

SPAM STOP:SPAM IDENTIFICATION AND DELETION

Submitted in Partial Fulfillment of The Requirements

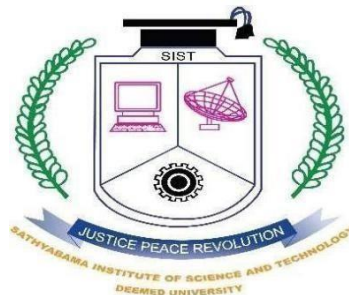
For The Award Of

Bachelor of Engineering Degree in Computer Science and Engineering

By

ATYAM VINAY (37110060)

AVULA MANOJ REDDY(37110063)



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 –

600119

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NACC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119



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 **ATYAM VINAY** (37110060), AVULA MONOJ REDDY (37110063) who carried out the project entitled "**SPAM STOP:SPAM IDENTIFICATION AND DELETION**" under my supervision from November 2020 to April 2021.

Internal Guide

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

Head of the Department

Submitted for viva voice Examination held on_____.

Internal Examiner

External Examiner

DECLARATION

I, **AVULA MANOJ REDDY(37110063)** hereby declare that the project report entitled **“SPAM STOP:SPAM IDENTIFICATION AND DELETION”** done by me under the guidance of **Dr. S. VIGNESHWARI, M.E., Ph.D.**, 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. Vigneshwari 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 his 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

In the **Existing** system, Spam has become the platform of choice used by cyber-criminals to spread malicious payloads such as viruses and Trojans. In the **Proposed** system, the problem of early detection of spam campaigns. Collaborative spam detection techniques can deal with large scale e-mail data contributed by multiple sources; however, they have the well-known problem of requiring disclosure of e-mail content. Distance-preserving hashes are one of the common solutions used for preserving the privacy of e-mail content while enabling message classification for spam detection. In the **Modification** process, use can create an android based application which principally categories all the messages based on the keyword filtration like banking category based on the keywords like OTP, Bank ID and so on. So apart from spam filtering we can also implement overall filtering based concept also in this project.

TABLE OF CONTENTS

ABSTRACT

v

LIST OF FIGURES

viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 DOMAIN INTRODUCTION	1
	1.2 PROJECT INTRODUCTION	5
	1.3 EXISTING SYSTEM	5
	1.4 PROPOSED SYSTEM	5
2.	LITERATURE SURVEY	6
3.	METHODOLOGY	9
	3.1 AIM OF THE PROJECT	9
	3.2 SYSTEM SPECIFICATION	9
	3.2.1 Hardware specification	9
	3.2.2 Software specification	9
	3.3 OVERVIEW OF THE PLATFORM	9
	3.3.1 Java	9
	3.3.2 MySQL Overview	11
	3.3.4 Support Vector Machine Algorithm	13
	3.4APPLICATIONS OF SVM	16
	3.5SYSTEM ARCHITECTURE	18

		19
4	MODULE DESCRIPTION	
	4.1 SYSTEM STUDY	19
	4.2 MODULE LIST	19
	4.3 DATA FLOW DIAGRAM	21
	4.4 UML DIAGRAM	23
	4.4.1 USE CASE DIAGRAMS	23
	4.4.2 CLASS DIAGRAM	24
	4.4.3 SEQUENCE DIAGRAM	25
	4.4.4 ACTIVITY DIAGRAM	26
5	RESULTS AND DISCUSSION	28
	CONCLUSION AND FUTUREWORK	29
6		
	REFERENCES	30
	APPENDIX	31
	A.SOURCE CODE	31
	B.SCREEN SHOTS	36
	C.PAPER PUBLICATION	41
	D.PLAGIARISM REPORT	49

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
------------	-------	----------

1.1	Four V's Of Big Data	2
3.1	STEPS IN SVM	14
3.2	CLASSSSIFICATION IN SVM	15
3.3	Architecture	18

CHAPTER I

INTRODUCTION

1.1 DOMAIN INTRODUCTION

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on. So we can implement big data in our project because every employ has instructed information so we can make analysis on this data.

Characteristics Of Big Data

Big data is a term utilized to refer to the increase in the volume of data that are difficult to store, process, and analyze through traditional database technologies. The nature of big data is indistinct and involves considerable processes to identify and translate the data into new insights. The term "big data" is relatively new in IT and business. However, several researchers and practitioners have utilized the term in previous literature. For instance, referred to big data as a large volume of scientific data for visualization. Several definitions of big data currently exist." Meanwhile and defined big data as characterized by three Vs: volume, variety, and velocity. The terms volume, variety, and velocity were originally introduced by Garter to describe the elements of big data challenges. IDC also defined big data technologies as "a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling the high velocity capture, discovery, and/or analysis." specified that big data is not only characterized by the three Vs mentioned above but may also extend to four Vs, namely, volume, variety, velocity, and value This

4V definition is widely recognized because it highlights the meaning and necessity of big data.

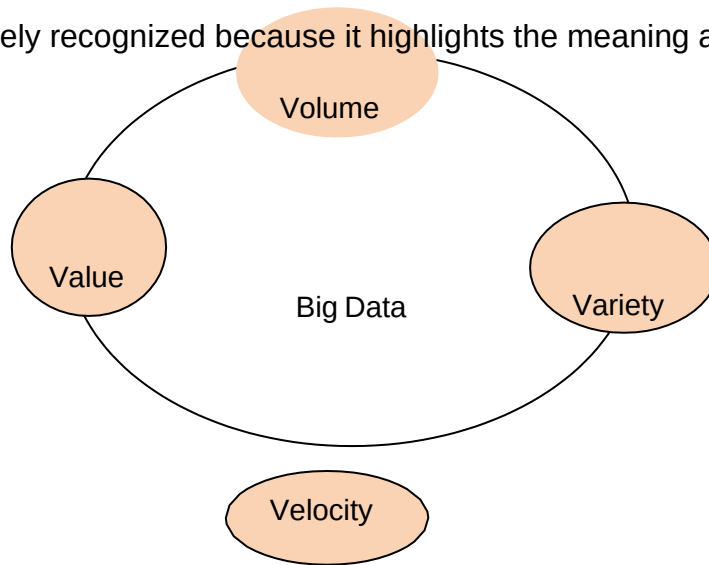


Fig 1.1 : Four V's Of Big Data

Volume - Refers to the amount of all types of data generated from different sources and continue to expand. The benefit of gathering large amounts of data includes the creation of hidden information and patterns through data analysis. Collecting longitudinal data requires considerable effort and underlying investments. Nevertheless, such mobile data challenge produced an interesting result similar to that in the examination of the predictability of human behavior patterns or means to share data based on human mobility and visualization techniques for complex data.

Variety-Refers to the different types of data collected via sensors, smart phones, or social networks. Such data types include video, image, text, audio, and data logs, in either structured or unstructured format. Most of the data generated from mobile applications are in unstructured format. For example, text messages, online games, blogs, and social media generate different types of unstructured data through mobile devices and sensors. Internet users also generate an extremely diverse set of structured and unstructured data.

Velocity -Refers to the speed of data transfer. The contents of data constantly change because of the absorption of complementary data collections, introduction of previously archived data or legacy collections, and streamed data arriving from multiple sources

Value- is the most important aspect of big data; it refers to the process of discovering huge hidden values from large data-sets with various types and rapid generation

Classification Of Big Data

Big data are classified into different categories to better understand their characteristics shows the numerous categories of big data. The classification is important because of large-scale data in the cloud. The classification is based on five aspects: (i) data sources, (ii) content format, (iii) data stores, (iv) data staging, and (v) data processing.

Data sources include internet data, sensing and all stores of transnational information, ranges from unstructured to highly structured are stored in various formats. Most popular is the relational database that come in a large number of varieties. As the result of the wide variety of data sources, the captured data differ in size with respect to redundancy, consistency and noise, etc.

Big Data Storage System

The rapid growth of data has restricted the capability of existing storage technologies to store and manage data. Over the past few years, traditional storage systems have been utilized to store data through structured RDBMS. However, almost storage systems have limitations and are inapplicable to the storage and management of big data. A storage architecture that can be accessed in a highly efficient manner while achieving availability and reliability is required to store and manage large data-sets

Several storage technologies have been developed to meet the demands of massive data. Existing technologies can be classified as direct attached storage (DAS), network attached storage (NAS), and storage area network (SAN). In DAS, various hard disk drives (HDDs) are directly connected to the servers. Each HDD receives a certain amount of input/output (I/O) resource, which is managed by individual applications. Therefore, DAS is suitable only for servers that are interconnected on a small scale. Given the aforesaid low scalability, storage capacity is increased but expand-ability and upgrade- ability are limited significantly. NAS is a storage device that supports a network. NAS is connected directly to a network through a switch or hub via TCP/IP protocols. In NAS, data are transferred as files. Given that the NAS server can indirectly access a storage device through networks, the I/O burden on a NAS server is significantly lighter than that on a DAS server. NAS can orient networks, particularly scalable and bandwidth- intensive networks. Such networks include high-speed networks of optical-fiber

connections. The SAN system of data storage is independent with respect to storage on the local area network (LAN). Multipath data switching is conducted among internal nodes to maximize data management and sharing. The organizational systems of data storage's (DAS, NAS, and SAN) can be divided into three parts: (i) disc array, where the foundation of a storage system provides the fundamental guarantee, (ii) connection and network subsystems, which connect one or more disc arrays and servers, and (iii) storage management software, which oversees data sharing, storage management, and disaster recovery tasks for multiple servers.

Hadoop Background

Hadoop is an open-source Apache Software Foundation project written in Java that enables the distributed processing of large data-sets across clusters of commodity. Hadoop has two primary components, namely, HDFS and Map Reduce programming framework. The most significant feature of Hadoop is that HDFS and Map Reduce are closely related to each other; each are co-deployed such that a single cluster is produced. Therefore, the storage system is not physically separated from the processing system.

HDFS is a distributed file system designed to run on top of the local file systems of the cluster nodes and store extremely large files suitable for streaming data access. HDFS is highly fault tolerant and can scale up from a single server to thousands of machines, each offering local computation and storage. HDFS consists of two types of nodes, namely, a name node called "master" and several data nodes called "slaves." HDFS can also include secondary name nodes. The name node manages the hierarchy of file systems and director namespace (i.e., metadata). File systems are presented in a form of name node that registers attributes, such as access time, modification, permission, and disk space quotas. The file content is split into large blocks, and each block of the file is independently replicated across data nodes for redundancy and to periodically send a report of all existing blocks to the name node.

Map Reduce is a simplified programming model for processing large numbers of datasets pioneered by Google for data-intensive applications. The Map Reduce model was developed based on GFS is adopted through open-source Hadoop implementation, which was popularized by Yahoo. Apart from the Map Reduce framework, several other current open-source Apache projects are related to the Hadoop ecosystem, including

Hive, Hbase, Mahout, Pig, Zookeeper, Spark, and Avro. Twister provides support for efficient and iterative MapReduce computations. An overview of current Map Reduce projects and related software is shown in MapReduce allows an inexperienced programmer to develop parallel programs and create a program capable of using computers in a cloud.

