

SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering in Computer Science Engineering

By

Katta Mrunalini (37110336)

Gali Mary Sasi deepika (37110218)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600 119

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Katta Mrunalini(37110336)** and **Gali Mary Sasi Deepika(37110218)** who carried out the project entitled "**SPAMMER DETECTION AND FAKE USER IDENTIFICATION ON SOCIAL NETWORKS**" under my supervision from **August 2020 to March 2021**.

Internal Guide

Dr. S. Vigneshwari, M.E., Ph.D.,

Head of the Department

Dr. L. Lakshmanan, M.E., Ph.D.,

Dr. S. Vigneshwari, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Katta Mrunalini (37110336)**, **Gali Mary Sasi deepika(37110218)** hereby declare that the Project Report entitled “**SPAMMER DETECTION AND FAKE USE IDENTIFICATION ON SOCIAL NETWORKS**” is done by me under the guidance of **Dr. S. VIGNESHWARI**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S.Vigneshwari, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

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 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 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.

TABLE OF CONTENTS

ABSTRACT

LIST OF FIGURES

LIST OF ABBREVIATIONS

CHAPTER NO	TITLE	PAGE NO
1.	INTRODUCTION	1
	1.1 DESCRIPTION OF THE PROJECT	1-4
	1.2 SCOPE OF THE PROJECT	5
	1.3 PROBLEM STATEMENT	5
2.	LITERATURE SURVEY	6
	2.1 LITERARY REVIEWS	6-11
3.	SYSTEM DESIGN AND REQUIREMENTS	12
	3.1 SYSTEM ANALYSIS	12
	3.2 EXISTING SYSTEM	12
	3.3 PROPOSED SYSTEM	13
	3.4 SYSTEM REQUIREMENTS	14
	3.5 SYSTEM REQUIREMENT SPECIFICATIONS	14-15
4.	METHODOLOGY	16
	4.1 SYSTEM ARCHITECTURE	16
	4.2 MODULE DESCRIPTION	16-17
	4.3 MACHINE LEARNING TECHNIQUE	17-18
	4.4 SYSTEM DESIGN	18-20
	4.5 CLASSES AND DESIGNS FOR THE SYSTEM	20
	4.5.1 USE CASE DIAGRAM	20
	4.5.2 SEQUENCE DIAGRAM	21
	4.6 INPUT AND OUTPUT DESIGN	22-23

	4.7 SOFTWARE ENVIRONMENT	23
	4.8 SYSTEM STUDY	24-31
	4.9 SYSTEM TESTING	32-33
	4.9.1 TYPES OF TESTING	33-34
	4.9.2 TEST STRATEGY AND APPROACH	34-36
5.	RESULTS AND DISCUSSION	37-46
6.	CONCLUSION	47

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
5.1	Path of source code	37
5.2	Login page	38
5.3	Module 1: Admin login	38
5.4	Module 2: Data Collection	39
5.5	Data Collection	39
5.6	Testing data in the background	40
5.7	Module 3: Testing Data	41
5.8	User Check	41
5.9	Enter user details	42
5.10	Details of the user are entered for verification	43
5.11	Details of the user are entered for verification	44
5.12	Detecting spam	44
5.13	Enter the user details	45
5.14	System verification of the user	45
5.15	Genuine user	46

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
OSN	Online Social Networks
RF	Random Forest
URL	Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 SOCIAL NETWORKS

What Is A Social Network?

Wikipedia defines a social network service as a service which “focuses on the building and verifying of online social networks for communities of people who share interests and activities, or who are interested in exploring the interests and activities of others, and which necessitates the use of software.”

A report published by OCLC provides the following definition of social networking sites: “Web sites primarily designed to facilitate interaction between users who share interests, attitudes and activities, such as Facebook, Mixi and MySpace.”

What Can Social Networks Be Used For?

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

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.

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

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).

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.

Common interface: A possible benefit of social networks may be the common interface which spans work / social boundaries. Since such services are often used in a personal capacity the interface and the way the service works may be familiar, thus minimising training and support needed to exploit the services in a professional context. This can, however, also be a barrier to those who wish to have strict boundaries between work and social activities.

Examples of Social Networking Services:

Examples of popular social networking services include:

Facebook: Facebook is a social networking Web site that allows people to communicate with their friends and exchange information. In May 2007 Facebook launched the Facebook Platform which provides a framework for developers to create applications that interact with core Facebook features

MySpace: MySpace is a social networking Web site offering an interactive, user-submitted network of friends, personal profiles, blogs and groups, commonly used for sharing photos, music and videos.

Ning: An online platform for creating social Web sites and social networks aimed at users who want to create networks around specific interests or have limited technical skills.

Twitter: Twitter is an example of a micro-blogging service. Twitter can be used in a variety of ways including sharing brief information with users and providing support for one's peers.

Note that this brief list of popular social networking services omits popular social sharing services such as Flickr and YouTube.

Opportunities and Challenges

The popularity and ease of use of social networking services have excited institutions with their potential in a variety of areas. However effective use of social networking services poses a number of challenges for institutions including long-term sustainability of the services; user concerns over use of social tools in a work or study context; a variety of technical issues and legal issues such as copyright, privacy, accessibility; etc.

Institutions would be advised to consider carefully the implications before promoting significant use of such services.

What is Secure Computing?

Computer security (Also known as cyber security or IT Security) is information security as applied to computers and networks. The field covers all the processes and mechanisms by which computer-based equipment, information and services are protected from unintended or unauthorized access, change or destruction. Computer

security also includes protection from unplanned events and natural disasters. Otherwise, in the computer industry, the term security -- or the phrase computer security -- refers to techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization. Most computer security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. A password is a secret word or phrase that gives a user access to a particular program or system.

Working conditions and basic needs in the secure computing:

If you don't take basic steps to protect your work computer, you put it and all the information on it at risk. You can potentially compromise the operation of other computers on your organization's network, or even the functioning of the network as a whole.

- **Physical security:**

Technical measures like login passwords, anti-virus are essential. (More about those below) However, a secure physical space is the first and more important line of defense.

Is the place you keep your workplace computer secure enough to prevent theft or access to it while you are away? While the Security Department provides coverage across the Medical center, it only takes seconds to steal a computer, particularly a portable device like a laptop or a PDA. A computer should be secured like any other valuable possession when you are not present.

Human threats are not the only concern. Computers can be compromised by environmental mishaps (e.g., water, coffee) or physical trauma. Make sure the physical location of your computer takes account of those risks as well.

- **Access passwords:**

The University's networks and shared information systems are protected in part by login credentials (user-IDs and passwords). Access passwords are also an essential protection for personal computers in most circumstances. Offices are usually open and shared spaces, so physical access to computers cannot be completely controlled.

To protect your computer, you should consider setting passwords for particularly sensitive applications resident on the computer (e.g., data analysis software), if the software provides that capability.

- **Prying eye protection:**

Because we deal with all facets of clinical, research, educational and administrative data here on the medical campus, it is important to do everything possible to minimize exposure of data to unauthorized individuals.

- **Anti-virus software:**

Up-to-date, properly configured anti-virus software is essential. While we have server-side anti-virus software on our network computers, you still need it on the client side (your computer).

- **Firewalls:**

Anti-virus products inspect files on your computer and in email. Firewall software and hardware monitor communications between your computer and the outside world. That is essential for any networked computer.

- **Software updates:**

It is critical to keep software up to date, especially the operating system, anti-virus and anti-spyware, email and browser software. The newest versions will contain fixes for discovered vulnerabilities.

Almost all anti-virus have automatic update features (including SAV). Keeping the "signatures" (digital patterns) of malicious software detectors up-to-date is essential for these products to be effective.

- **Keep secure backups:**

Even if you take all these security steps, bad things can still happen. Be prepared for the worst by making backup copies of critical data, and keeping those backup copies in a separate, secure location. For example, use supplemental hard drives, CDs/DVDs, or flash drives to store critical, hard-to-replace data.

- **Report problems:**

If you believe that your computer or any data on it has been compromised, you should make an information security incident report. That is required by University policy for

all data on our systems, and legally required for health, education, financial and any other kind of record containing identifiable personal information.

1.2 SCOPE OF THE PROJECT

- In this project, we came up with a framework through which we can detect fake profile using machine learning algorithms so that the social life of people become secured.
- In today's online social networks there has been a lot of problems like fake profile, online impersonation, etc. To date, no one has come up with a feasible solution to these problems.
- In this project, we intend to give a framework with which the automatic detection of fake profiles can be done so that the social life of people become secured and by using this automatic detection technique we can make it easier for the sites to manage the huge number of profiles, which can't be done manually.

1.3 PROBLEM STATEMENT

Fake 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.

CHAPTER 2

LITERATURE SURVEY

[1] F. Benevenuto, G. Magno,” Detecting spammers on Twitter”, 2018.

In this paper 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.

Advantages: Manual classification is done, less number of non spammers are misclassified

Disadvantages: Less accuracy for detecting spammers

[2] T. Wu, S. Wen, Y. Xiang, and W. Zhou,” Twitter spam detection: Survey of new approaches and comparative study”, 2018.

Twitter spam has long been a critical but difficult problem to be addressed. So far, researchers have proposed many detection and defence 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 summarise the unsolved challenges in current Twitter spam detection techniques. Solutions to these open issues are of great significance to both academia and industries.

Advantages: Lot of studies and researches are made for developing new methods

Disadvantages: Implementation is not made

[3] B. Erçahin, Ö. Aktaş, D. Kiliç, and C. Akyol," Twitter fake account detection", Oct. 2017.

Social networking sites such as Twitter and Facebook attracts 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 realworld to the society. In our study, we present a classification method for detecting the fake accounts on Twitter.

Advantages: Used EMD which improves performance

Disadvantages: Only confined to twitter, lacks accuracy

[4] S. Gharge, and M. Chavan,"An integrated approach for malicious tweets detection using NLP", 2017.

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 on the basis of 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.

Advantages: Focuses mainly on analysing the tweets which improves accuracy in analysis

Disadvantages: Not focuses mainly on users which might reflect not so well for classification

[5] F. Concone, A. De Paola, G. Lo Re, and M. Morana,” Twitter analysis for real-time malware discovery”, 2017.

