUT DESIGN**

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.



**Fig-5.2** Analysis of User login

**5.3 OUTPUT DESIGN**

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

**CHAPTER 6**

**FLOW CHARTS AND DIAGRAMS**

**6.1 DATA FLOW DIAGRAM**

**Data Flow Diagram**

View all users and authorize

Web Server

Login

Tweet Server

Response

Request

Response

Request

Create Tweets

Post Your Tweet,View all TweetDetails,View Trending News,View Your profile,View Friend's Retweets and Give Comments

Register and Login

View All Tweets ,View All Spam Details on Reviews,View All Spam Analysis on Tweets,View Dislikes Results,View Likes Results,View Remote Users,View All User Reviews,View All Spam Users,View All Fake Users,View Fake User Identification Results,View Fake Tweet Identification Results

End User

**6.2 FLOW CHART**

* **Flow Chart : User**

User Register

Start

Login

Yes No

Username & Password Wrong

View users Profile

Post Your Tweet

View All Tweets

View Trending News

Logout

View Your profile

View all trending tweets

**6.3 Flow Chart : Admin**

Start

Admin

Login

Yes No

Username & Password Wrong

View All Users Profile

View All Tweets

Logout

View All Spam Details on Reviews

View All Spam Analysis on Tweets

View Dislikes Results,View Likes Results

View Remote Users

View All User Reviews,View All Spam Users,View All Fake Users

**6.4 Sequence Diagram:**

**Server**

User

Tweet Server

List All Users and authorize

Register and Login, View User Profile

View all Tweet Details and like,dislike and review

View All Tweets

View All Spam Details on Reviews

Create Tweet

View All Spammers Detection

View all your tweets and comments  


View All Fake User Identification

View Trending News

View Dislikes Results ,View Likes ResultsView Remote Users,View All User Reviews

,View All Spam Users

,View All Fake Users

View Your profile

**Register and Login**

**6.5 Use case:**

**List Users and authorize**

**like,dislike and review**



**View Trending News**

**Create Tweet**

**User**

**View all your tweets and comments**

**View All User Reviews,View All Spam Users**

**,View All Fake Users,View Fake Tweet Identification Results**

**View all other tweets and make your comment**

**View All Fake User Identification**

**View all your friends and their profile details**

**6.6 Class diagram:**

View All Tweets ,View All Spam Details on Reviews ,View All Spam Analysis on Tweets ,View Dislikes Results ,View Likes Results ,View Remote Users ,View All User Reviews ,View All Spam Users ,ViewAll Fake Users

Tweet Name,TweetDescription,Tweet Uses ,Select Tweet Image,Tweet Owner,Tweet Date and Time

Tweet Server

Methods

Methods

Members

User & Author Login

Remote User

Login, Register, Reset

User Name, Password

Methods

Register, Reset, Search

Name, Password, DOB, Gender, Address, City, Country, Email, Mobile, History, Post

Methods

Members

Members

End User

++

Post Your Tweet,View all Tweet Details,View Trending News,View Your profile

Tweet Name,TweetDescription,Tweet Uses ,Select Tweet Image,Tweet Owner,Tweet Date and Time

Members

**CHAPTER 7**

**IMPLEMENTATION**

**7.1 MODULES**

**7.1.1 ADMIN**

**Admin**

In this module, the Admin has to login by using valid user name and password. After login successful he can do some operations such as View All Tweets,View All Spam Details on Reviews,View All Spam Analysis on Tweets,View Dislikes Results,View Likes Results,View Remote Users,View All User Reviews,View All Spam Users,View All Fake Users