1.2 PROJECT INTRODUCTION

The word spam was originally used to describe unsolicited e-mails sent in bulk. It is hard to define the term spam more accurately. Some argue spam is about the lack of consent on the part of the recipient, while others believe it is about unsolicited e-mail quantity or scale. Other definitions also stressed the commercial nature of spam; for example, the US SPAM act of 2003 established stringent requirements for sending commercial e-mails [1]. Later, spam got closely associated with cyber-crime. Spam emails often try to lure the recipient to click on a fake or infected URL that links to a malicious Website (phishing) or downloads a malicious attachment containing a zero-day exploit (spear-phishing). Spam in all of its forms is still considered as one of the hardest challenges of the connected generation. This in return drove a massive amount of research towards countering it. The effort led to the gradual decline of spamming activities which lead many to believe that spam was no longer a threat. However, recent thorough studies and statistics have concluded otherwise [2]. In fact, about 66.34% of all e-mails sent worldwide are considered spam e-mails according to Kaspersky's Spam and Phishing Statistics for the first quarter of 2014 [3] which leads us to conclude that it is still an evolving phenomenon and is still an active cyber threat.

1.3 EXISTING SYSTEM:

Spam has become the platform of choice used by cyber-criminals to spread malicious payloads such as viruses and trojans .

1.4 PROPOSED SYSTEM:

In the proposed system, user can create an android based application which principally categories all the messages based on the keyword filtration like banking category based on the keywords like OTP, Bank ID and so on. So apart from spam filtering we can also implement overall filtering based concept also in this project.

ADVANTAGES:

- Spam word detection
- Fake product identification
- Fake agencies identification

CHAPTER II

LITERATURE SURVEY

[1] Cybercriminals continue to target online users of banks. They are improving their techniques and using high levels of skill in their attacks. Their continued search for different methods to commit crime makes the existing protection system less effective. They have developed crime toolkits which have become more accessible and simpler to use, and this has attracted more cybercriminals to cybercrime. In this chapter, the authors study the methods that are used in crime toolkits. They present the development and current trend of crime toolkits and reveal the methods that have been used to commit cybercrime successfully.

[2] Survey of review spam detection using machine learning techniques, Online reviews are often the primary factor in a customer's decision to purchase a product or service, and are a valuable source of information that can be used to determine public opinion on these products or services. Because of their impact, manufacturers and retailers are highly concerned with customer feedback and reviews. Reliance on online reviews gives rise to the potential concern that wrongdoers may create false reviews to artificially promote or devalue products and services. This practice is known as Opinion (Review) Spam, where spammers manipulate and poison reviews (i.e., making fake, untruthful, or deceptive reviews) for profit or gain. Since not all online reviews are truthful and trustworthy, it is important to develop techniques for detecting review spam. By extracting meaningful features from the text using Natural Language Processing (NLP), it is possible to conduct review spam detection using various machine learning techniques. Additionally, reviewer information, apart from the text itself, can be used to aid in this process. In this paper, we survey the prominent machine learning techniques that have been proposed to solve the

problem of review spam detection and the performance of different approaches for classification and detection of review spam. The majority of current research has focused on supervised learning methods, which require labeled data, a scarcity when it comes to online review spam. Research on methods for Big Data are of interest, The primary goal of this paper is to provide a strong and comprehensive comparative study of current research on detecting review spam using various machine learning techniques and to devise methodology for conducting further investigation.

[3] Nasreen M Shajideen, Bindu V, Emails are inevitable in this modern era. It has become an effective tool in the communication field. Because of its easiness the number of users are increasing day-by-day. With the increased number of email users, the number of spam mails have also increased. Spam can cause great loss to users. Many spam filtering techniques have been introduced to distinguish

between ham and spam. A mail that appear as spam may appear as ham to another user and vice versa. That is, it depends on the personal preference of user. Therefore an ontology based personalized mail access is necessary. This paper compares the ontology based spam filtering and conventional spam filtering.

[4] Ecosystem of Spamming on Twitter: Analysis of Spam Reporters and Spam Reportees

Lately, there has been a growing trend in the Internet space particularly among the Online Social Media (OSMs) platforms like Twitter, Facebook etc which are becoming huge repositories of information. This information, by design, is posted by users of these websites and consequently, this information is vast, un-organized, unreliable and dynamic. It is commonly observed that along with genuine users, a lot of activity is seen from spammers or users with the intent of spreading malicious or irrelevant content. In our work, we focus on spamming activity on Twitter. Spamming activity in Twitter is can typically be reported by its users, who we refer as reporters and those who indulge in spamming activities are referred as reportees. We collected data of suspected spammers, i.e. reportees as well as of the users who reported them, i.e. reporters. Thereafter, we classified them into various categories and tried to study the ecosystem of these reportees

and reporters. We used three data mining techniques i.e., decisiontree , K-nearest neighbors and random forest classifier for the classification tasks. Finally, we have compared these three algorithms on the basis of their accuracy.

[5] A Study of the Personalization of Spam Content using Facebook Public Information

Millions of users per day are affected by unsolicited e-mail campaigns. Spam filters are capable of detecting and avoiding an increasing number of messages, but researchers have quantified a response rate of a 0.006% [1], still significant to turn a considerable profit sending millions of emails, as the spammers do. While research directions are addressing topics such as better spam filters, or spam detection inside online social networks (OSNs), in this article, we demonstrate that a classic spam model using OSN information can harvest a 7.62% of click-through rate. We collect email addresses from the Internet, complete email owner information using their public social network profile data, and analyse response of personalized spam sent to users according to their profile using a fake website. Finally, we demonstrate the effectiveness of these profile-based emails to circumvent spam detection and we compare results between typical spam and personalized spam.

CHAPTER 3

METHODOLOGY

3.1 AIM OF THE PROJECT

The objective of the project is to identify the spam based keywords, fake urls, vulgar words, fake information and fake products using big data.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware specification:

Hard Disk	:	4 GB
RAM	:	500 GB
Processor	:	Core i5

3.2.2 Software specification:

Operating system	:	Windows 7
Front End	:	Jdk 7/ 8.1
Back End	:	MySQL

3.3 OVERVIEW OF THE PLATFORM

3.3.1 Java

The Java platform is the ideal platform for network computing. Running across all

platforms -- from servers to cell phones to smart cards -- Java technology unifies business infrastructure to create a seamless, secure, networked platform for your business. The Java platform benefits from a massive community of developers and supporters that actively work on delivering Java technology-based products and services as well as evolving the platform through an open, community-based, standards organization known as the Java Community Process program. You can find Java technology in cell phones, on laptop computers, on the Web, and even trackside at Formula One Grand Prix races. The fact is today, you can find Java technology just about everywhere!

Features of java

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code(class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is considered by many as one of the most influential programming languages of the 20th century, and widely used from application software to web application.

Java programming language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portal
- Distributed
- High performance

Java platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine*
- The *Java Application Programming Interface (API)*

Java virtual machine

A **Java virtual machine (JVM)** is a virtual machine that can execute Java byte code. It is the code execution component of the Java software platform. A Java virtual machine is a program which executes certain other programs, namely those containing Java byte code instructions. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware. A JVM provides an environment in which Java byte code can be executed, enabling such features as automated exception handling, which provides root-cause debugging information for every software error (exception), independent of the source code.

Java API

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

3.3.2 MySQL Overview

What does it do?

- **MySQL is a database management system.**

A database is a structured collection of data. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.

- **MySQL software is Open Source.**

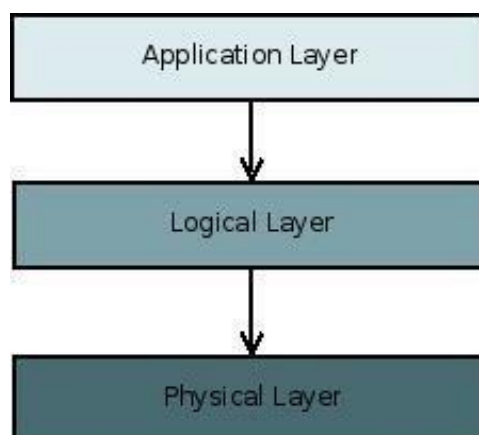
Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. If you want to redistribute altered versions MySQL, other licences are available

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

MySQL Architecture

Three layer model:



- The application layer contains common network services for connection handling, authentication and security. This layer is where different clients interact with MySQL. These clients can be written in different APIs: .NET, Java, C, C++, PHP, Python, Ruby, Tcl, Eiffel, etc...
- The Logical Layer is where the MySQL intelligence resides, it includes functionality for query parsing, analysis, caching and all built-in functions (math, date...). This layer also provides functionality common across the storage engines.
- The Physical Layer is responsible for storing and retrieving all data stored in "MySQL". Associated with this layer are the storage engines, which MySQL interacts with very basic standard APIs. Each storage engine has its strengths and weaknesses, some of these engines are MyISAM, InnoDB, CSV, NDB Cluster, Falcon, etc...

Introduction to Machine Learning using Java

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data. In this chapter, basics of Machine Learning, and implementation of a simple machine learning algorithm using Java are discussed.

3.3.3 Support Vector Machine(SVM) Introduction

Mastering machine learning algorithms isn't a myth at all. It is simple to learn and use, but does that solve our purpose? Of course not! Because you can do so much more than just Regression!

Think of machine learning algorithms as an armoury packed with axes, sword, blades, bow, dagger, etc. You have various tools, but you ought to learn to use them at the right time. As an analogy, think of 'Regression' as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data. On the contrary, 'Support Vector Machines' is like a sharp knife – it works on smaller

datasets, but on the complex ones, it can be much stronger and powerful in building machine learning models

3.3.4 Support Vector Machine Algorithm

Now how would a human being like you and me classify a set of objects scattered on the surface of a table. Ofcourse we will consider all their physical and visual characteristics and then identify based on our prior knowledge. We can easily identify and distinguish apples and oranges based on their colour, texture, shape etc.

Now bringing back the machines, how would a machine identify an apple or an orange. Not surprisingly, it is based on the characteristics that we provide the machine with. It can be size, shape, weight etc. The more features we consider the easier it is to identify and distinguish both.

For the time being, we will just focus on the weight and size(diameter) of apples and oranges. Now how would a machine using SVM, classify a new fruit as either apple or orange just based on the data on the size and weights of some 20 apples and oranges that were observed and labelled?

The objective of SVM is to draw a line that best separates the two classes of data points.

SVM generates a line that can cleanly separate the two classes. How clean, you may ask. There are many possible ways of drawing a line that separates the two classes, however, in SVM, it is determined by the margins and the support vectors.

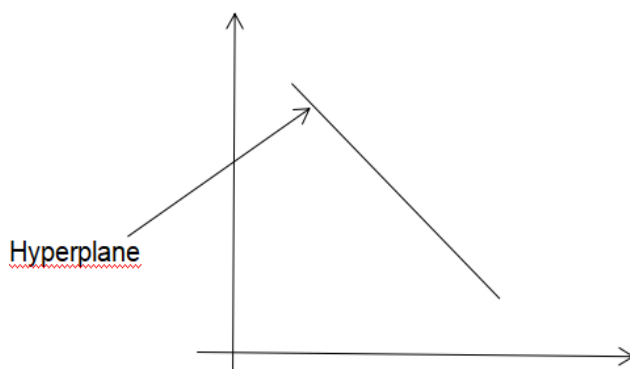
The margin is the area separating the two dotted green lines as shown in the image above. The more the margin the better the classes are separated. The support vectors are the data points through which each of the green lines passes through. These points are called support vectors as they contribute to the margins and hence the classifier itself. These support vectors are simply the data points lying closest to

the border of either of the classes which has a probability of being in either one.

The SVM then generates a hyperplane which has the maximum margin, in this case the black bold line that separates the two classes which is at an optimum distance between both the classes.

In case of more than 2 features and multiple dimensions, the line is replaced by a hyperplane that separates multidimensional spaces.

Fig 3.1 : Steps in SVM



The above image shows the plotting of the training set after fitting the training data to the classifier. The border that separates both the white and black colours represent the Maximum Margin Hyperplane or Line in this case.

According to the SVM classifier, any new data point that falls within the white region is classified as oranges(denoted in orange colour) and any data point that falls on black region is classified as apples(denoted in red colour).