In recent years, the increasing number of cyberattacks has gained the development of innovative tools to quickly detect new threats. A recent approach to this problem is to analyze the content of Social Networks to discover the rising of new malicious software. Twitter is a popular social network which allows millions of users to share their opinions on what happens all over the world. The subscribers can insert messages, called tweet, that are usually related to international news. In this work, we present a system for real-time malware alerting using a set of tweets captured through the Twitter API's, and analyzed by means of a Bayes Naïve classifier. Then, groups of tweets discussing the same topic, e.g, a new malware infection, are summarized in order to produce an alert.

Advantages: Tests have been performed to evaluate the performance of the system and results show the effectiveness

Disadvantages: uses same approach, Accuracy can be improved

[6] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min, “Statistical features-based real-time detection of drifted Twitter spam”, 2017.

Twitter spam has become a critical problem nowadays. Recent works focus on applying machine learning techniques for Twitter spam detection, which make use of the statistical features of tweets. In our labeled tweets data set, however, we observe that

the statistical properties of spam tweets vary over time, and thus, the performance of existing machine learning-based classifiers decreases. This issue is referred to as “Twitter Spam Drift”. In order to tackle this problem, we first carry out a deep analysis on the statistical features of one million spam tweets and one million non-spam tweets, and then propose a novel Lfun scheme. The proposed scheme can discover “changed” spam tweets from unlabeled tweets and incorporate them into classifier’s training process. A number of experiments are performed to evaluate the proposed scheme. The results show that our proposed Lfun scheme can significantly improve the spam detection accuracy in real-world scenarios.

Advantages: Works on real time detection helps for immediate response

Disadvantages: Hasn’t implemented yet

[7] Jinyuan Ji, Binghui Wang, Neil Zhenqiang Gong, “Random Walk Based Fake Account Detection in Online Social Networks”, 2017.

Online social networks are known to be vulnerable to the so-called Sybil attack, in which an attacker maintains massive fake accounts (also called Sybils) and uses them to perform various malicious activities. Therefore, Sybil detection is a fundamental security research problem in online social networks. Random walk based methods, which leverage the structure of an online social network to distribute reputation scores for users, have been demonstrated to be promising in certain real-world online social networks. In particular, random walk based methods have three desired features: they can have theoretically guaranteed performance for online social networks that have the fast-mixing property, they are accurate when the social network has strong homophily property, and they can be scalable to large-scale online social networks. However, existing random walk based methods suffer from several key limitations: (1) they can only leverage either labeled benign users or labeled Sybils, but not both, (2) they have limited detection accuracy for weak-homophily social networks, and (3) they are not robust to label noise in the training dataset. In this work, we propose a new random walk based Sybil detection method called SybilWalk. SybilWalk addresses the limitations of existing random walk based methods while maintaining their desired features. We perform both theoretical and empirical evaluations to compare SybilWalk with previous random walk based methods. Theoretically, for online social networks

with the fast-mixing property, SybilWalk has a tighter asymptotical bound on the number of Sybils that are falsely accepted into the social network than all existing random walk based methods. Empirically, we compare SybilWalk with previous random walk based methods using both social networks with synthesized Sybils and a large-scale Twitter dataset with real Sybils.

Advantages: online social networks that have the fast mixing property.

Disadvantages: They have limited detection accuracy.

[8] C. Buntain and J. Golbeck, "Automatically identifying fake news in popular Twitter threads", 2017.

Information quality in social media is an increasingly important issue, but web-scale data hinders experts' ability to assess and correct much of the inaccurate content, or "fake news," present in these platforms. This paper develops a method for automating fake news detection on Twitter by learning to predict accuracy assessments in two credibility-focused Twitter datasets: CREDBANK, a crowdsourced dataset of accuracy assessments for events in Twitter, and PHEME, a dataset of potential rumors in Twitter and journalistic assessments of their accuracies. We apply this method to Twitter content sourced from BuzzFeed's fake news dataset and show models trained against crowdsourced workers outperform models based on journalists' assessment and models trained on a pooled dataset of both crowdsourced workers and journalists. All three datasets, aligned into a uniform format, are also publicly available. A feature analysis then identifies features that are most predictive for crowdsourced and journalistic accuracy assessments, results of which are consistent with prior work. We close with a discussion contrasting accuracy and credibility and why models of nonexperts outperform models of journalists for fake news detection in twitter

Advantages: Automatically identifies without wasting much time

Disadvantages: Identifies spam news in twitter but not fake users

[9] S. J. Soman, "A survey on behaviours exhibited by spammers in popular social media networks", 2016.

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

Advantages: This survey analysis the user behavior which helps finding the spammers

Disadvantages: Implementation is not yet done

[10] A. Gupta, H. Lamba, and P. Kumaraguru,” Analyzing fake content On Twitter”,2013.

In today’s world, online social media plays a vital role during real world events, especially crisis events. Malicious content is posted online during these events, which can result in damage, chaos and monetary loss in the offline world. In our paper, we highlight the role of Twitter in two major crisis events: Hurricane Sandy and Boston Marathon Bombings in spreading fake content about the events. We performed a characterization analysis, to understand the temporal, social reputation and influence patterns for the spread of such fake information. Our results indicate that automated techniques can be used to identify characteristics of fake information on Twitter

Advantages: Analysis is made specifically for contents in twitter

Disadvantages: Doesn’t identify fake users rather analyses the content

CHAPTER-3

SYSTEM DESIGN AND REQUIRMENTS

3.1 SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays a key role in system analysis which gives the target for design and development. In systems engineering and software engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

3.2 EXISTING SYSTEM

In the recent era of technology as the applications and the utilizations increase on our daily life, we continuously posting some unwanted and unaware stuff in the social networking and creating mess in the social platform. In the issue we can consider fake accounts and the people with fake accounts using this platform for their money and how we can stop in this kind of serious situation. Let's consider few social networking platforms for our research work. First is twitter. As we now that twitter mining is the well known application of research and we can have the public dataset of the twitter mining. In this approach we need to get the information about the people who are posting some unwanted things and they need be caught and have to remove those before it creates problem. The twitter account will ask some email ID or the phone number to create an account and for this kind of fake accounts people are creating fake mail accounts and we don't need to produce any authentication for creating mail Id expect phone number. If you are following any improper things in the social networking we need to identify

that before spreading to others. Some issues like country problems. A political drama can be added as a tag in the twitter with and Hash tags.

DISADVANTAGES

- Because of Privacy Issues the Facebook dataset is very limited and a lot of details are not made public.
- Assuring of the fake accounts is difficult.
- Human are poor in labelling reviews as fake or genuine.

3.3 PROPOSED SYSTEM

The Application Domain of the following project was Community Detection. Community detection is key to understanding the structure of complex networks, and ultimately extracting useful information from them. In this project, we came up with a framework through which we can detect a fake profile using machine learning algorithms so that the social life of people become secured.

1. Classification starts from the selection of profile that needs to be classified.
2. Once the profile is selected, the useful features are extracted for the purpose of classification.
3. The extracted features are then fed to trained classifier.
4. Classifier is trained regularly as new data is fed into the classifier.
5. Classifier then determines whether the profile is genuine or fake.
6. The result of classification algorithm is then verified and feedback is fed back into the classifier.
7. As the number of training data increases the classifier becomes more and more accurate in predicting the fake profiles.

ADVANTAGES

- The social networking sites are making our social lives better but nevertheless there are a lot of issues with using these social networking sites.
- The issues are privacy, online bullying, potential for misuse, trolling, etc. These are done mostly by using fake profiles.

- In this project, we came up with a framework through which we can detect a fake profile using machine learning algorithms so that the social life of people become secured.

3.4 SYSTEM REQUIREMENTS:

Hardware Requirements

- System : Intel Pentium IV 2.80 GHz.
- Monitor : LED.
- Mouse : Logitech.
- Ram : 4.00 GB or above 4.00 GB
- Hard Disk : 250 GB

Software Requirements:

- Operating system : Windows 7, Ubuntu
- Language : Python 3

3.5 SYSTEM REQUIREMENT SPECIFICATION

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS also functions as a blueprint for completing a project with as little cost growth as possible.

The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

It is important to note that an SRS contains functional and non-functional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

Functional Requirement

- Requirement analysis is a software engineering task that bridges the gap between system level software allocation & software design.
- It enables the system engineer to specify software interface with other system elements & establishes design constraints that the software must meet.
- It provides the software design with a representation of information & function that can be translated to data, architecture & procedural design.

Non-functional Requirement

- Security:

Project level security is set. User needs to login when they start the program option is also provided to create the additional user and level security. Presently user level security is not set but can be implemented with few modifications.

- Reliability, Availability, Maintainability:

It is very user friendly, software is secure and there is not much maintenance. Project can be upgraded as per the requirement step by step.

- Configuration and Compatibility:

Describes requirements such as those connected with individual customization or operations in specific competing environments.

- Usability:

Describes items that will ensure the user friendliness of the software.