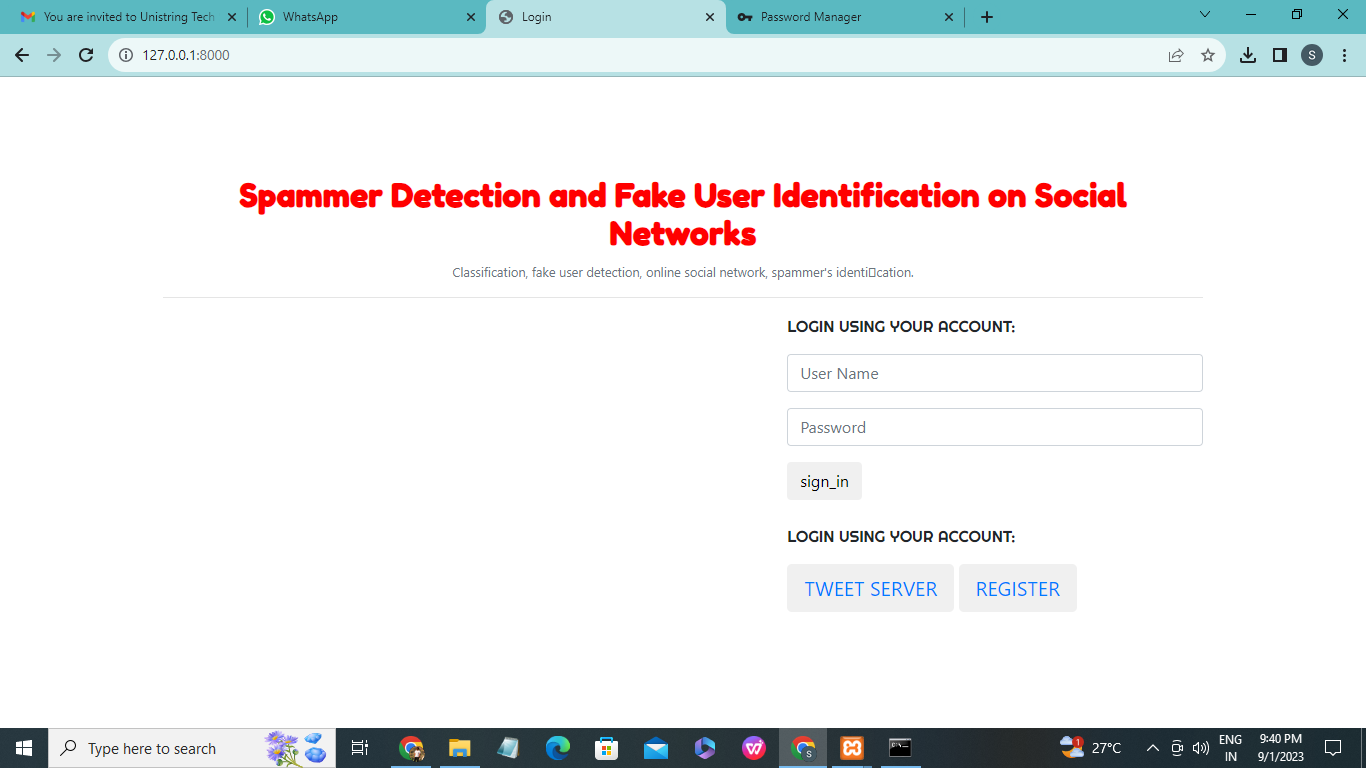
**7.1.2 USER**

* **User**

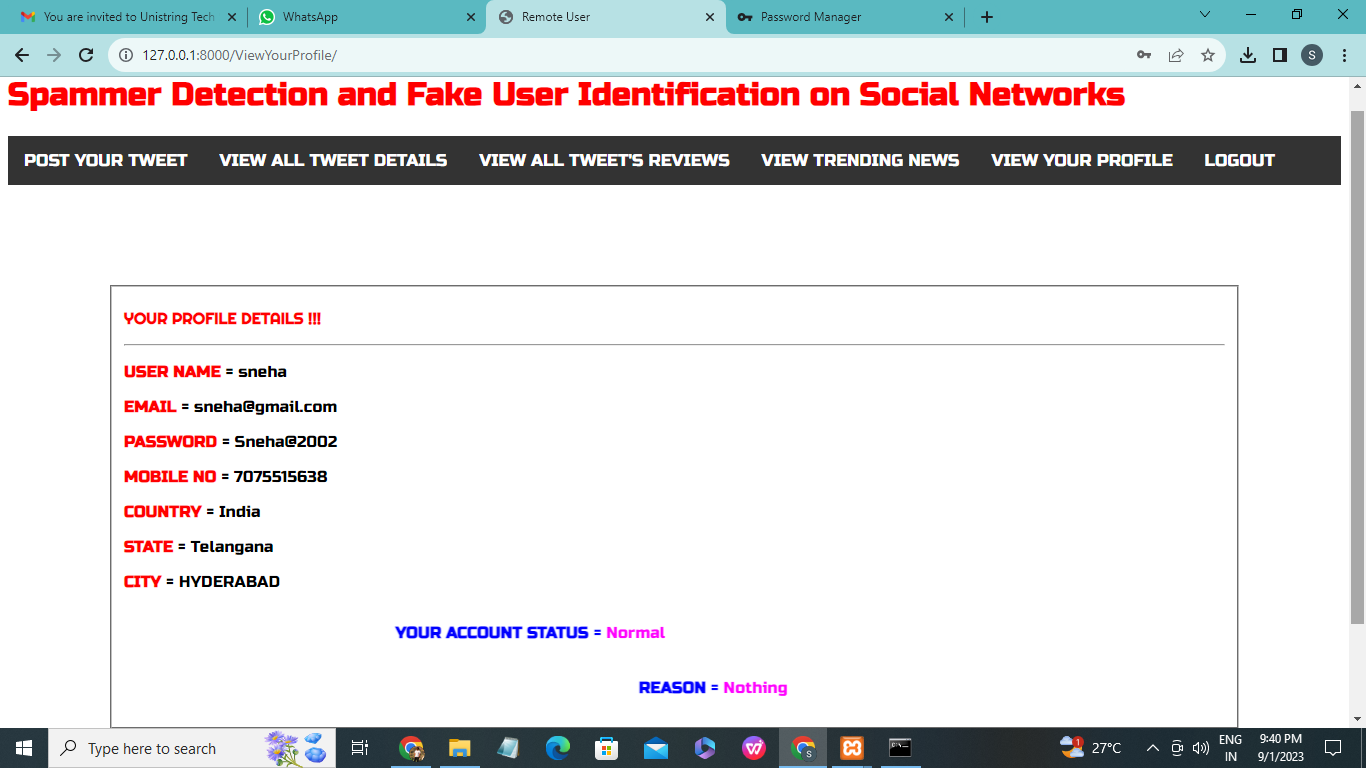
In this module, there are n numbers of users are present. User should register before doing some operations. After registration successful he has to wait for admin to authorize him and after admin authorized him. He can login by using authorized user name and password. Login successful he will do some operations like Post Your Tweet,View all Tweet Details and like,dislike and review,View Trending News,View Your profile

****

**Fig-7.1** Registration

****

**Fig-7.2** User login

****

**Fig-7.3** Profile Details

**CHAPTER 8**

**SYSTEM TESTING**

**8.1 INTRODUCTION**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.1.1 UNIT TESTING**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**8.1.2 INTEGRATION TESTING**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**8.1.3 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**8.2 TESTING METHODOLOGY**

The following are the Testing Methodologies:

* **Unit Testing.**
* **Integration Testing.**
* **User Acceptance Testing.**
* **Output Testing.**
* **Validation Testing.**

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

**Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**The following are the types of Integration Testing:**

**1)Top Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

**2. Bottom-up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

* The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
* A driver (i.e.) the control program for testing is written to coordinate test case input and output.
* The cluster is tested.
* Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

**8.3 OTHER TESTING METHODOLOGIES**

**User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**Validation Checking**

Validation checks are performed on the following fields.

**Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

**USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

**MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

**TESTING STRATEGY :**

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

**UNIT TESTING:**

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.The future holds a lot to offer to the development and refinement of this project.

**CHAPTER 9**

**SCREENSHOTS**

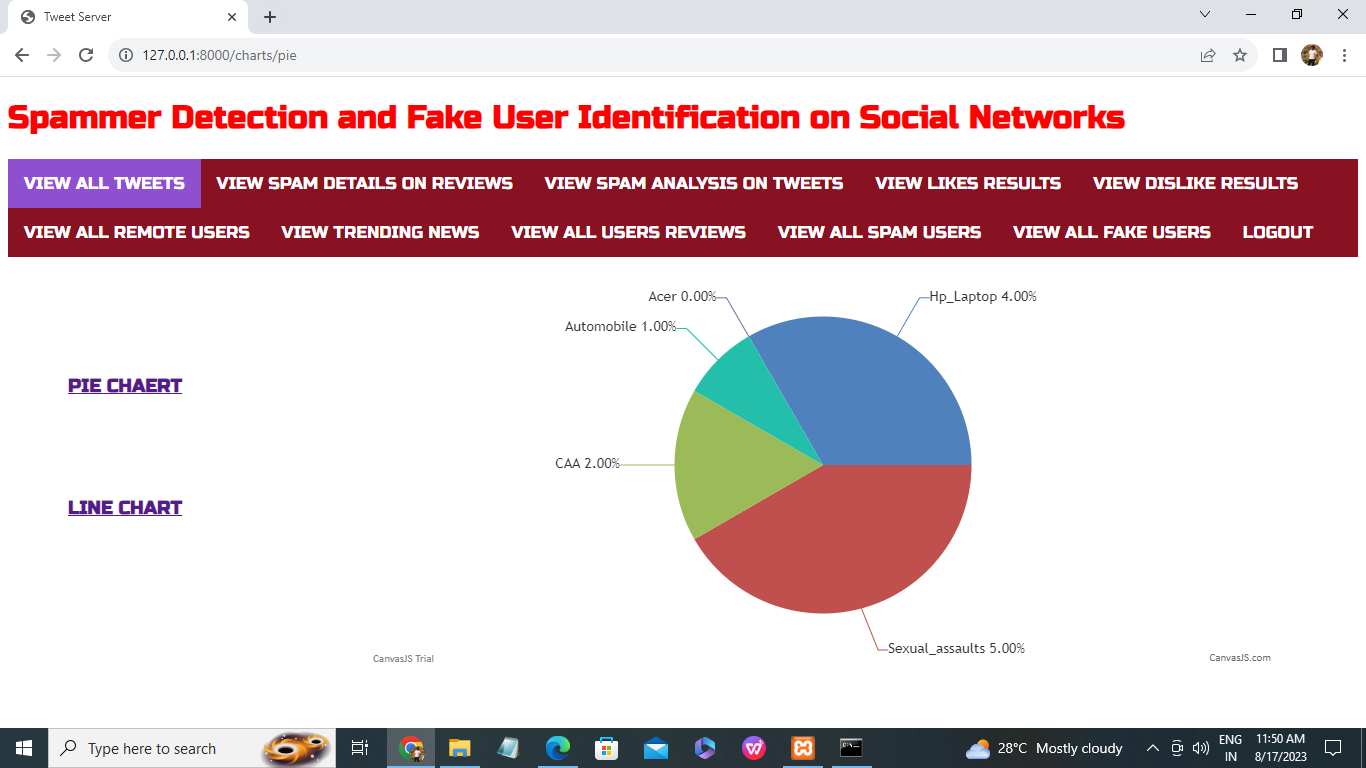
****

Fig9.1 Pie Chart

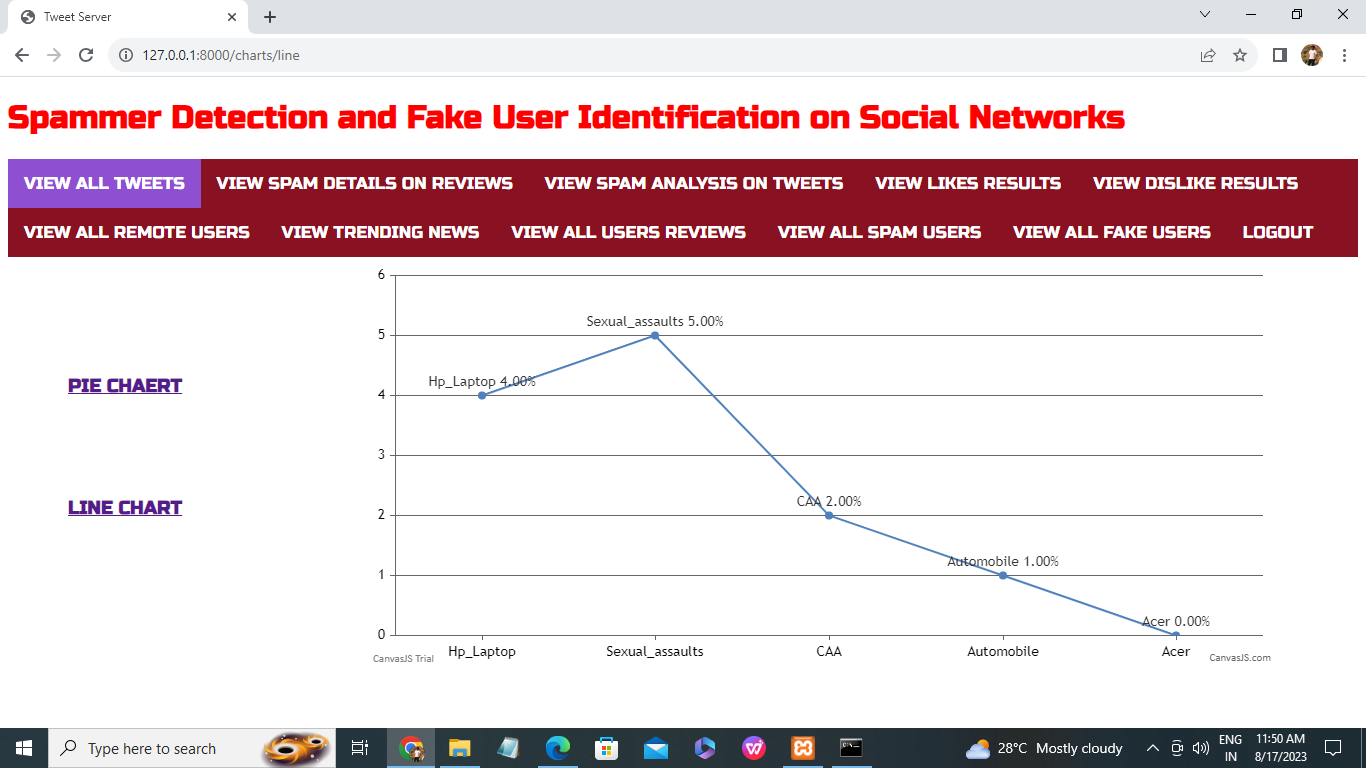
****

Fig-9.2 Line Chart