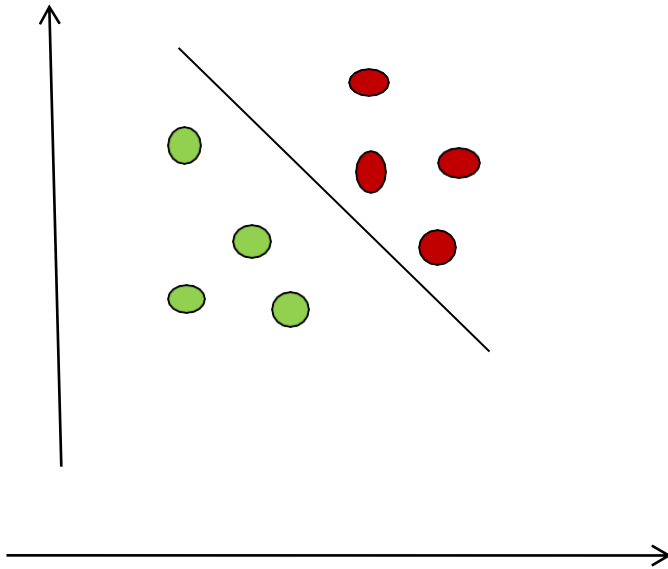


Fig 3.2 : CLASSIFICATION IN SVM

TYPES OF SVM

1. LINEAR SVM
2. NON-LINEAR SVM

LINEAR SVM

Linear SVM is the newest extremely fast machine learning (data mining) algorithm for solving multiclass classification problems from ultra large data sets that implements an original proprietary version of a cutting plane algorithm for designing a linear support vector machine. Our comparisons with other known SVM models clearly show its superior performance when high accuracy is required.

NON - LINEAR SVM

When the data is non-separable these optimization problems are not feasible. Such a problem can be solved by enlarging the feature space through a

function known as the kernel.

Mathematically, a kernel is some function that corresponds to an inner product in some expanded feature space. If every data point is mapped into high-dimensional space via some transformation $\Phi: x \rightarrow \phi(x)$, then the kernel function is defined as:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

3.4 APPLICATIONS OF SVM

1. Inverse Geosounding Problem

The geosounding problem is one of the important SVM applications that helps to determine the layered structure of the planet. It demands solutions of various inversion problems. In the case of electromagnetic data, we use a linear function for the problem and we use the SV learning algorithm for models. The models are developed here with linear programming Techniques. But, the size of the dimension

in which the model is developed might be small here, as the size of the problem is also small.

2. Seismic Liquefaction Potential

Here SVM is considered as a great *Machine Learning algorithm* because the result accuracy is great. We have two problems namely SPT (Standard Penetration Test) **and** CPT (Cone Penetration Test). In both of these tests, we check the occurrence and non-occurrence of liquefaction. Here, we use field datasets. The accuracy achieved with the datasets using SVM was 96 and 97 % respectively. Here, SVM helps to build a model for complex combinations like soil parameters and liquefaction potentials.

3. Protein Fold and Remote Homology Detection

Remote homology detection makes great use of SVM. In this case, it depends on how the protein sequences are modelled. We use different methods to solve the kernel functions that are being used. The kernel functions help to find the similarity between different protein sequences. The truth is, many times these kernels produce better results than many SVM based techniques. Kernel is actually a part in SVM itself. In this way, SVM is influencing areas like computational biology.

4. Data Classification using SVM

Here, SVM helps in a lot of mathematics based problems. We use various smoothing methods here for solving various math problems. There is also a slight change, we won't use the normal SVM method here. We will use the **SVM**.

SVM makes classification easier as it possesses a strong convexity (a math property). Also, the use of kernel function here is arbitrary. For SVM's unconstrained optimization problem, we use the Newton-Armijo algorithm here. SVM works faster on larger datasets than the normal SVM. It could also generate checkerboard shaped nonlinear dividing surface.

5. Facial Expression Classification

There are many ways to use this classification. This can be used in various life-care systems, we can use it in normal happy or sad look classification as well. We can use it in filters. If we make certain facial expressions, it would add the specific filter as per the expression. The range of expressions lies between happy and sad. SVM can be of great use here.

6. Texture Classification using SVM

In this SVM application, we use the words of certain classes and use that data to classify whether the surface is smooth or not. This SVM application will be of great use. If there is a sensitive camera to take pictures and use that data in our model, we could train a really powerful model. Also, if we take images of surfaces, we could determine the surface as smooth or gritty. SVM here classifies the surface as smooth or gritty.

7. Text Classification

We could use SVM in differentiating between the handwritings of two different people. The dataset would contain images of various alphabets and sentences written in both handwriting. The SVM classifier would be trained using the sample data. We can also use SVM to differentiate between human writing and computer alphabets.

8. Speech Recognition

Using speech recognition, we could make an interface for communication with deaf people. Here, the acoustics would be the data. There are many functions like LPC, LPCC, and MFCC which collect the acoustic data. The SVM uses the acoustic data to train its models.

9. Stenography Detection in Digital Images

Using SVM, it is easy to find out if an image is pure or adulterated. This could be used in security-based organizations to uncover secret messages. Yes, we can encrypt messages in high-resolution images. In high-resolution images, there are more pixels, hence, the message is more hard to find. We can segregate the pixels and store in data in various datasets. We can analyze those datasets using SVM.



it is easy to find out if an image is pure or adulterated. This could be used in security-based organizations to uncover secret messages. Yes, we can encrypt messages in high-resolution images. In high-resolution images, there are more pixels, hence, the message is more hard to find. We can segregate the pixels and store in data in various datasets. We can analyze those datasets using SVM.

SYSTEM ARCHITECTURE

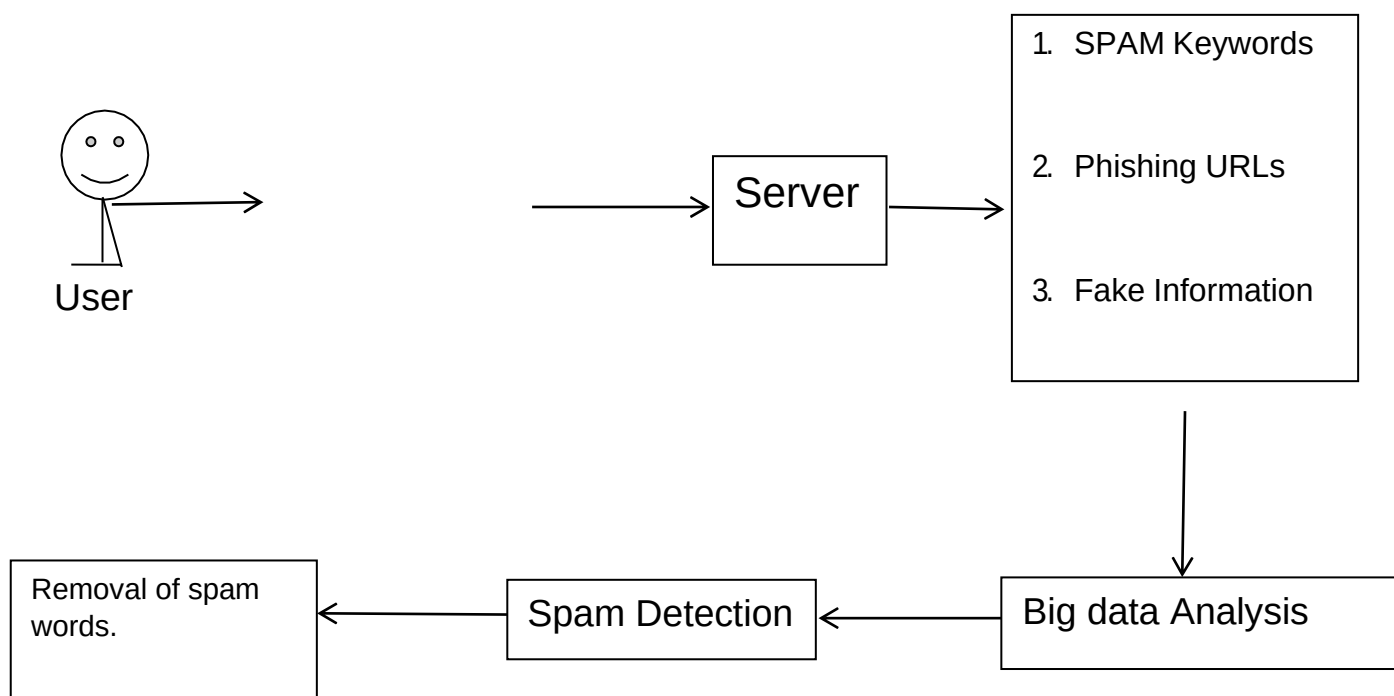


Fig 3.3:Architecture

CHAPTER 4

MODULE DESCRIPTION

4.1 SYSTEM STUDY

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of the system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comprehensibility, Modular Understandability, Modular continuity, Modular Protection.

The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

Generally testing process is carried out to ensure that the system has been developed according to the required specifications and the expected output is properly obtained. There are two main categories of testing namely, the White Box Testing and the Black Box Testing. Each of this testing in turn consists of many types of testing.

4.2 MODULE LIST

1. User Interface design
2. Server
3. Spam Keyword training
4. Spam detection

5. Phishing URL detection
6. Fake product & agencies detection
7. Fake information detection

1. USER INTERFACE DESIGN

In this Module, user interface module is designed using web application using Java. The user has to be registering their vehicle details. Such as Name, Mobile number, E mail ID and other credentials. All the data provided by the user is stored in the centralized server.

2. SERVER

In this module, all the Data are stored in the centralized server. All the user information as well as private credential where stored in the centralized server. Whenever it is needed all the data or all the information is used for verification / authentication. We use MySQL as the database part.

3. SPAM KEYWORDS TRAINING

In this module, Spam keywords are stored and kept for data analysis and comparison. The Spam words can be added whenever it is needed for fetching more accuracy. All the spam keywords are used for comparison component of analysis from the user input. This module helps to train the spam keywords so that our application detects and separates the spam contents from the posting made by the user.

4. SPAM DETECTION

In this module, after training of spam keywords the server automatically detects the spam words which are present in the content posted by the user. We deploy detection module through Naïve Bayes Classifier algorithm. Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use bag of words features to identify spam e-mail, an approach commonly used in text classification.

5. PHISHING URL DETECTION

In this module, Holistic approach is used for detection of phishing URLs. We implement this approach by comparing with the Dataset. Once we find the requested URL is present in the phishing set URLs then the requested URL is blocked by the server. URL is standard URL link and Sub links are the vulnerable keywords which are followed after com or org or in or co.in and so on

6. FAKE PRODUCTS & AGENCIES DETECTION

In this module, Fake products and fake Agencies are detected through the multiple feedbacks posted by the different users into the server. User's Feedbacks are analyzed through our application and fake products and agencies are detected and notified to the new users. Based on the feedbacks fake products and agencies can be omitted from the list, so that user may not be shown with those fake products and agencies.