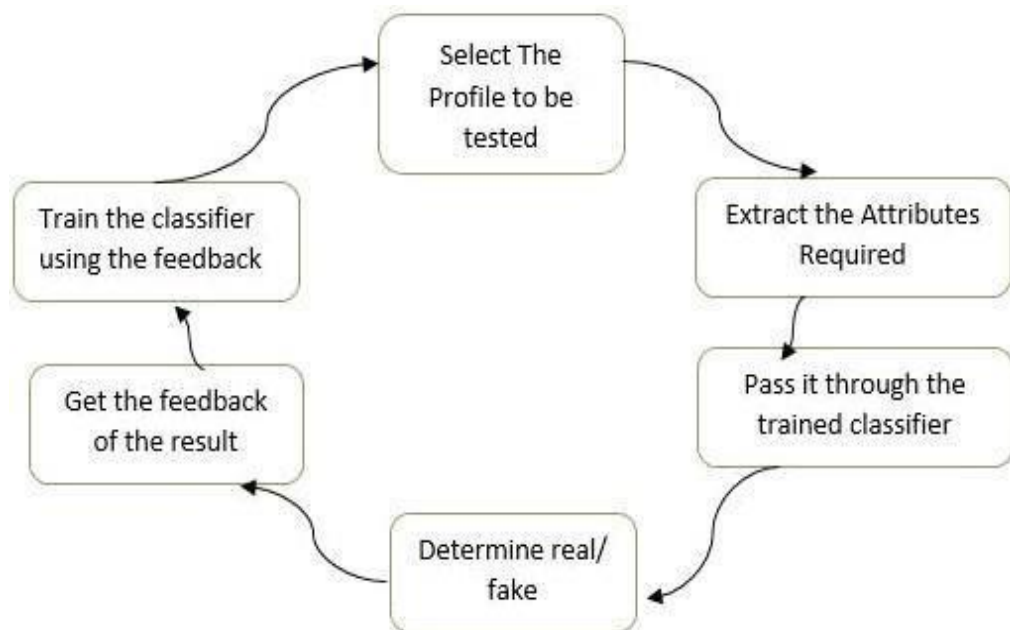
Eg: Includes error messages that direct the user to a solution, input range checkings soon as entries are made and order of choices and screen corresponding to user performances

CHAPTER 4

METHODOLOGY

4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual design that defines the structure and behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.



4.2 MODULE DESCRIPTIONS

- Admin Module
- Data Collection
- Train and Test

- Machine Learning Technique
- Detection of Fake User

Admin Module:

In the first module, we develop the Online Social Networking (OSN) system module. We build up the system with the feature of Online Social Networking System, Twitter. Where, this module is used for admin login with their authentication.

Data Collection:

We will be using a Python Library called *Tweepy* 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.

Some of the most important fields are:

- *text*, which contains the text included in the tweet.
- *created_at*, which is a timestamp of when the tweet was created.
- *user*, which contains information about the user that created the tweet, like the username and user id.

Train and Test:

- We present the proposed framework for metadata features are extracted from available additional information regarding the tweets of a user, whereas content-based features aim to observe the message posting behavior of a user and the quality of the text that the user uses in posts.

4.3 Machine Learning Technique:

- The number of features, which are associated with tweet content, and the characteristics of users are recognized for the detection of spammers. These features are considered as the characteristics of machine learning process for categorizing users, i.e., to know whether they are spammers or not.

- In order to recognize the approach for detecting spammers on Twitter, the labelled collection in pre-classification of fake user and legitimate user has been done. Next, those steps are taken which are needed for the construction of labeled collection and acquired various desired properties.
- In other words, steps which are essential to be examined to develop the collection of users that can be labelled as fake user or legitimate user. At the end, user attributes are identified based on their behavior, e.g., who they interact with and what is the frequency of their interaction.
- In order to confirm this instinct, features of users of the labelled collection has been checked. Two attribute sets are considered, i.e., content attributes and user behavior attributes, to differentiate one user from the other.

Detection of Fake User:

- In this module, we implement 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 fake user is performed to check through all datasets that are available to detect the malignant.
- Feature extraction separates the characteristics construct based on the language model that uses language as a tool and helps in determining whether the user is 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 fake user detection uses the classification technique to accept tweets as the input and classify the fake user and legitimate user.

4.4 SYSTEM DESIGN

Design is a creative process; a good design is the key to effective system. The system

“Design” is defined as “The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. Various design features are followed to develop the system.

The design specification describes the features of the system, the components or elements of the system and their appearance to end-users.

- **System development methodology**

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

- **Model phases**

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

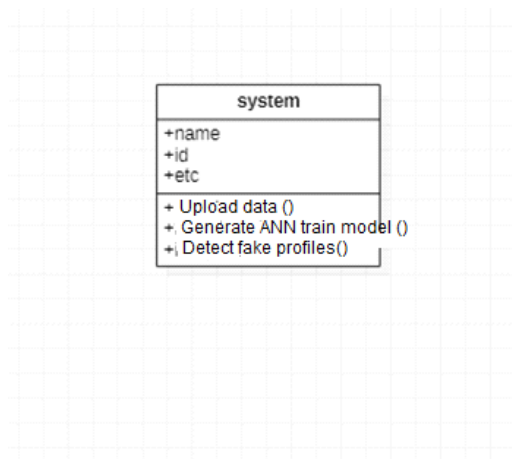
- **Requirement Analysis:** This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.
- **System Design:** Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm, data structure, software architecture etc.
- **Coding:** In this phase programmer starts his coding in order to give a full sketch of product.

In other words system specifications are converted in to machine readable compute code. **Implementation:** The implementation involves coding or programming of the software. The output of this phase is typically user manuals and additional software documentation .

- **Testing:** In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation
- **Maintenance:** The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

4.5 CLASSES DESIGNED FOR THE SYSTEM

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The class diagram is shown below.



4.5.1 USE CASE DIAGRAM OF THE SYSTEM

A use case diagram is a type of behavioral diagram 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.

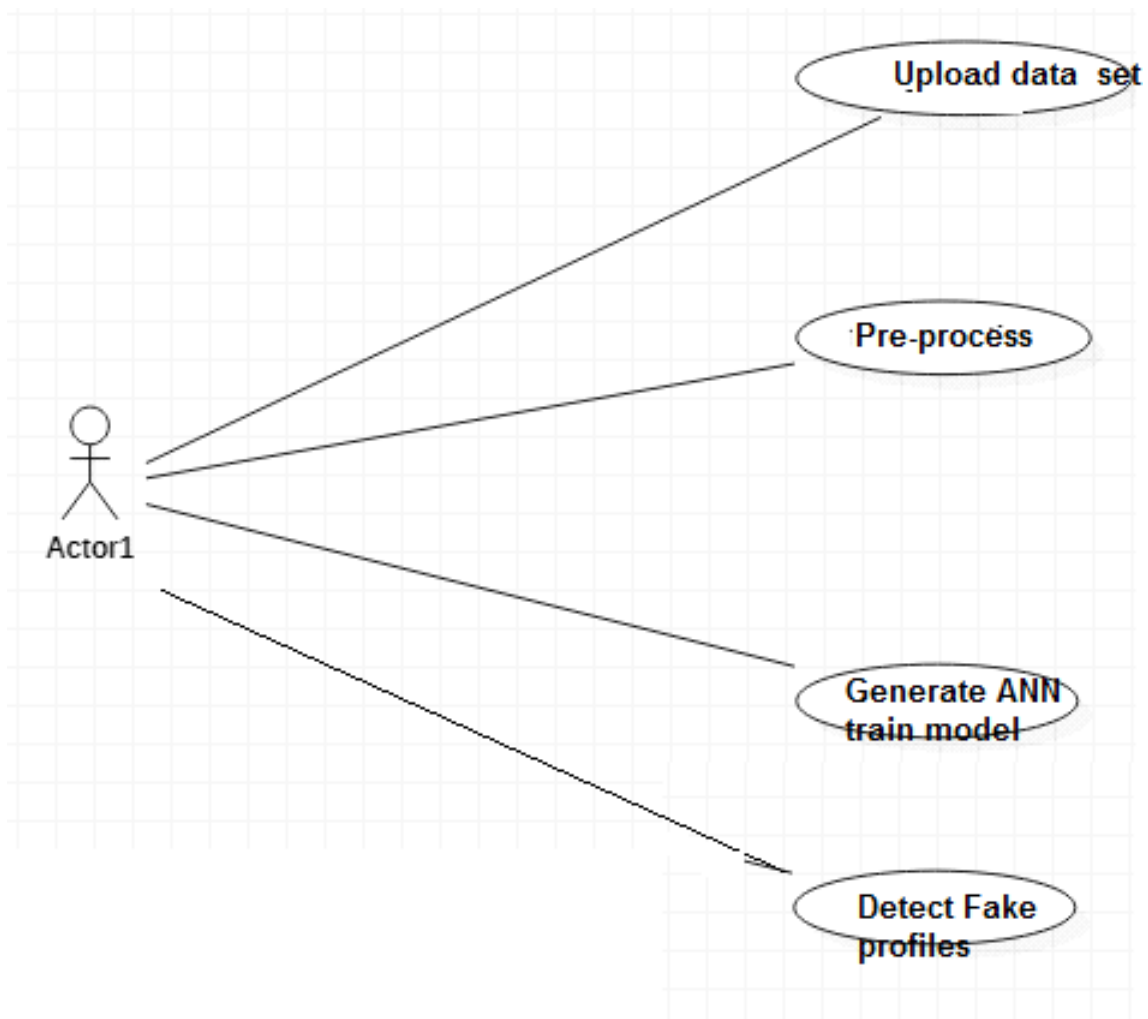


Fig 4.1 Use case Diagram

4.5.2 SEQUENCE DIAGRAM OF SYSTEM OPERATION

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 Chart. The sequence diagrams shows below.