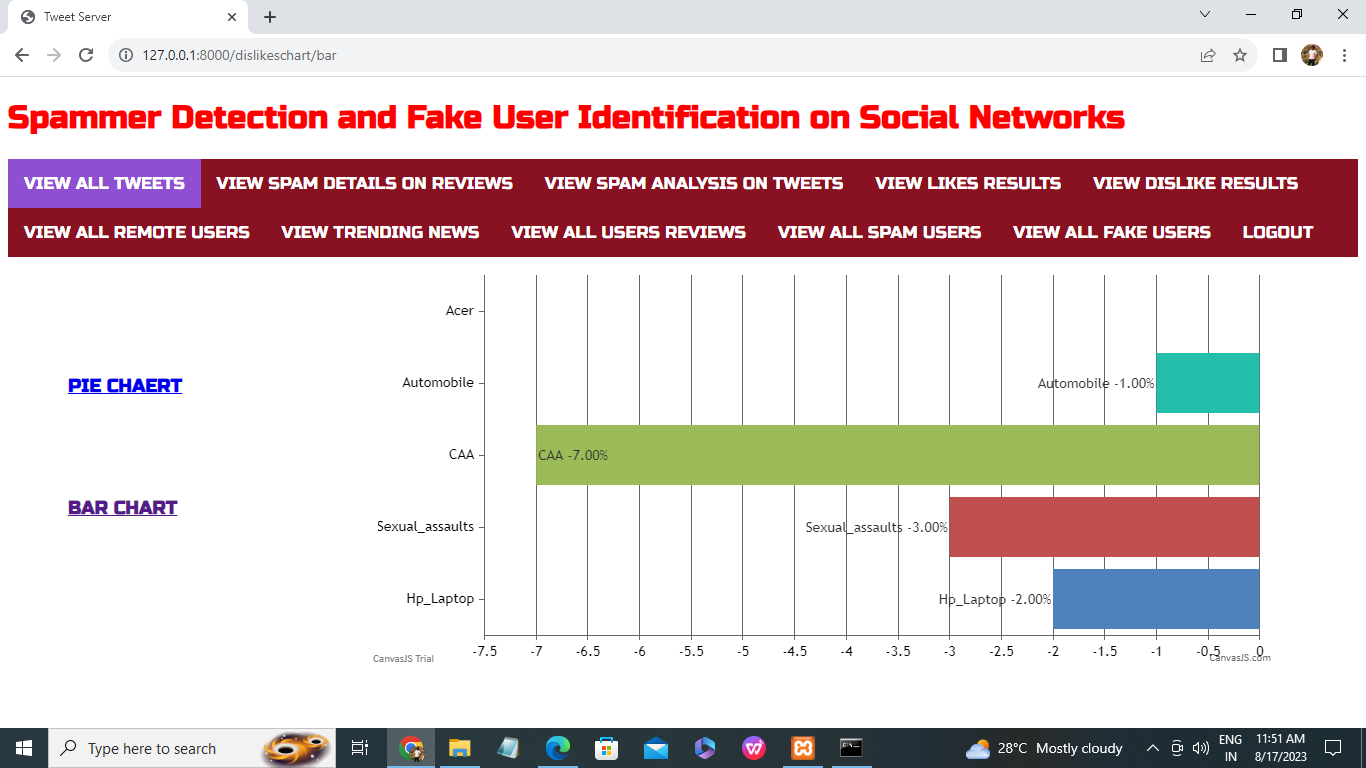
****

Fig-9.3 Bar Chart

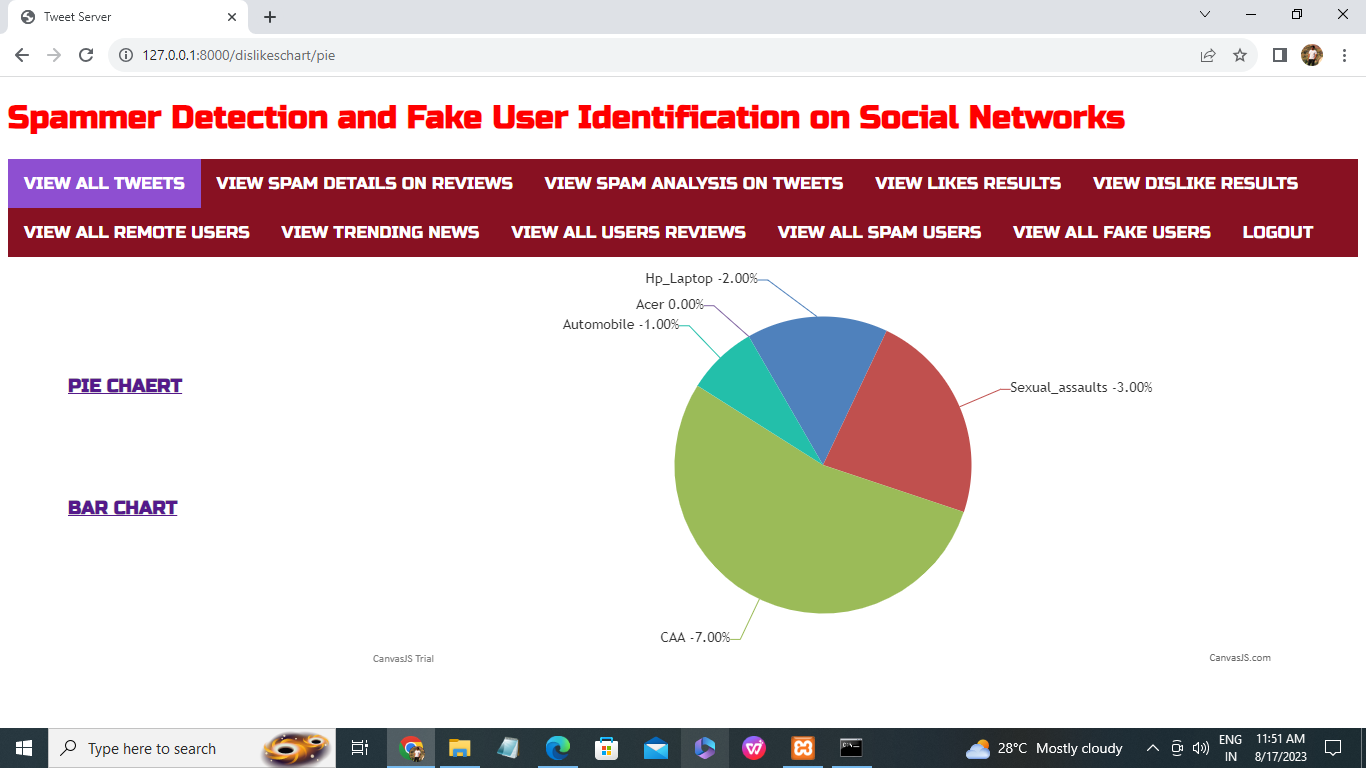
****

Fig-9.4 Pie Chart

**CHAPTER 10**

**CONCLUSION**

In this paper, we performed a review of techniques used for detecting spammers on Twitter. In addition, we also presented a taxonomy of Twitter spam detection approaches and categorized them as fake content detection, URL based spam detection, spam detection in trending topics, and fake user detection techniques. We also compared the presented techniques based on several features, such as user features, content features, graph features, structure features, and time features. Moreover, the techniques were also compared in terms of their specified goals and datasets used. It is anticipated that the presented review will help researchers find the information on state-of-the-art Twitter spam detection techniques in a consolidated form.

Despite the development of efficient and effective approaches for the spam detection and fake user identification on Twitter [34], there are still certain open areas that require considerable attention by the researchers. The issues are

briery 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 [25]. 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.

**CHAPTER 11**

**REFERENCES**

[1] B. Erçahin, Ö. Akta³, D. Kilinç, and C. Akyol, ``Twitter fake account

detection,'' in Proc. Int. Conf. Comput. Sci. Eng. (UBMK), Oct. 2017,

pp. 388392.

[2] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, ``Detecting

spammers on Twitter,'' in Proc. Collaboration, Electron. Messaging, Anti-

Abuse Spam Conf. (CEAS), vol. 6, Jul. 2010, p. 12.

[3] S. Gharge, and M. Chavan, ``An integrated approach for malicious tweets

detection using NLP,'' in Proc. Int. Conf. Inventive Commun. Comput.