7. FAKE INFORMATION DETECTION

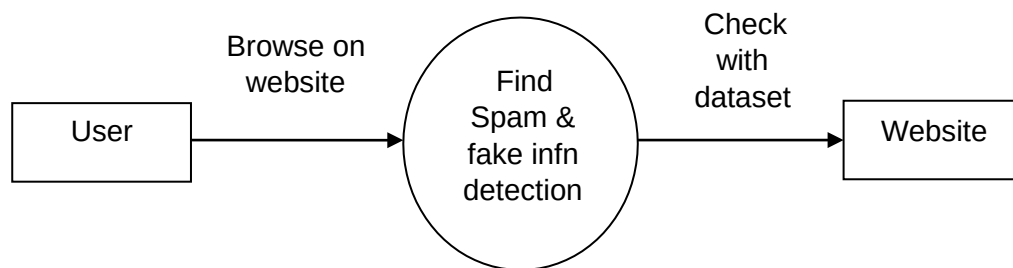
In this module, Fake information is detected and then deletes from the fake information from the respective server. We deploy effective Fake Information detection from the centralized server by analyzing the posted data with comparison with number of likes and dislikes of the same data by users. This is an effective system to remove all the unwanted data from the social media.

4.3 DATA FLOW DIAGRAM

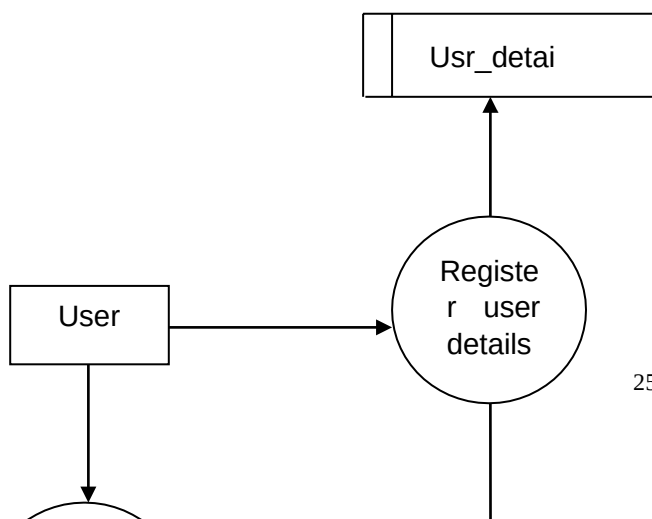
A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation. Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.

LEVEL 0:



LEVEL 1:



4.4 UML DIAGRAM:

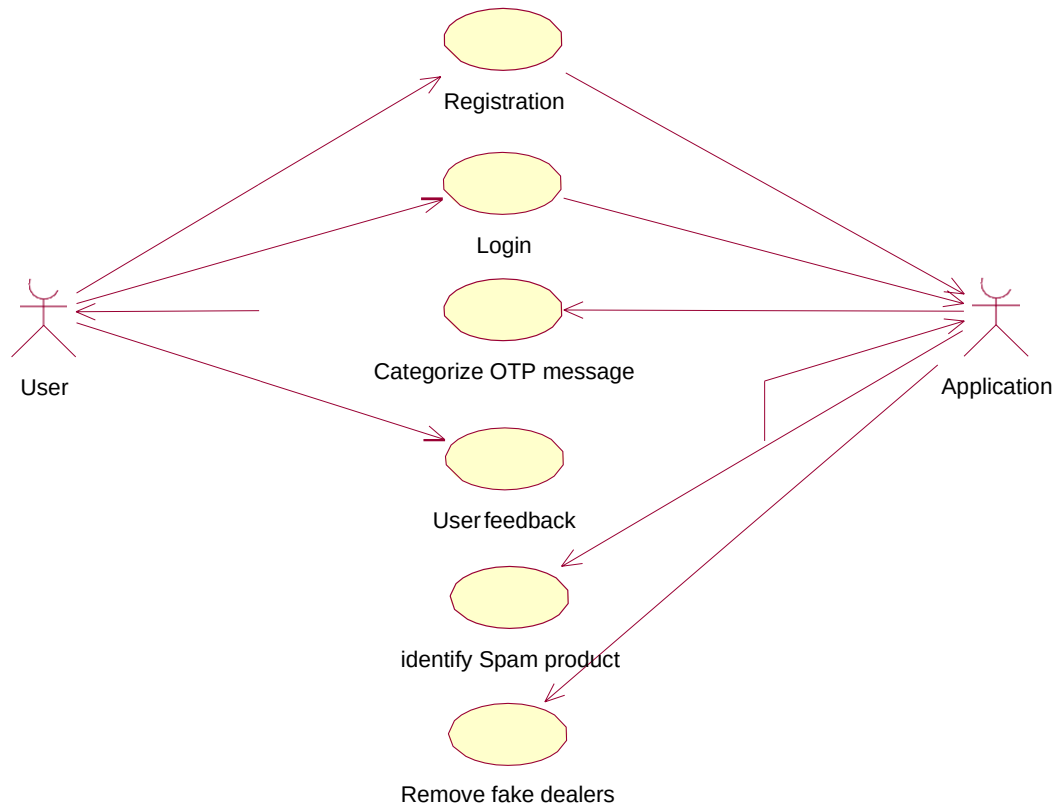
The Unified Modeling Language (UML) was created to forge a common, semantically and syntactically rich visual modeling language for the architecture, design, and implementation of complex software systems both structurally and behaviorally. UML has applications beyond software development, such as process flow in manufacturing. It is analogous to the blueprints used in other fields, and consists of different types of diagrams. In the aggregate, UML diagrams describe the boundary, structure, and the behavior of the system and the objects within it.

UML is not a programming language but there are tools that can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design.

4.4.1 USE CASE DIAGRAMS:

A use case is a list of steps that define interaction between an actor (a human who interacts with the system or an external system) and the system itself. Use case diagrams depict the specifications of a use case and model the functional units of a system. These diagrams help development teams understand the requirements of their system, including the role of human interaction therein and the differences between various use cases. A use case diagram might display all use cases of the system, or just one group of use cases with similar functionality.

1. To begin a use case diagram, add an oval shape to the center of the drawing.
2. Type the name of the use case inside the oval.
3. Represent actors with a stick figure near the diagram, then use lines to model relationships between actors and use cases.

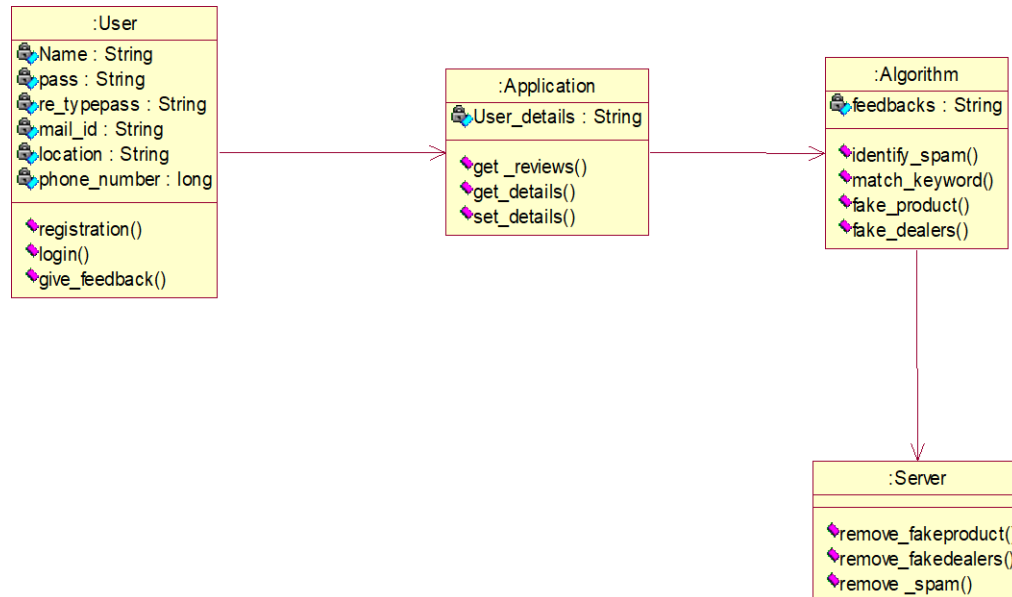


4.4.2 CLASS DIAGRAM:

Class diagrams represent the static structures of a system, including its classes, attributes, operations, and objects. A class diagram can display computational data or organizational data in the form of implementation classes and logical classes, respectively. There may be overlap between these two groups.

- 1 Classes are represented with a rectangular shape that is split into thirds. The top section displays the class name, while the middle section contains the class' attributes. The bottom section features the class operations (also known as methods).
- 2 Add class shapes to your class diagram to model the relationship between those objects. You may need to add subclasses, as well.

- 3 Use lines to represent association, inheritance, multiplicity, and other relationships between classes and subclasses. Your preferred notation style will inform the notation of these lines.

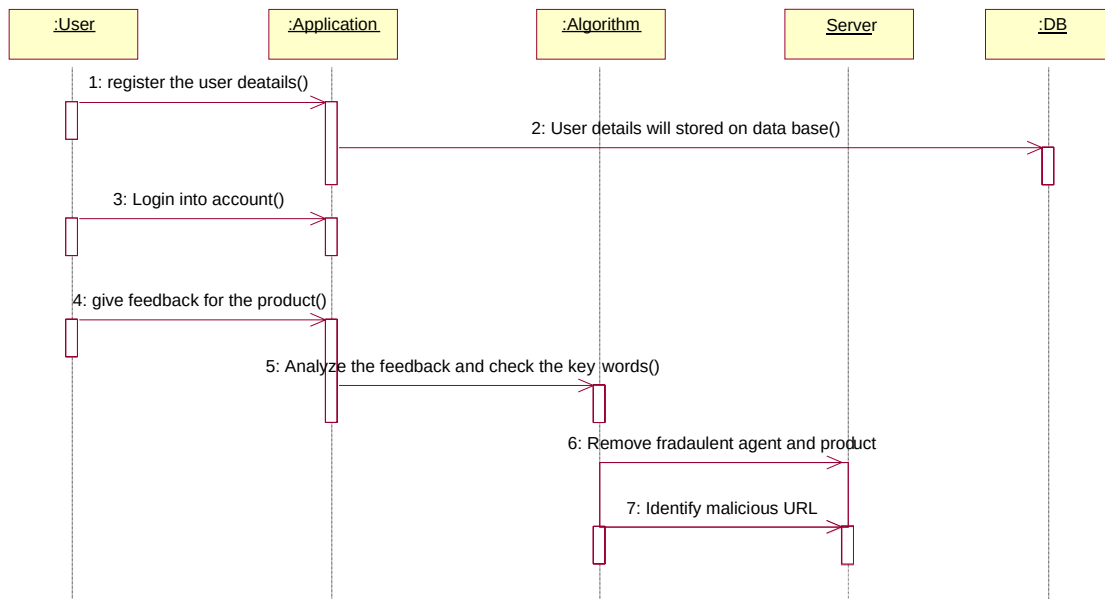


4.4.3 SEQUENCE DIAGRAM :

Sequence diagrams, also known as event diagrams or event scenarios, illustrate how processes interact with each other by showing calls between different objects in a sequence. These diagrams have two dimensions: vertical and horizontal. The vertical lines show the sequence of messages and calls in chronological order, and the horizontal elements show object instances where the messages are relayed.

1. To create a sequence diagram, write the class instance name and class name in a rectangular box.
2. Draw lines between class instances to represent the sender and receiver of messages.