Sequence Diagram for upload review

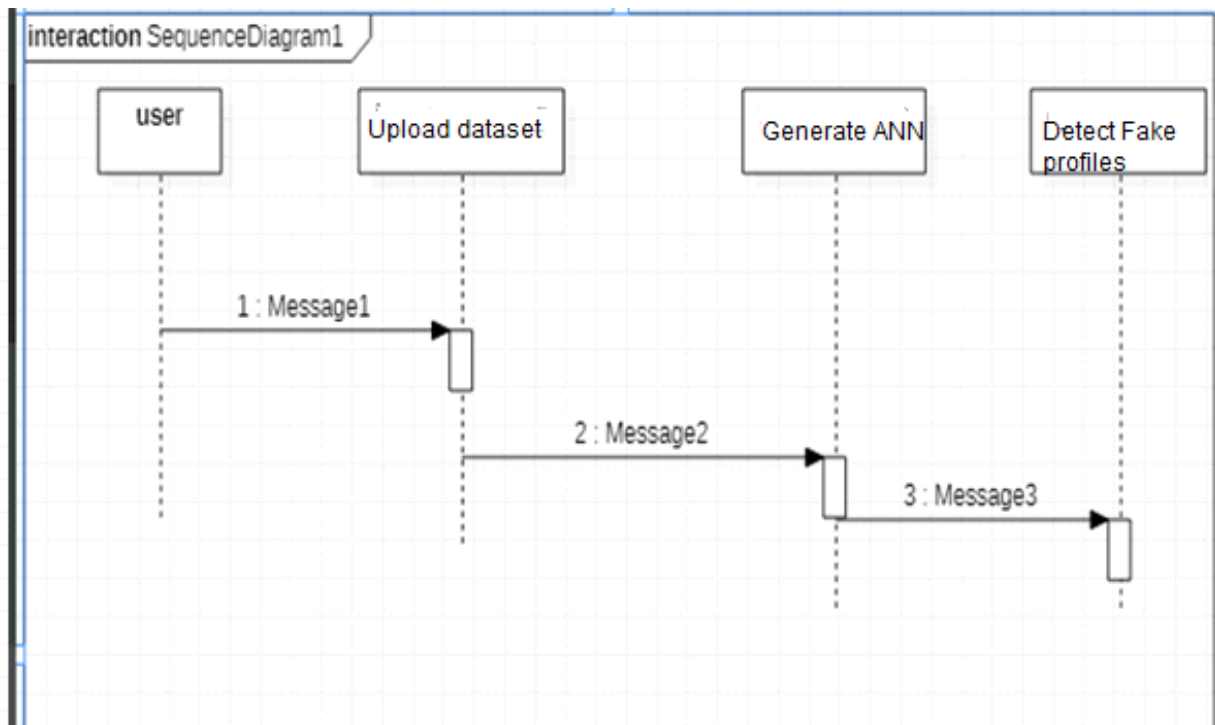


Fig 4.2 Sequence Diagram

4.6 INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?

- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

4.7 SOFTWARE ENVIRONMENT

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.

- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.

To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

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

```
$ python
```

```
Python2.4.3(#1,Nov112010,13:34:43)
```

```
[GCC 4.1.220080704(RedHat4.1.2-48)] 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 –

```
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!

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

S r . N o	Methods & Description
	GET
1	Sends data in unencrypted form to the server. Most common method.

2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<form action="http://localhost:5000/login" method="post">

<p>Enter Name:</p>

<p><input type="text" name="nm"/></p>

<p><input type="submit" value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask import Flask, redirect, url_for, request
```

```
app=Flask(__name__)
```

```
@app.route('/success/<name>')
```

```
def success(name):
```

```
    return 'welcome %s'% name
```

```
@app.route('/login', methods=['POST', 'GET'])
```

```
def login():
```

```
    if request.method == 'POST':
```

```
        user=request.form['nm']
```

```
        return redirect(url_for('success', name= user))
```

```
    else:
```

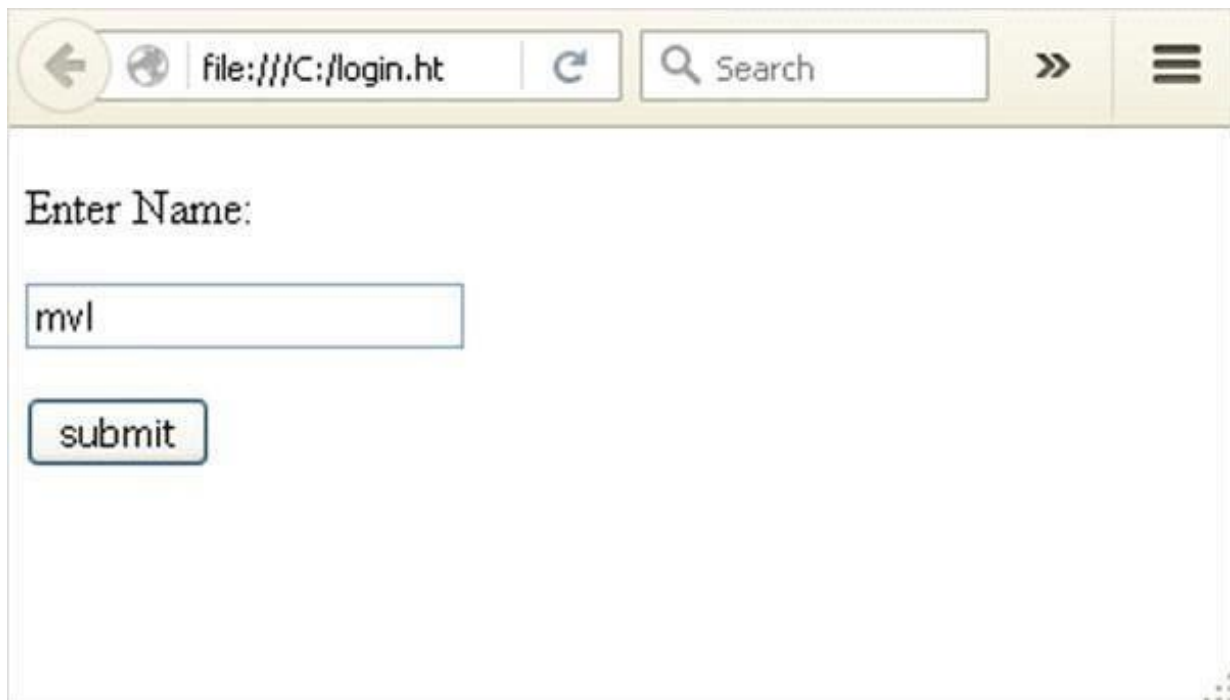
```
        user=request.args.get('nm')
```

```
        return redirect(url_for('success', name= user))
```

```
if __name__ == '__main__':
```

```
    app.run(debug = True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.



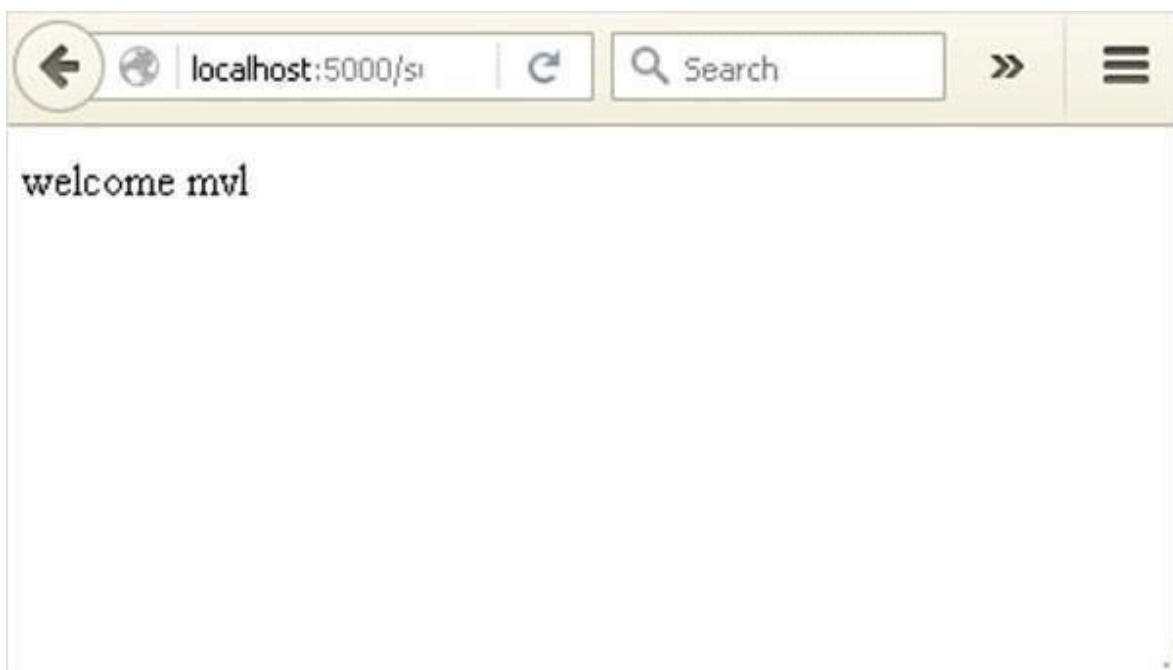
Enter Name:

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by –

```
user = request.form['nm']
```

It is passed to '**success**' URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Change the method parameter to '**GET**' in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by –

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

4.8 SYSTEM STUDY

- **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.

4.9 SYSTEM TESTING

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.

4.9.1 TYPES OF TESTS

- **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

- **System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System

testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

- **White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

- **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

- **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.

4.9.2 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.

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.

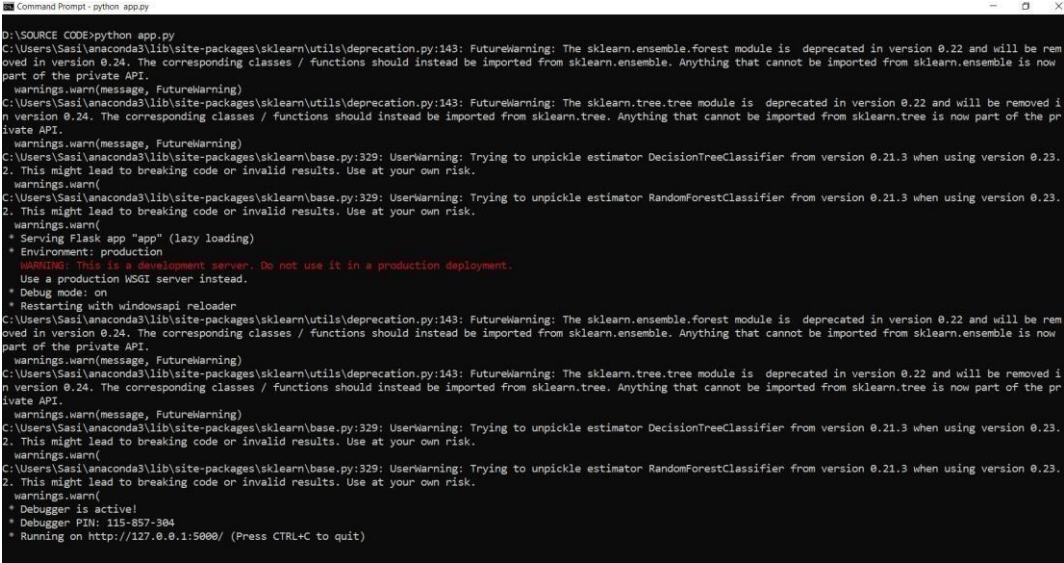
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

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.