Technol. (ICICCT), Mar. 2017, pp. 435438.

[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. 265284, Jul. 2018.

[5] S. J. Soman, ``A survey on behaviors exhibited by spammers in popular

social media networks,'' in Proc. Int. Conf. Circuit, Power Comput. Tech-

nol. (ICCPCT), Mar. 2016, pp. 16.

[6] A. Gupta, H. Lamba, and P. Kumaraguru, ``1.00 per RT #BostonMarathon

# prayforboston: Analyzing fake content on Twitter,'' in Proc. eCrime

Researchers Summit (eCRS), 2013, pp. 112.

[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., Sep. 2017,

pp. 16.

[8] N. Eshraqi, 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), Nov. 2015, pp. 347351.

[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. 914925, Apr. 2017.

[10] C. Buntain and J. Golbeck, ``Automatically identifying fake news in popular

Twitter threads,'' in Proc. IEEE Int. Conf. Smart Cloud (SmartCloud),

Nov. 2017, pp. 208215.

[11] C. Chen, J. Zhang, Y. Xie, Y. Xiang,W. Zhou, M. M. Hassan, A. AlElaiwi,

and M. Alrubaian, ``A performance evaluation of machine learning-based

streaming spam tweets detection,'' IEEE Trans. Comput. Social Syst.,

vol. 2, no. 3, pp. 6576, Sep. 2015.

[12] G. Stafford and L. L. Yu, ``An evaluation of the effect of spam on

Twitter trending topics,'' in Proc. Int. Conf. Social Comput., Sep. 2013,

pp. 373378.

[13] M. Mateen, M. A. Iqbal, M. Aleem, and M. A. Islam, ``A hybrid approach

for spam detection for Twitter,'' in Proc. 14th Int. Bhurban Conf. Appl. Sci.

Technol. (IBCAST), Jan. 2017, pp. 466471.

[14] A. Gupta and R. Kaushal, ``Improving spam detection in online social networks,''

in Proc. Int. Conf. Cogn. Comput. Inf. Process. (CCIP), Mar. 2015,

pp. 16.

[15] F. Fathaliani and M. Bouguessa, ``A model-based approach for identifying

spammers in social networks,'' in Proc. IEEE Int. Conf. Data Sci. Adv.

Anal. (DSAA), Oct. 2015, pp. 19.

[16] V. Chauhan, A. Pilaniya, V. Middha, A. Gupta, U. Bana, B. R. Prasad,

and S. Agarwal, ``Anomalous behavior detection in social networking,''

in Proc. 8th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT),

Jul. 2017, pp. 15.

[17] S. Jeong, G. Noh, H. Oh, and C.-K. Kim, ``Follow spam detection based on

cascaded social information,'' Inf. Sci., vol. 369, pp. 481499, Nov. 2016.

[18] M. Washha, A. Qaroush, and F. Sedes, ``Leveraging time for spammers

detection on Twitter,'' in Proc. 8th Int. Conf. Manage. Digit. EcoSyst.,

Nov. 2016, pp. 109116.

[19] B. Wang, A. Zubiaga, M. Liakata, and R. Procter, ``Making the most

of tweet-inherent features for social spam detection on Twitter,'' 2015,

arXiv:1503.07405. [Online]. Available: https://arxiv.org/abs/1503.07405

[20] M. Hussain, M. Ahmed, H. A. Khattak, M. Imran, A. Khan, S. Din,

A. Ahmad, G. Jeon, and A. G. Reddy, ``Towards ontology-based multilingual

URL ltering: A big data problem,'' J. Supercomput., vol. 74, no. 10,

pp. 50035021, Oct. 2018.

[21] C. Meda, E. Ragusa, C. Gianoglio, R. Zunino, A. Ottaviano, E. Scillia, and

R. Surlinelli, ``Spam detection of Twitter trafc: A framework based on

random forests and non-uniform feature sampling,'' in Proc. IEEE/ACM

Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2016,

pp. 811817.