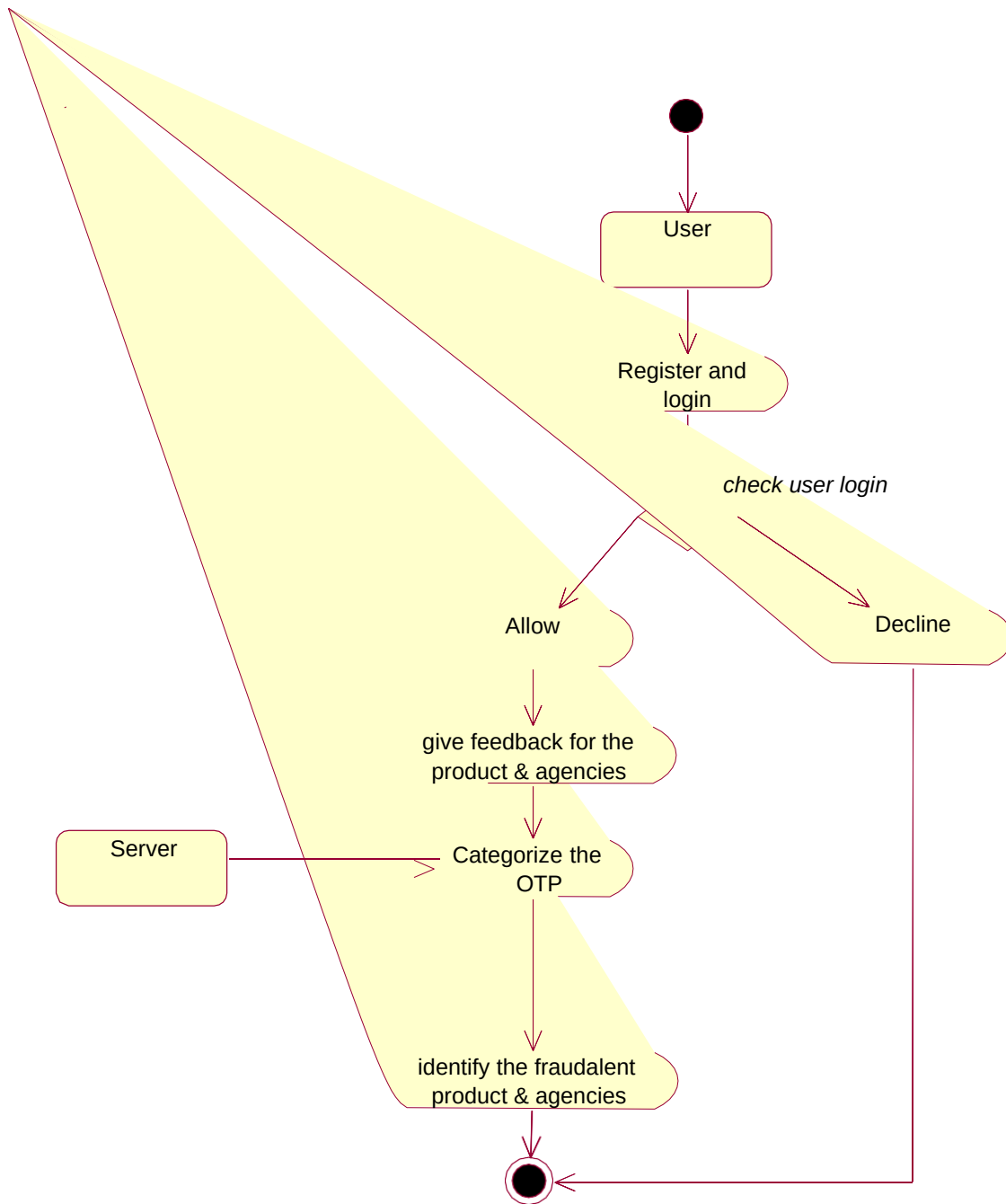
Use solid arrowheads to symbolize synchronous messages, open arrowheads for asynchronous messages, and dashed lines for reply messages.



4.4.4 ACTIVITY DIAGRAM:

Activity diagrams show the procedural flow of control between class objects, along with organizational processes like business workflows. These diagram are made of specialized shapes, then connected with arrows. The notation set for activity diagrams is similar to those for state diagrams.

1. Begin your activity diagram with a solid circle.
2. Connect the circle to the first activity, which is modeled with a round-edged rectangle. Now, connect each activity to other activities with lines that demonstrate the stepwise flow of the entire process.
3. You can also try using swimlanes to represent the objects that perform each activity.



CHAPTER 5

RESULTS AND DISCUSSION

The main motive for this spam stop is to give users a decent and good platform to post the usefull posts that can help others to find it in a easy way. So that the users can be aware of different types frauds in this respective social media. This project can be implemented in various social media applications that occur with various frauds and spam messages.

The results of this project is to give user a social media application that is free from spam and fraud messages. This helps the users whether the information that he/she is willing to post is spam or not.

In this project, we have developed various models so that various spams can be detected using these models. Each model consists of different spams. So that various frauds and spams can be detected and deleted from the respective server.

In this project, we have created a social media web page. Now the user can register his details as he registers in various applications and sign in to the web page application with his respective email id and password. Now the can post an message that he is willing to post. If that respective post contains any fake url, spam keywords, vulgar words, fake information the respective content will be detected and deleted by the admin in the respective account. But for the vulgar words the content will not be posted and gives you a display message as it contains vulgar words. For fake information it is categorized as likes and dislikes if dislike is greater than like then the respective fake information post will be deleted. Here the detection of spam is done using datasets of various spam that we provided in the source code.

CHAPTER 6

CONCLUSION AND FUTUREWORK

We Have deployed social media like web page to deduct the malicious URLs and dealers so as to ensure proper and genuine product sales through social media we also identify the fake dealers and fake products through this system. We deploy android based application to cluster the messages based on keywords.

The final outcome of the project is to detect the Spam words in the social media, removal of vulgar words posted by any user. Apart from this our application also detects phishing URLs which has to be notified to the user. As well as our application detects the Fake products and Organization along with the Fake information / Rumor data posted in the social media. Android application is also created to segregate the SMS messages .

The future work for this project is to collect the spam keywords, phishing urls, vulgar words and fake information, products and update it in the database dynamically without the interruption of admin and also this project can be implemented in real time social media application so that it can reduce lot of spam undergoing various social media applications and make these social media applications free from spam.

REFERENCES

- [1] S. Heron. Technologies for spam detection. Network Security, Vol. 2009: Pages 11 – 15, 2009.
- [2] D. Wang, D. Irani, and C. Pu. A study on evolution of email spam over fifteen years. In 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), pages 1–10. IEEE, 2013.
- [3] Kaspersky Lab, “Spam and Phishing Statistics Report Q1-2014”. [Online]. Available: <https://usa.kaspersky.com/internet-securitycenter/threats/spam-statistics-report-q1-2014/>. Accessed: 2016-06-2.
- [4] Kaspersky Lab, “Spam and Phishing Statistics for 2016”. [Online]. Available: <https://www.kaspersky.com/about/pressreleases/2016/kaspersky-lab-reports-significant-increase-in-malicious-spam-emails-in-q1-2016>. Accessed: 2016-06-2.
- [5] P. Wood, B. Nahorney, K. Chandrasekar, S. Wallace, and K. Haley. Symantec internet security threat report. Vol. 21: pages 27–36, 2016.
- [6] G. Cormack. Email spam filtering: A systematic review. Foundations and Trends in Information Retrieval, Vol. 1: Pages 335–455, 2007.
- [7] A. Khraisat, A. Alazab, M. Hobbs, J. Abawajy, and A. Azab. “Trends in Crime Toolkit Development” in Network Security Technologies: Design and Applications: Design and Applications. IGI Global. ch2, pages 28-43, 2014.
- [8] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna. The underground economy of spam: A botmaster’s perspective of coordinating large-scale spam campaigns. USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), Vol. 11: Page 4, 2011.
- [9] M. Crawford T. Khoshgoftaar J. Prusa. “survey of review spam detection using machine learning techniques”. Journal Of Big Data, Vol. 2:pages 23, 2015.

[10] M. Sheikhalishahi, A. Saracino, M. Mejri, N. Tawbi, and F. Martinelli. Fast and effective clustering of spam emails based on structural similarity. In International Symposium on Foundations and Practice of Security, pages 195–211. Springer, 2015.

APPENDIX

A.SOURCE CODE

User registration:

```
public class UserRegistration extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            UserDetails _userDtls = new UserDetails();
            _userDtls.setAboutU(request.getParameter("about_you"));
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");
            Date date = null;
            try {
                date = sdf.parse(request.getParameter("dob"));
            } catch (ParseException ex) {
```

```

Logger.getLogger(UserRegistration.class.getName()).log(Level.SEVERE, null, ex);

    }

    _userDtls.setDateOfBirth(date);

    _userDtls.setGender(request.getParameter("gender"));

    _userDtls.setLivesIn(request.getParameter("lives_in"));

    _userDtls.setMobileNumber(request.getParameter("mobile_no"));

    _userDtls.setPassword(request.getParameter("password"));

    _userDtls.setStudiedAt(request.getParameter("studied_at"));

    _userDtls.setUserName(request.getParameter("username"));

    _userDtls.setWorkedAt(request.getParameter("worked_at"));

    _userDtls.setBluetoothNo("");

    _userDtls.setEdu(request.getParameter("edu"));

    String[] userLikes = request.getParameterValues("user_likes");

    _userDtls.setName(request.getParameter("name"));

    File file = new File(constants.Constants.USER_LIST_FILE_LOC);

    if (!file.exists()) {

        file.createNewFile();

    }

    FileWriter fileWriter = null;

    try {

        fileWriter = new FileWriter(file, true);

        String json = new Gson().toJson(_userDtls);

```

```

        System.out.println("User registered json=>" + json);

        fileWriter.write(json + "\n");

    } catch (Exception ex) {

        ex.printStackTrace();

    } finally {

        fileWriter.flush();

        fileWriter.close();

    }

}

//com.nura.dropbox.AuthenticateUser.upload(file.getAbsolutePath(),
constants.Constants.DROPBOX_LOC);

try {

    for (Part parts : request.getParts()) {

        System.out.println("Parts");

        InputStream inputStream = parts.getInputStream();

        byte[] getImgytes = new byte[inputStream.available()];

        inputStream.read(getImgytes);

        String fileName = getFileName(parts);

        System.out.println("File name:-" + fileName);

        if (fileName != null) {

            _userDtls.setUserImage(getImgytes);

        }

    }

}

```

```

        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    if (new UserDetailsDAOImpl().saveUserDtls(_userDtls)) {
        try {
            for (String like : userLikes) {
                UserLikes _userLike = new UserLikes();
                _userLike.setLikeDtls(like);
                _userLike.setUserDtls(_userDtls);
                new UserLikeDAOImpl().saveUserLike(_userLike);
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }

        response.sendRedirect("loginPage.jsp?msg=User Details saved");
    } else {
        response.sendRedirect("loginPage.jsp?msg=Alert!Unable      to
save.Contact Admin");
    }
}
}
}

```

```
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on  
the + sign on the left to edit the code.">
```

```
@Override
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
    processRequest(request, response);
```

```
}
```

```
@Override
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
    processRequest(request, response);
```

```
}
```

```
@Override
```

```
public String getServletInfo() {
```

```
    return "Short description";
```

```
}// </editor-fold>
```

```
// Extract file name from content-disposition header of file part
```

```

private String getFileName(Part part) {

    final String partHeader = part.getHeader("content-disposition");

    System.out.println("***** partHeader: " + partHeader);

    for (String content : part.getHeader("content-disposition").split(";")) {

        if (content.trim().startsWith("filename")) {

            return content.substring(content.indexOf('=') + 1).trim()

                .replace("\\'", "");

        }

    }

    return null;

}
}

```

ValidateUser:

```

public class ValidateUser extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        HttpSession session = request.getSession();

        try (PrintWriter out = response.getWriter()) {

            /* TODO output your page here. You may use following sample code. */