CHAPTER-5

RESULTS AND DISCUSSION

We present the evaluation of spammer detection, by analyzing the ground truth described, consisting of many spam accounts including benign accounts (from verified accounts and their followings). We compute feature values for each of these accounts and feed them to four different supervised learning algorithms - Decision Tree (DT), Random Forest (RF)



```
D:\SOURCE CODE>python app.py
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.ensemble.forest module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.ensemble. Anything that cannot be imported from sklearn.ensemble is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.tree.tree module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.tree. Anything that cannot be imported from sklearn.tree is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.21.3 when using version 0.23. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.21.3 when using version 0.23. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdogpy reload
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.ensemble.forest module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.ensemble. Anything that cannot be imported from sklearn.ensemble is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.tree.tree module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.tree. Anything that cannot be imported from sklearn.tree is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.21.3 when using version 0.23. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\Sasi\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.21.3 when using version 0.23. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
* Debugger is active!
* Debugger PIN: 115-857-304
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig 5.1 path of the source code

Also, we chose these supervised learning algorithms to compare our approach with existing solutions. All performance statistics reported here are based on random forest validation over the ground truth (or trainset).

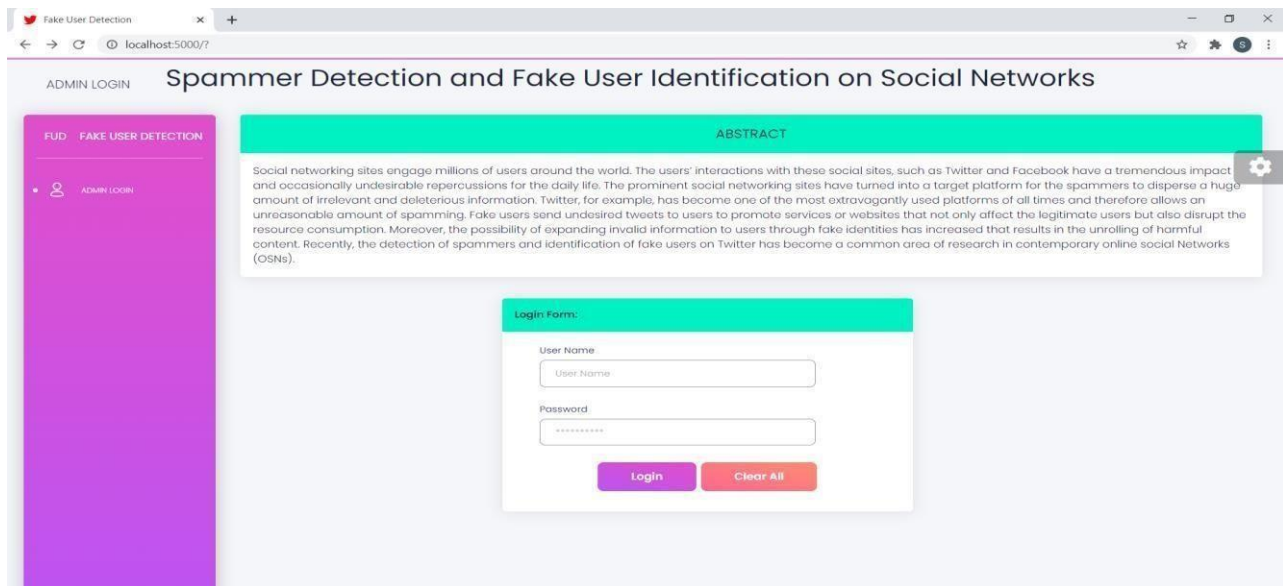


Fig 5.2 Login page

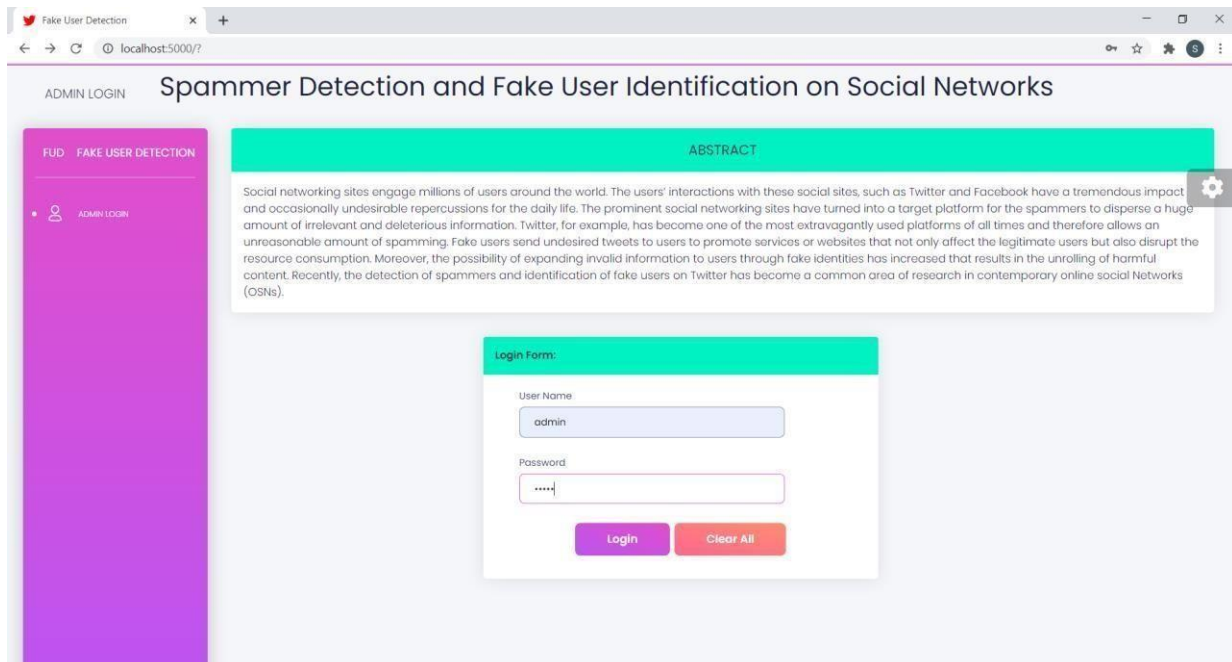


Fig 5.3 Module-1: Admin login

	created_at	username	tweet_id	text	favorite_count	retweet_count	place
0	Mon Feb 22 17:11:29 +0000 2021	Akikabi	1363899224539160585	RT @banshee44mods: Banshee-44 is selling:\n\nQuick Access Sling\n- Sold 7 times in the last year\n- 1.92% year drop rate\n- Last sold Feb 4th 20...	0	5	None
1	Mon Feb 22 17:23:34 +0000 2021	AnR_Gakupo	1363902264469950464	I like eggplant, eggplant someone please give me> < #bot	0	0	None
2	Mon Feb 22 17:20:37 +0000 2021	AttackingFootie	1363901520475996173	@cfcjex... how you calling someone a nerd while you sit in a dark room beating to some computer generated sketch🤔🤔🤔	0	0	None
3	Mon Feb 22 17:28:13 +0000 2021	BizSales247	1363903433393254404	RT @JolliffeCork: If you want to stay ahead with the latest tax, employment and economic news check out our daily news updates 📩 https://t...	0	1	None
4	Sun Feb 21 22:40:21 +0000 2021	Blogging__Guide	1363619597300109314	Signal Review-How to Auto Tweet Your Blog Posts Blogging Guide https://t.co/WIA16jGivh\n#autotweet #tweet... https://t.co/Ww2lsSKkIq	0	0	None
5	Mon Feb 22 15:01:04	Blogging__Guide	1363866402109685762	Signal Review-How to Auto Tweet Your Blog Posts Blogging Guide https://t.co/WIA16jGivh\n#autotweet #tweet... https://t.co/Ww2lsSKkIq	0	0	None

Fig 5.4 Module-2: Data Collection

We clustered and analyzed the users detected by our approach. Clustering users based on what they are spamming and mode of spam was a difficult task because of presence of randomness and varying tactics amongst users.

Various techniques used by spammers to hide themselves in huge online social networks which make it difficult to cluster them. We intend to study the state-of-art and outreach of spammers campaigns on Twitter.

	created_at	username	tweet_id	text	favorite_count	retweet_count	place
	+0000 2021			https://t.co/4xmPCBGyZo #Hourly			{\"states\": {\"contained_within\": [], \"bounding_box\": {\"polygon\": {\"coordinates\": [[[-75.903781, 40.148297], [-75.8778851, 40.148297], [-75.8778851, 40.16162], [-75.903781, 40.161624]]], \"attributes\": {}}}}
8	Thu Feb 25 06:04:48 +0000 2021	BotRakim	13648018601333698562	random number 66 #twitterbot	0	0	None
9	Thu Feb 25 05:04:48 +0000 2021	BotRakim	1364803501289844737	random number 67 #twitterbot	0	0	None
10	Thu Feb 25 04:38:04 +0000 2021	CalefybTech	1364796781901451265	RT @hasdidi: #ReactJS #Automated React Static Tweets https://t.co/aZq9erYv8w Spotify Now Available In Jamaica - https://t.co/HnVKJpdNj	0	2	None
11	Thu Feb 25 06:10:58 +0000 2021	DubCorner	1364820164059955203	#dubcorner @dalevirgo #trending #updates #jamaica... https://t.co/nGVO4X9Z9x	0	0	None
12	Thu Feb 25 06:11:29 +0000 2021	DubCorner	1364820293378842626	Spotify Now Available In Jamaica - https://t.co/ubpdmP850j #dubcorner @dalevirgo #trending #updates #jamaica... https://t.co/On1vd7LGWh	0	0	None

Fig 5.5 Data Collection

We find that a cluster of users post URLs which when visited respond differently to different HTTP user-agents. For instance, a visit to a spam URL using conventional probe tools (such as we get) does not reveal the page, as against the case where some of the same URL is visited through a web browser. More specifically, following anomalous patterns have been observed with such URLs:

- Bad http response
- Returning different URL based on HTTP request agent.
- URL no longer removed.
- Domain name no longer available.
- Blogger deleted either the post or the posting account.

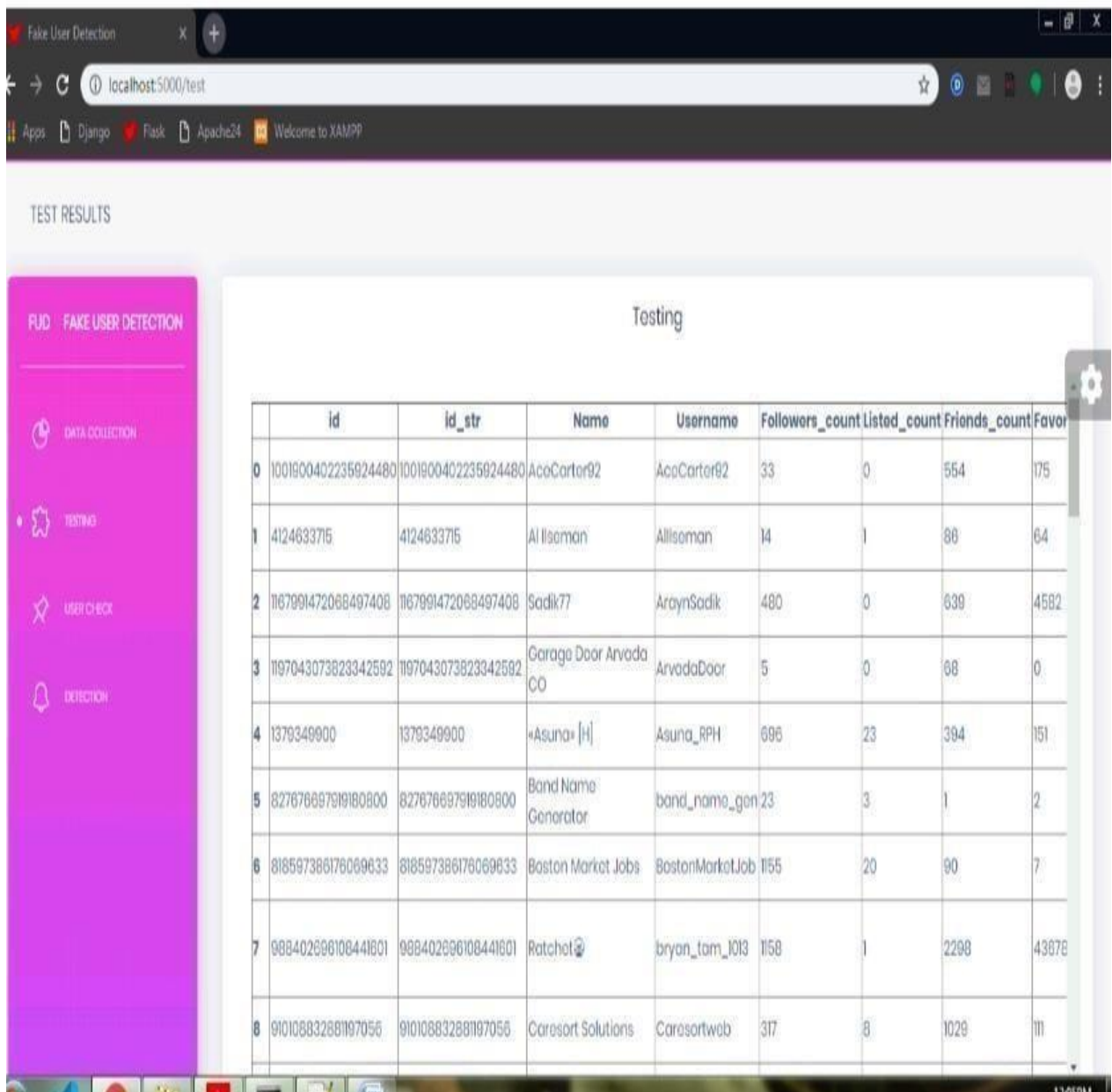
Twitter suspends only a small portion of users whose URL content is missing. While we highlight a good performance using our features when compared to existing techniques for spam detection, we also note that spammers may modify their own methodologies to evade detection. For instance, individual features proposed in this work, may be altered in a fashion which resembles the activity of a legitimate account