```

```

        if (request.getParameter("username").equals("admin@gmail.com") &&
request.getParameter("password").equals("admin")) {

            response.sendRedirect("adminPage.jsp");

        } else {

            UserDetails _userDtls = new
UserDetailsDAOImpl().validateUserDetails(request.getParameter("username"),
            request.getParameter("password"));

            if (_userDtls != null) {

                session.setAttribute(constants.Constants.SESSION_USER_OBJ,
_userDtls);

                response.sendRedirect("homePage.jsp");

            } else {

                response.sendRedirect("loginPage.jsp?msg=Invalid
Cedentials");}}}

```

B.SCREENSHOTS

VULGAR words detection and deletion

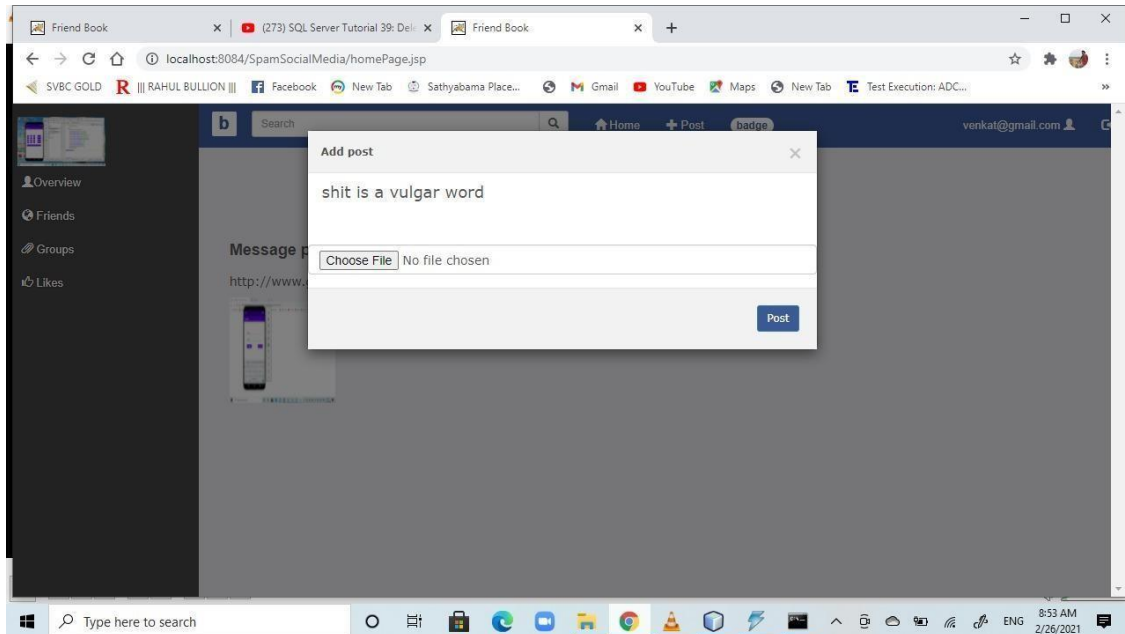


FIG User enters a VULGAR word

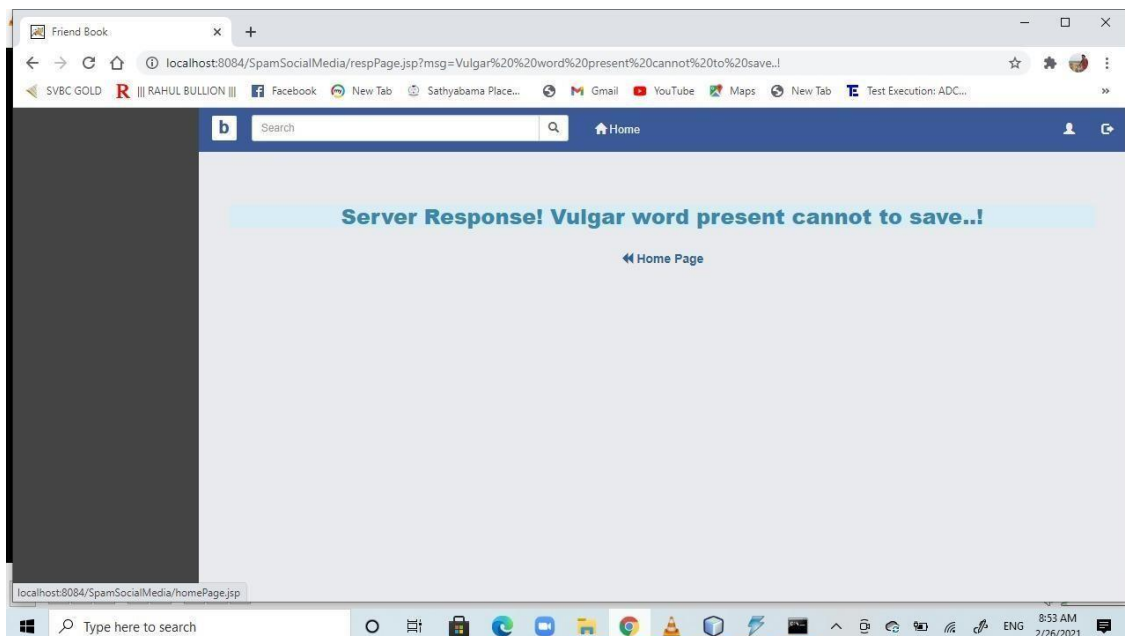


FIG:Server detecting and deleting words

URL detection(based on feedback)

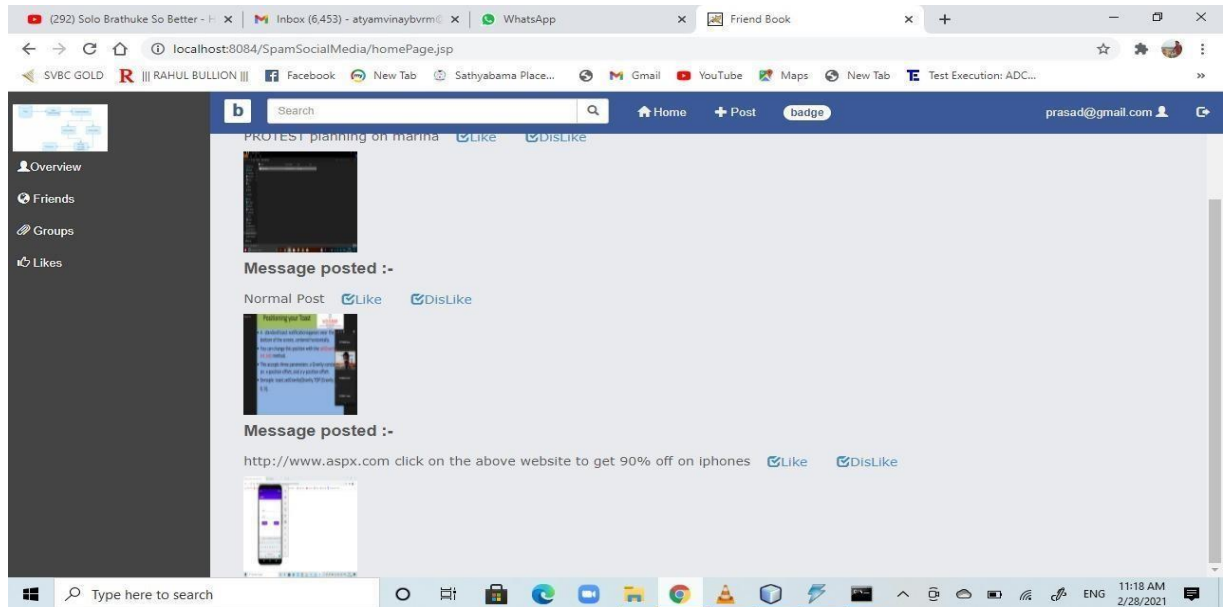


FIG :user posted a picture

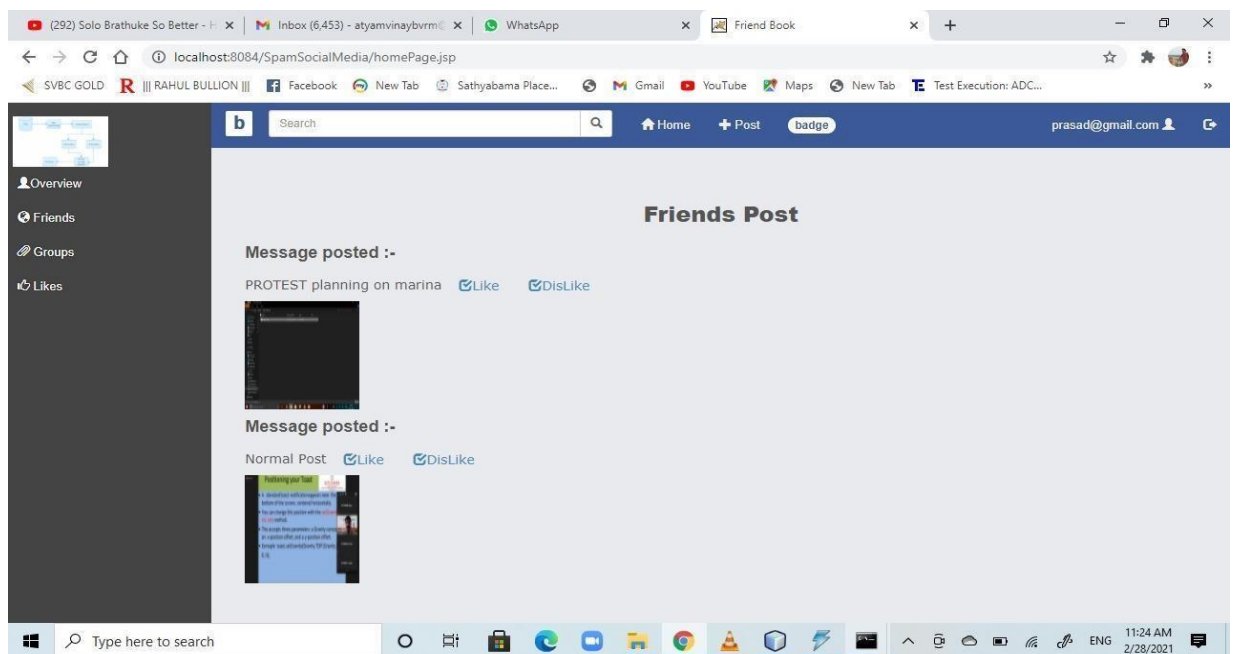


FIG :Link has been deleted due to dislikes

Spam keywords detection

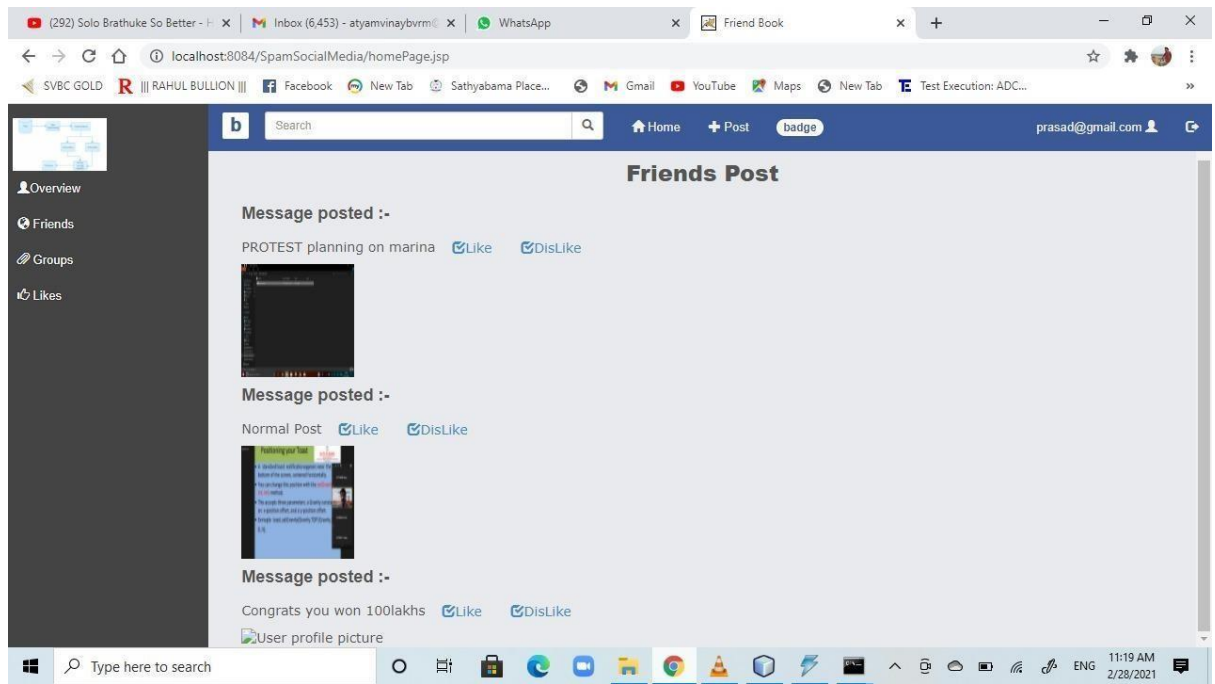


FIG:user posted spam keyword

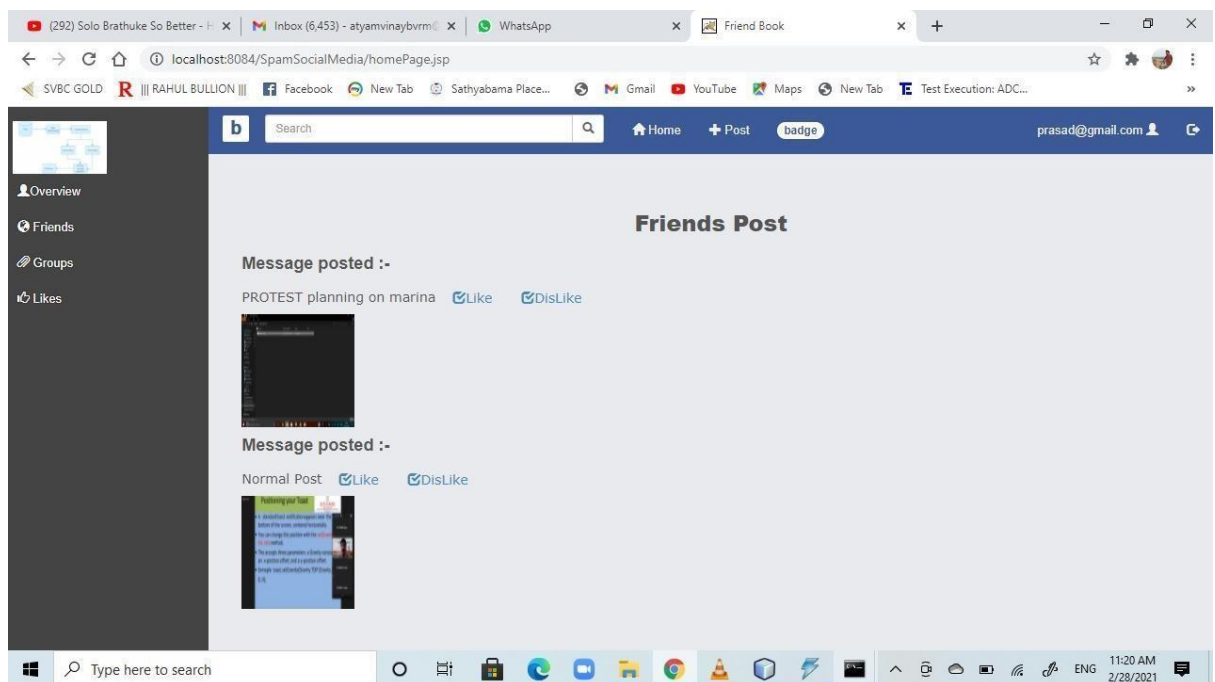


FIG:spam detected and deleted

Fake information detection

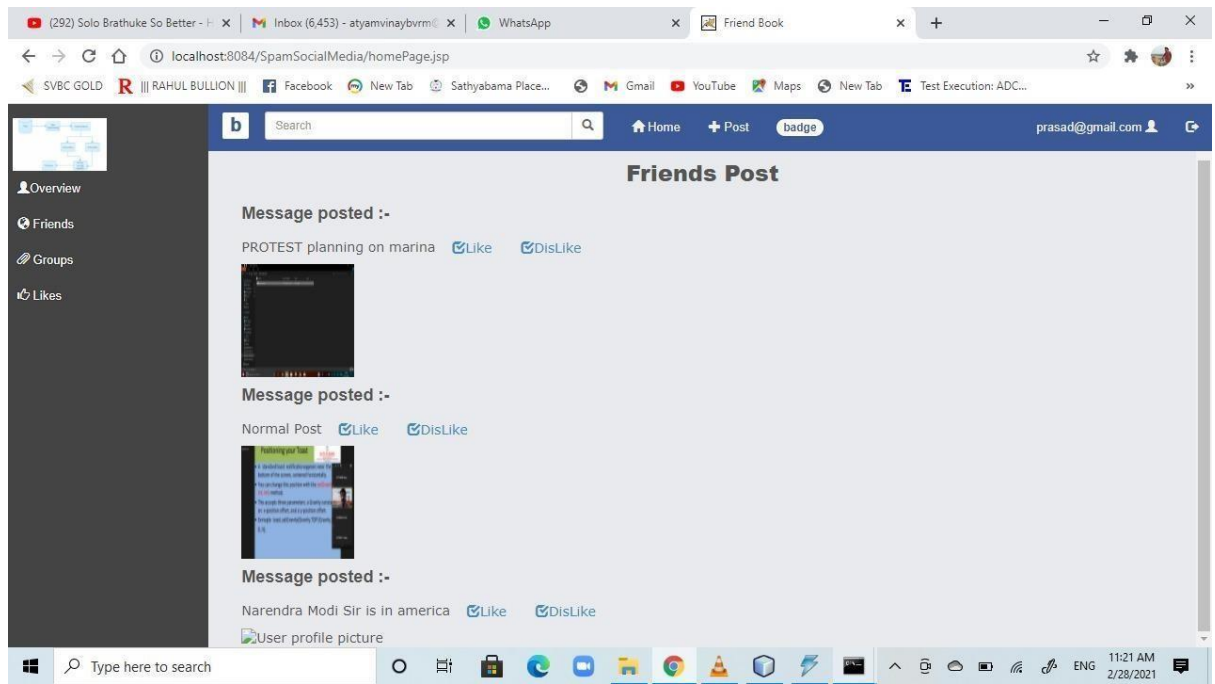


FIG : User posted a message

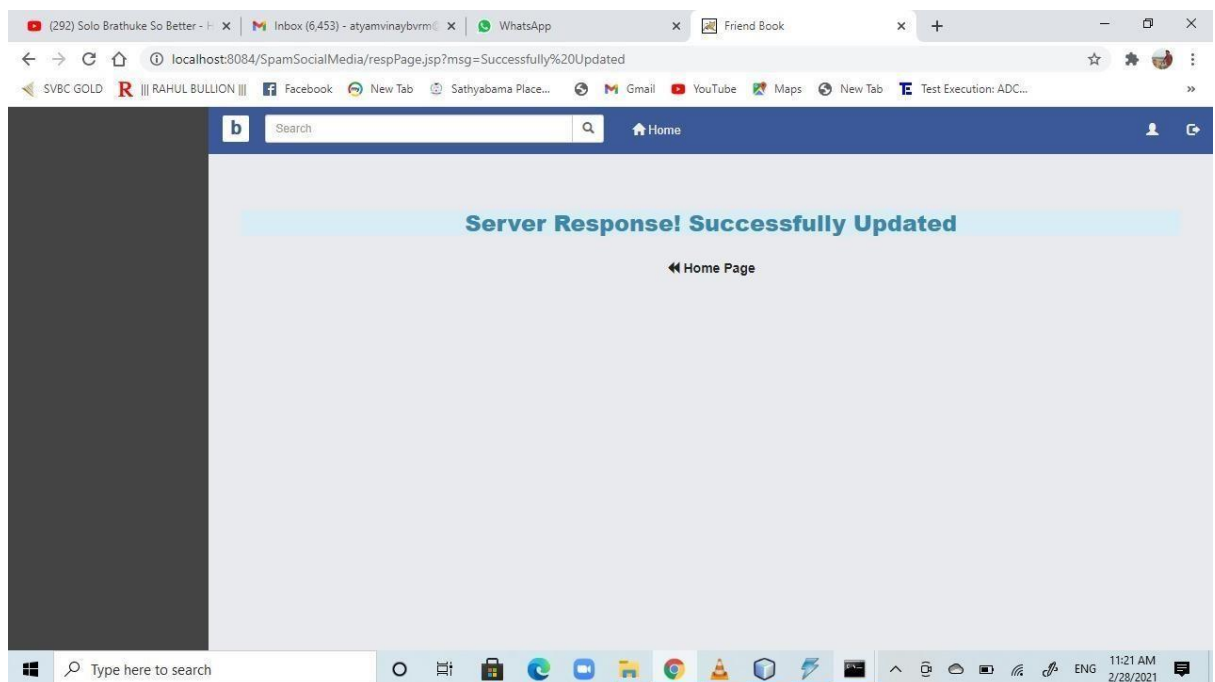


FIG :information detected and server responded

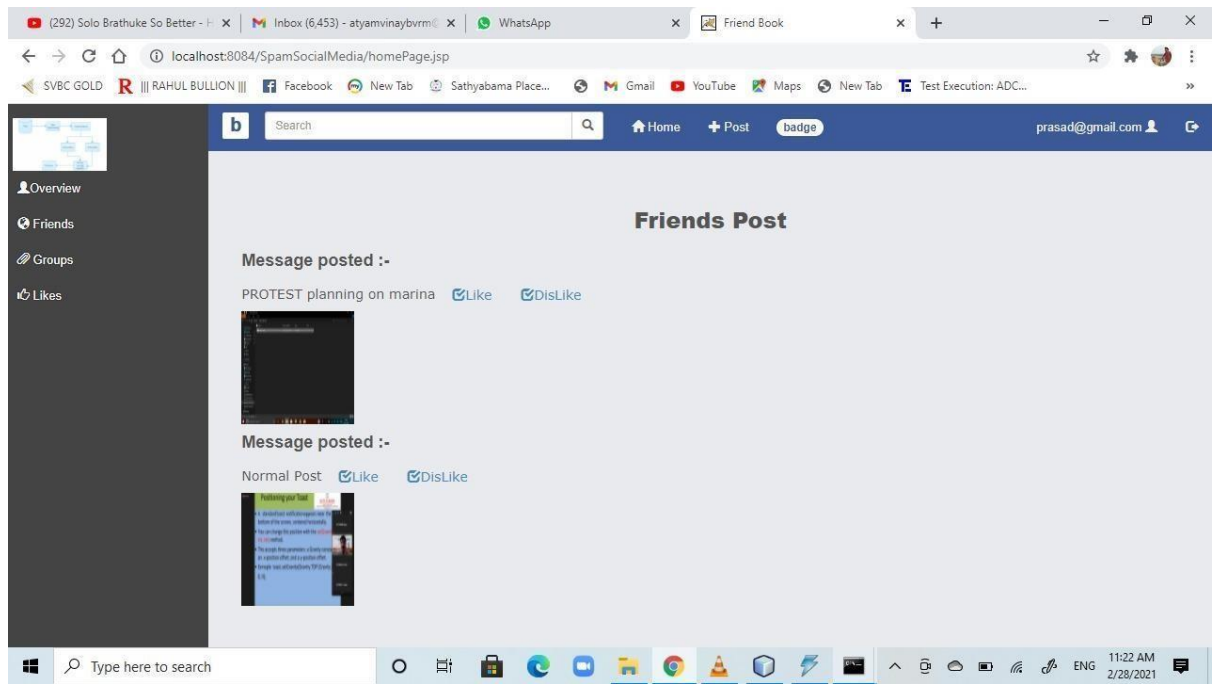


FIG : Information will be deleted if it is fake

PAPER PUBLICATION

SPAM STOP:SPAM IDENTIFICATION AND DELETION

Authors:

ATYAM VINAY

AVULA MANOJ REDDY

Under the guidance of :

Dr. S. Vigneshwari, ME., PhD.D.