```
Command Prompt - python app.py
wrote tweets by DubCorner to CSV.
Gnerdicus
wrote tweets by Gnerdicus to CSV.
Inri_shou
wrote tweets by Inri_shou to CSV.
JaakkoTheRipper
wrote tweets by JaakkoTheRipper to CSV.
JusticeWAR007
wrote tweets by JusticeWAR007 to CSV.
KhAAdvocates
wrote tweets by KhAAdvocates to CSV.
MEP_LLC
wrote tweets by MEP_LLC to CSV.
MusicianFighter
wrote tweets by MusicianFighter to CSV.
OficialW24PY
wrote tweets by OficialW24PY to CSV.
PaperboyOnline
wrote tweets by PaperboyOnline to CSV.
Richard03954394
wrote tweets by Richard03954394 to CSV.
SanJogAul
wrote tweets by SanJogAul to CSV.
TwitMarke7s
wrote tweets by TwitMarke7s to CSV.
Tyrithall
wrote tweets by Tyrithall to CSV.
YoursShakti
wrote tweets by YoursShakti to CSV.
akademixtoo
wrote tweets by akademixtoo to CSV.
band_name_gen
wrote tweets by band_name_gen to CSV.
band_name_gen
wrote tweets by band_name_gen to CSV.
bloggingcash
wrote tweets by bloggingcash to CSV.
cornellooo
wrote tweets by cornellooo to CSV.
cornellooo
```

Fig 5.6 Testing data in background

We note that previous approaches have proposed methods with a smaller set of features for detecting spammers. Such an approach, and several from the past have explored the social graph and quantified metrics based who and how is a spammer connected to other users. We note that collection of an extensive (and sufficiently informational) social graph of an account is non-trivial. On the contrary, our technique utilizes a modest number of tweets for a given account for analysis.



The screenshot shows a web browser window with the address bar at 'localhost:5000/test'. The page title is 'Fake User Detection'. On the left is a pink sidebar with navigation links: 'FAKE USER DETECTION', 'DATA COLLECTION', 'TESTING' (highlighted), 'USER CHECK', and 'DETECTION'. The main content area is titled 'Testing' and displays a table of test results.

	id	id_str	Name	Username	Followers_count	Listed_count	Friends_count	Favor
0	1001900402235924480	1001900402235924480	AcoCarter92	AcoCarter92	33	0	554	175
1	4124633715	4124633715	Al Ilisoman	Allisoman	14	1	88	64
2	1167991472068497408	1167991472068497408	Sadik77	AraynSadik	480	0	639	4582
3	1197043073823342592	1197043073823342592	Garage Door Arvada CO	ArvadaDoor	5	0	68	0
4	1379349900	1379349900	«Asuna» [H]	Asuna_RPH	696	23	394	151
5	827676697919180800	827676697919180800	Band Name Generator	band_name_gen23	3	1	2	
6	818597386176069633	818597386176069633	Boston Market Jobs	BostonMarketJob	155	20	90	7
7	988402696108441601	988402696108441601	Ratchet	bryan_tam_1013	1158	1	2298	43878
8	91010683288197056	91010683288197056	Caresort Solutions	Caresortweb	317	8	1029	111

Fig 5.7 Module -3: Testing Data

Location	Count	Profile	URL	Is Fake	Percentage	Classification
Mumbai, India	610	Analyst, TA, Mentor, Consultant Panelist - ET...	https://t.co/MobXoRD9d	False	10.0%	Legitimate User
Atlanta, Georgia	3897	Social Media Maker for #music #art #digital #p...	https://t.co/F4Uwc37oO	True	15.0%	Legitimate User
School Library	10011	[Verified by @CO_RP] Famous Genius Girl Kotem...	NaN	False	64.0%	Fake/Bot User
NaN	228	NaN	NaN	False	60.0%	Fake/Bot User
NaN	187	NaN	NaN	False	50.0%	Legitimate User
Hong Kong	2368	An adventurer, A writer, A dreamer, \n\nFictio...	NaN	False	17.0%	Legitimate User
Global	151	Disruptive #Fintech Assata w/#Blockchain LONG ...	https://t.co/w3rduToQp	False	24.0%	Legitimate User
NaN	26456	I'm the jellicle leader of the jellicles and m...	NaN	False	90.0%	Fake/Bot User
		Trihaa Ratinam Humma Na				Fake/Bot

Fig 5.7.1 Module -3: Testing Data

User Check

Fake User Detection

@AccCarter92 Click search button

Next | Detect

Fig 5.8 User Check

The system asks to enter the user name for the verification when entered into user check page

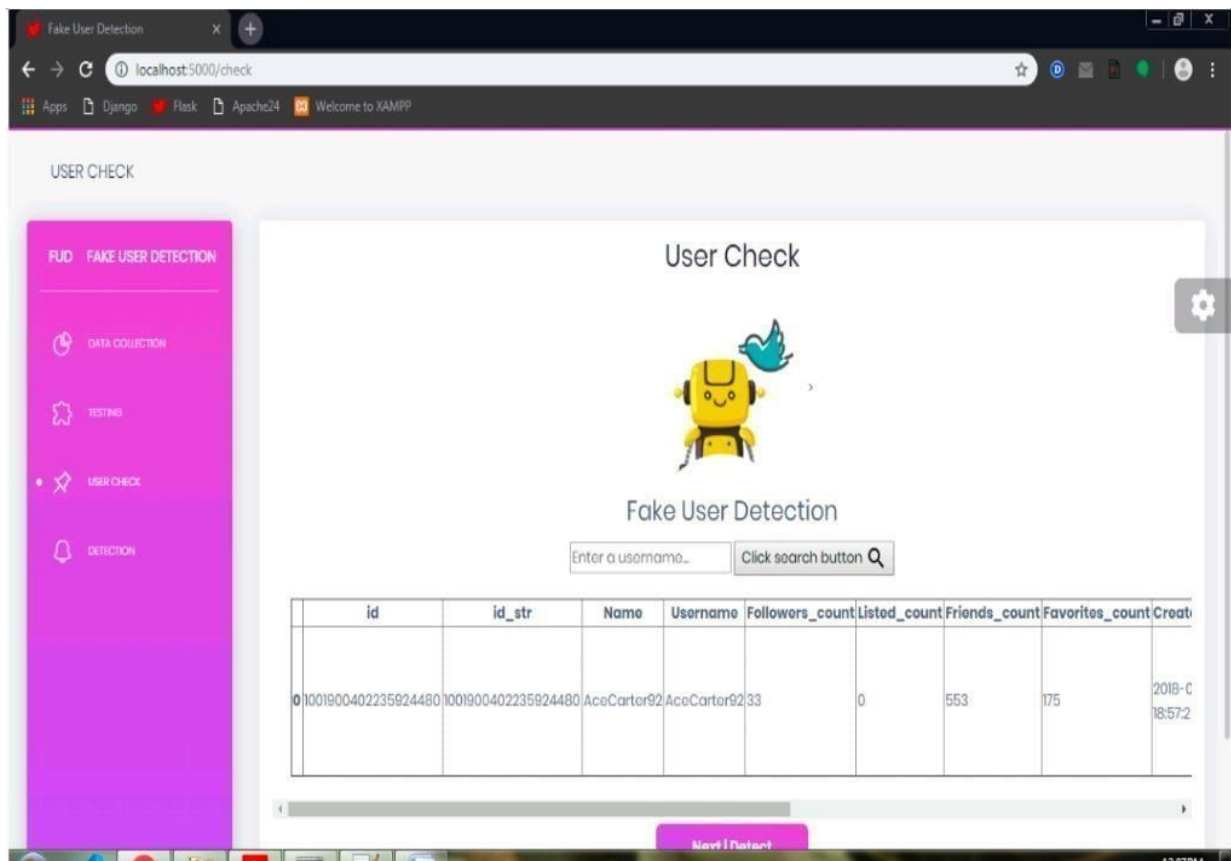


Fig 5.9 Enter user details

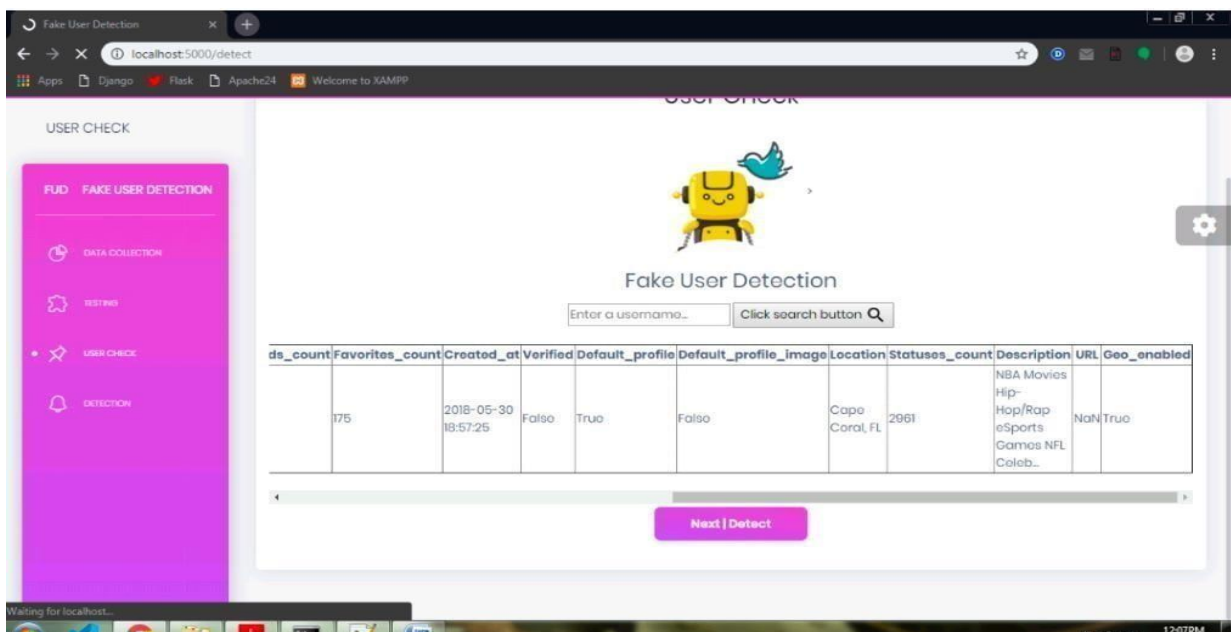


Fig 5.10 Details of the user are entered for verification

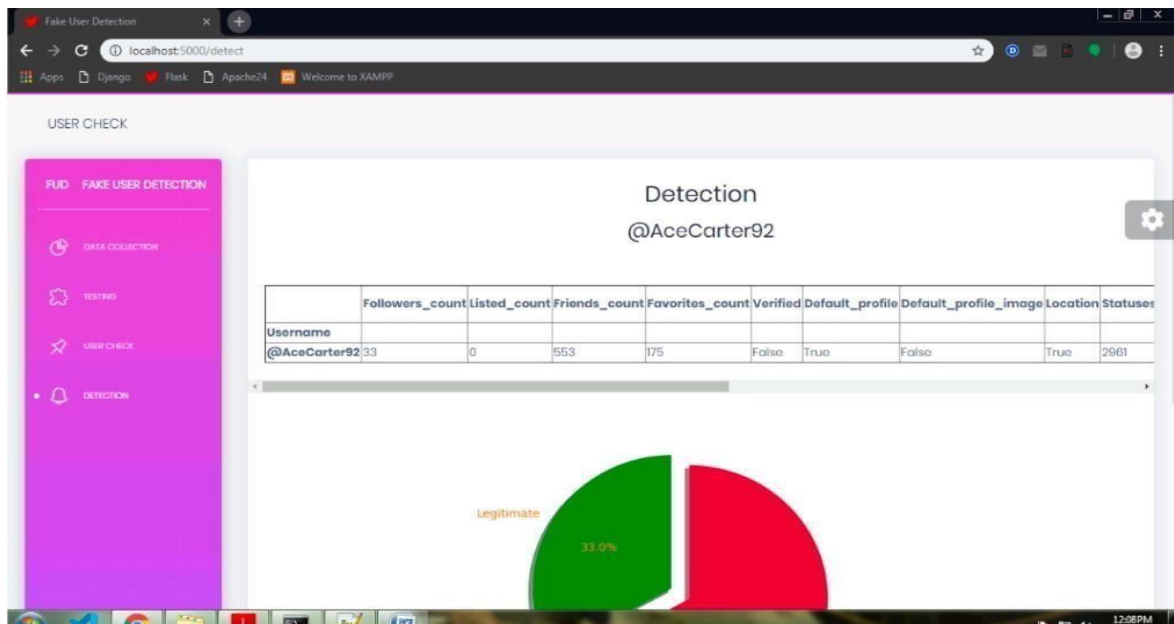


Fig 5.10 Details of the user are entered for verification

Twitter is already heavily burdened with its daily increasing usage. Analysis of each single user needs tweets and user information which overburdens Twitter database. In order to avoid this, we built a system which can classify all incoming tweets into two classes - spam, legitimate

This can be used not only in alerting users about possible spam in tweets as it is posted, it can also be helpful in filtering users

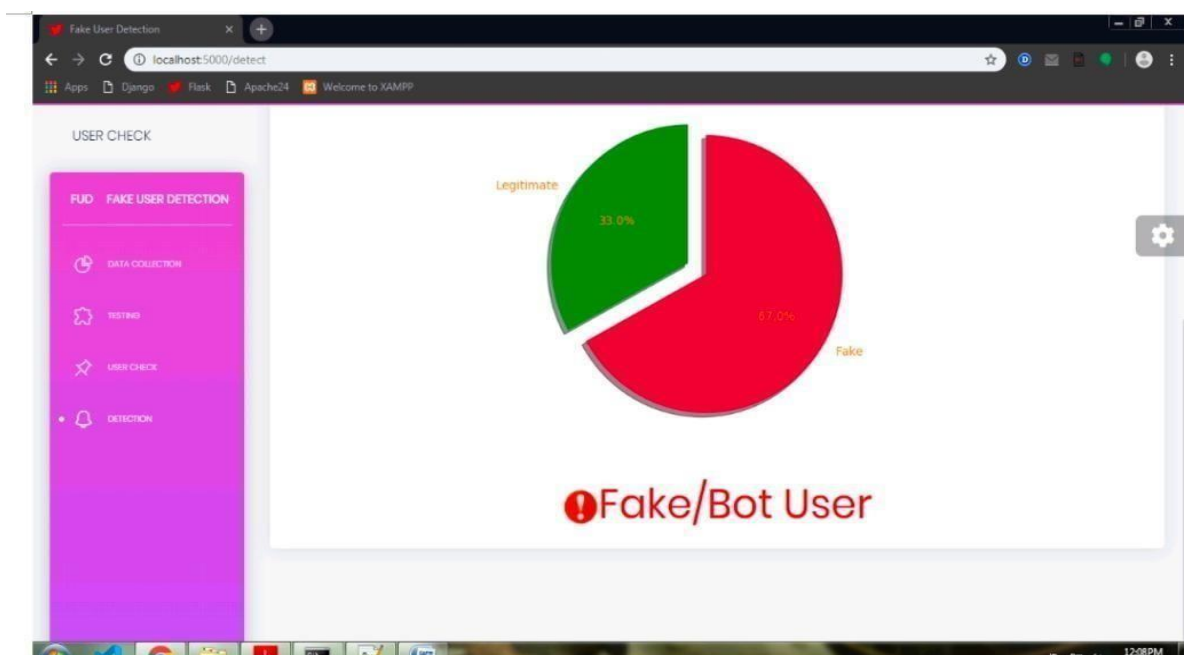


Fig 5.11 Detecting spam

The entered user is 67% fake and 33% genuine as per the system

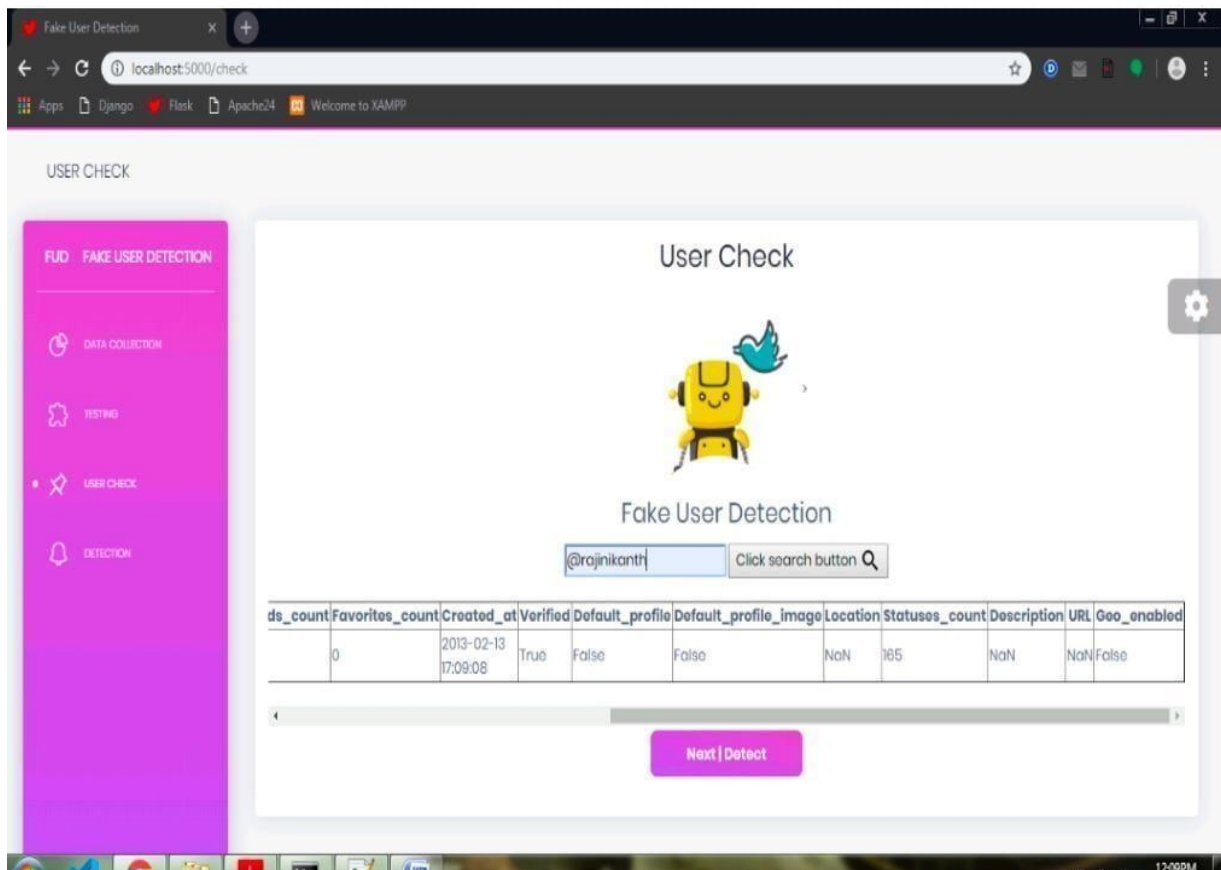


Fig 5.12 Enter the user details

Details of the user are entered for verification

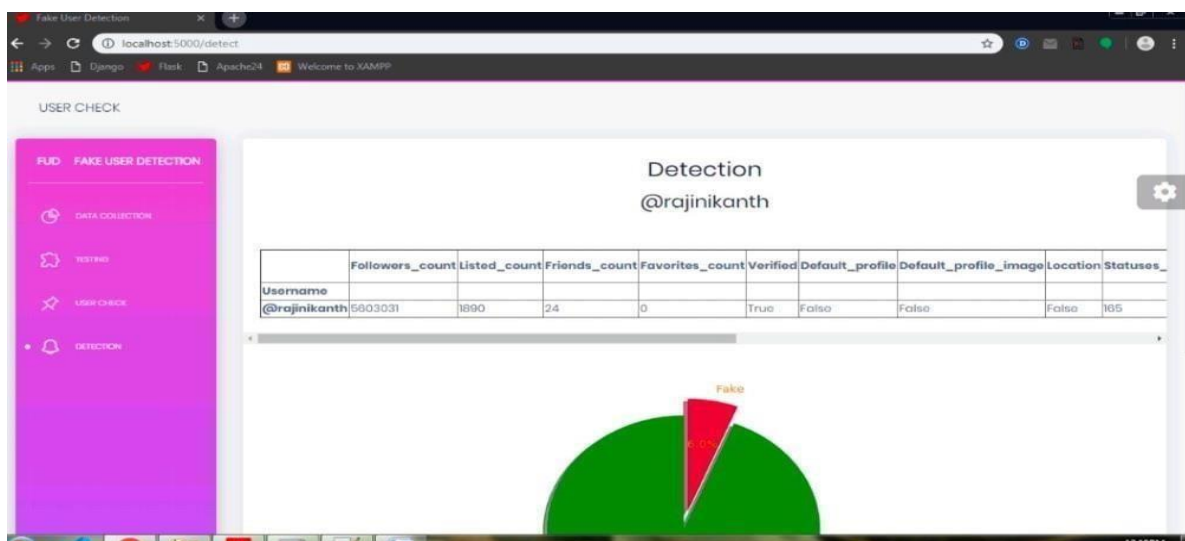


Fig 5.13 System verification of the user

The entered user is 15% fake and 85% genuine as per the system

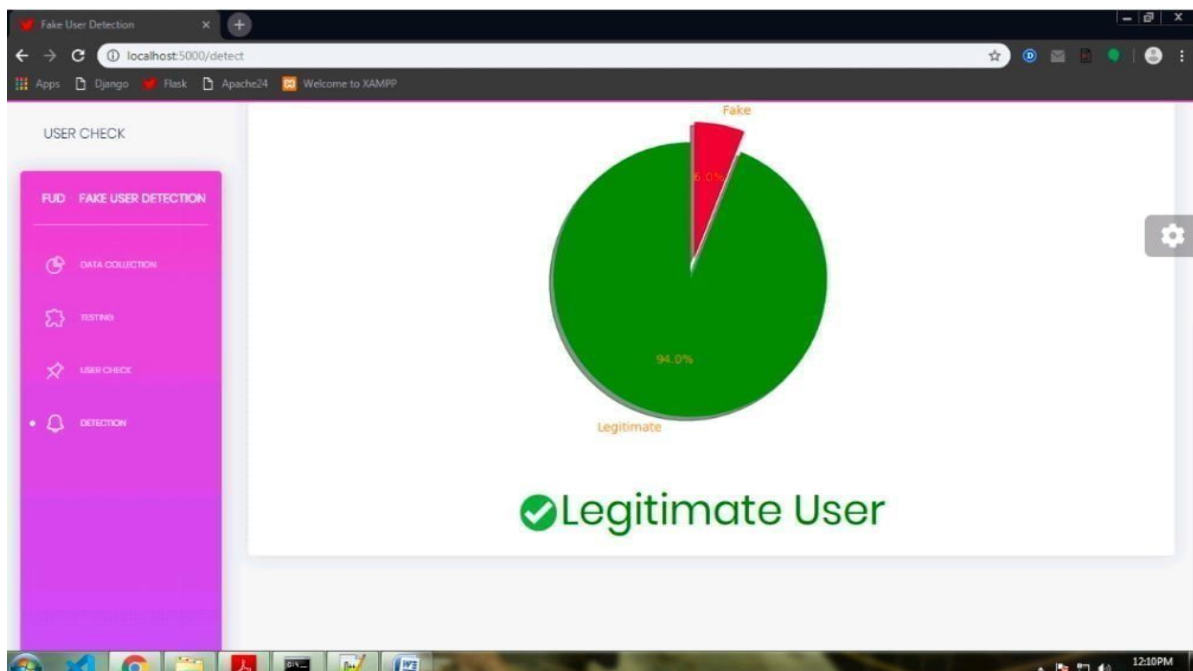


Fig 5.14 Genuine user

The system predicts whether the given user is spam or a genuine user.

As a future work, we plan to expand upon the category of baits used to compromise victims. Therefore, we plan to further categorize the trends that the spammers use to be able to distinguish them from legitimate users. Additionally, we plan to use evolving campaigns as indicators for the proliferation of spam, and use it as an antidote for anomalies.

CHAPTER-6

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 interms 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 issue are briefly highlighted as under: False news identification on social media networks is an issue that needs to be exploredbecause 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 rumorsources on social media. Although a few studies based on statistical method shave 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.

REFERENCES

- [1] B. Erçahin, Ö. Akta³, D. Kiliç, and C. Akyol, ``Twitter fake account detection,'' in Proc. Int. Conf. Comput. Sci. Eng. (UBMK), Oct. 2017, pp. 388_392.
- [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. 435_438.
- [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 behaviors exhibited by spammers in popular social media networks,'' in Proc. Int. Conf. Circuit, Power Comput. Tech- nol. (ICCPCT), Mar. 2016, pp. 1_6.
- [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. 1_12.

[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. 1_6.

[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. 347_351.

[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), Nov. 2017, pp. 208_215.

[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. 65_76, 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. 373_378.
- [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. 466_471.
- [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. 1_6.
- [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. 1_9.
- [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. 1_5.
- [17] S. Jeong, G. Noh, H. Oh, and C.-K. Kim, ``Follow spam detection based on cascaded social information,'' Inf. Sci., vol. 369, pp. 481_499, 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. 109_116.

[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 filtering: A big data problem,'' J. Supercomput., vol. 74, no. 10, pp. 5003_5021, Oct. 2018.

[21] C. Meda, E. Ragusa, C. Gianoglio, R. Zunino, A. Ottaviano, E. Scillia, and R. Surlinelli, ``Spam detection of Twitter traf_c: 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. 811_817.