Abstract:

In the Existing System, Spam has become the platform of choice used by cyber-criminals to spread malicious payloads such as viruses and Trojans. In the Proposed system, the problem of early detection of spam campaigns. Collaborative spam detection can deal with the e-mail data contributed by multiple sources; however, they have the well-known problem of requiring disclosure of e-mail content. hashes are one of the common solutions that are used for preserving the privacy of e-mail content while enabling message classification for spam detection. In the Modification process, users can create a web page that principally categorizes all the messages based on the keyword alliteration like banking category based on the keywords like OTP, Bank ID, and so on. So apart from spam filtering, we can also implement an overall filtering-based concept in this project.

Introduction:

The word spam was used to describe e-mails sent in bulk.

Some spam is about the lack of consent on the part of the recipient, while others believe it is about unsolicited e-mail quantity or scale.

LITERATURE SURVEY

Paper 1: Trends in Crime Toolkit Development

Author: Transatlantic, Michael Hobbs, Taramasalata, Majestically, Ahmad Ahab

Abstract: Criminals continue to target online users of banks. They are improving their techniques and using a high level of skills in their attacks. Their continued search for the different methods to commit crime makes the

existing protection system less effective. They have developed crime tool kits that have become more accessible and simpler to use, and this has attracted more criminals to cybercafes. In this chapter, the authors study the methods that are used in crime tool kits. They present the development and current trend of crime tool kits and reveal the methods that have been used to commit cybercafes successfully.

Paper 2: Survey of review spam detection using machine learning techniques

Author: Michael Crawford, Tag hi M. Sharpshooter, Joseph D. Prussia, Aaron N. Richter, and Yamaha Al Adana

Abstract: Online reviews are often the primary factor in a customer's decision to purchase a product or service and are a valuable source of information that can be used to determine public opinion on these products or services. Because of their impact, manufacturers and retailers are highly concerned with customer feedbacks and reviews. Reliance on online reviews gives the rise to the potential concern that wrongdoers may create false reviews to artificially promote or devalue products and services. This practice is also known as Opinion Spam, where spammers are going to manipulate and poison reviews (i.e., making fake, untruthful, or deceptive reviews) for profit or gain. Since not all online reviews are trustworthy and trustful, it is important to develop techniques for detecting review spam. By extracting

the meaningful features from these texts using Natural Language Processing (NLP), it is possible to conduct review spam detection using various machine learning techniques. Additionally, reviewer information, apart from these texts themselves, can be used to aid in this process. In this paper, we survey prominent machine learning techniques that have been proposed to solve the problem of review spam detection and the performance of different approaches for classification and detection of review spam. The majority of current research has focused on the supervised learning methods, which require labeled data, a scarcity when it comes to online review spam. The primary goal of this paper is to provide a strong and comprehensive comparative study of current research on detecting spam using various machine learning techniques and to devise a methodology for conducting further investigation.

Paper 3: Conventional and Ontology-Based Spam Filtering

Author: Nature M Handmaiden, Bind V

Abstract: Emails are inevitable in this modern era. It has become the most important tool in the communication field. Because of its easiness, the number of users is increasing day-by-day. With the increased number of email users, the number of spam emails has also increased. Spam can cause great loss to users. Many spam filtering techniques have been introduced to distinguish between ham

and spam. A mail that appears as spam may appear as ham to another user and vice versa. That is, it depends on the personal preference of the user. Therefore ontology-based personalized mail access is necessary. This paper compares ontology-based spam filtering and conventional spam filtering.

Paper 4: Ecosystem of Spamming on Twitter: Analysis of Spam Reporters and Spam Deportee

Author: Popinjay, Macintosh, Gunman Mali, and Imperishably

Abstract:

Lately, there has been a growing trend in the Internet space particularly among the Online Social Media (OSMs) platforms like Twitter, Facebook, etc which are becoming huge repositories of information. This information, by design, is posted by users of these websites and consequently, this information is vast, UN-organized, unreliable, and dynamic. A lot of activity is seen from spammers or users with the intent of spreading malicious or irrelevant content. In our work, we focus on spamming activity on Twitter. Spamming activity in Twitter is typically reported by its users, whom we refer to as reporters, and those who indulge in spamming activities are referred to as deportees. We collected data of suspected spammers, i.e. deportee as well as of the users who reported them, i.e. reporters. Thereafter, we classified them into various categories and tried to study the ecosystem of this deportee and reporters. We used three data mining

techniques i.e., decision, K-nearest neighbors, and random forest classifier for the classification tasks. Finally, we have compared the three algorithms on the basis of their accuracy.

Paper 5: A Study of the Internalization of Spam Content using Facebook Public Information

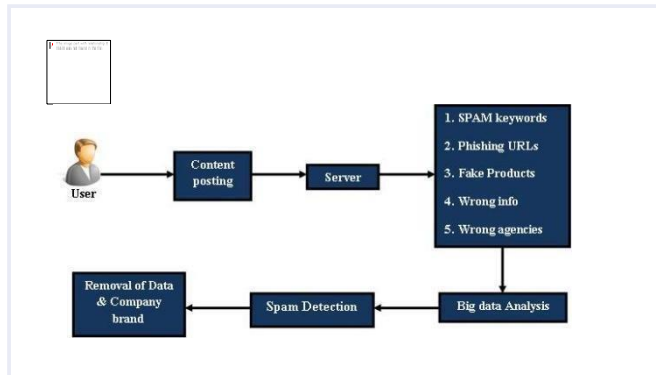
Author: ENAITZ EZPELETA and URKO ZURUTUZA,

Abstract:

Millions of users per day and day by

day are affected by unsolicited e-mail campaigns. Spam filters are capable of detecting and avoiding an increasing number of messages, but researchers have quantified a response rate of 0.006% [1], still significant to a considerable profit sending millions of emails, as the spammers do. While research directions are addressing topics such as better spam filters, or spam detection inside online social networks (OSNs), in this article, we demonstrate that a classic spam model using OSN pieces of information can harvest a 7.62% of click-through rate. We collect email addresses from the Internet, complete email owner information using their public social network profile data, and analyze responses of personalized spam sent to users according to their profile using a fake website. Finally, we demonstrate the effectiveness of this profile-based on emails to circumvent spam detection and we compare results between typical spam and personalized spam.

ARCHITECTURE DIAGRAM:



PROPOSED SYSTEM:

In the proposed system, user can create a web page which principally categories all the messages based on the keyword filtration category based on the keywords like vulgar words, spam words and so on. So apart from spam filtering we can also implement overall filtering based concept also in this project.

ADVANTAGES:

- Spam word detection
- Vulgar words detection and deletion
- Fake agencies identification

EXISTING SYSTEM:

Spam has become the platform of choice used by cyber-criminals to spread malicious payloads such as viruses and Trojans .

MODULES

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of the system. Software with effective modularity is easier to develop because the function may be compartmentalized and interfaces are simplified.

- Modularity is a single software that allows a program to be intellectually manageable. The five important criteria that enable us to

evaluate a design method concerning its ability to define an effective modular design are Modular comparability, Modular Compatibility, Modular Understaffed, Modular continuity, Modular Protection.

- The following are the modules of the project, which are planned in aid to complete the project concerning the proposed system while overcoming the existing system and also providing the support for the future enhancement.

MODULE LIST:

1. User Interface design
2. Server
3. Spam Keyword training
4. Spam detection
5. Phishing URL detection
6. Fake product & agencies detection
7. Fake information detection

1. USER INTERFACE DESIGN

In this module, the user interface module is designed using a web application using Java. The user has to be registering their vehicle details. Such as Name, Mobile Number, Email ID, and other credentials. All the data provided by the user is stored in the centralized server.

2. SERVER

In this module, all the Data are stored in the centralized server. All the user information, as well as private credentials, were stored in the centralized server. Whenever it is needed all the data or all the information is used for verification/authentication. We use Mys as the database part.

3. SPAM KEYWORDS TRAINING

In this module, Spam keywords are stored

and kept for data analysis and comparison. The Spam words can be added whenever it is needed for fetching more accuracy. All the spam keywords are used for the comparison component of analysis from the user input. This module helps to train the spam keywords so that our application detects and

separates the spam contents from the posting made by the user.

4. SPAM DETECTION

In this module, after training spam keywords the server automatically detects the spam words which are present in the content posted by the user. We deploy the detection module through the Naive Bayes Classifier algorithm. Bayes classifiers are a popular technique for e-mail filtering. They use a bag of features and words to identify spam e- mail, an approach commonly used in text classification.

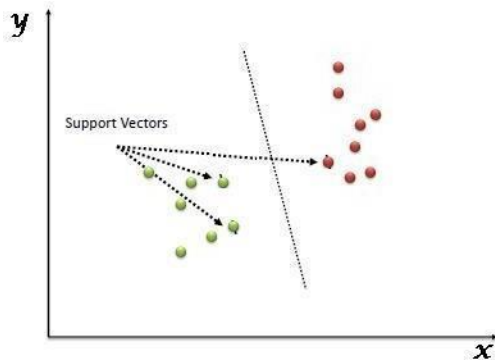
5. PHISHING URL DETECTION

In this module, the Holistic approach is used for the detection of phishing URLs. We implement this approach by comparing it with the Data. Once we find the requested URL is present in the phishing set URLs then the requested URL is blocked by the server. URL is standard URL link and Sub links are the vulnerable keywords which are followed after com or org or in or co. in and

so on

6. FAKE PRODUCTS & AGENCIES DETECTION

In this module, Fake products and fake Agencies are detected through the multiple feedbacks posted by the different users on the server. User Feedbacks are analyzed through our application and fake products and agencies are detected and notified to the new users. Based on the feedbacks fake products and agencies can be omitted from the list, so that users may not be shown with those fake products and agencies.



Data flow diagram

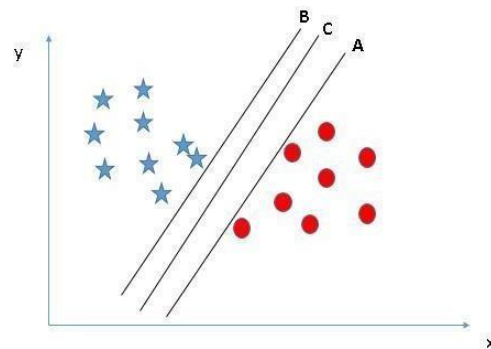
A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various processes the data moves through. DFDs are built using standardized

ALGORITHM USED: SVM

The SVM is a supervised machine learning algorithm that is used for both classification or regression challenges.

In the SVM, we can plot the data item as point in n-dimensional space with the value of each feature being the value of a particular coordinate.

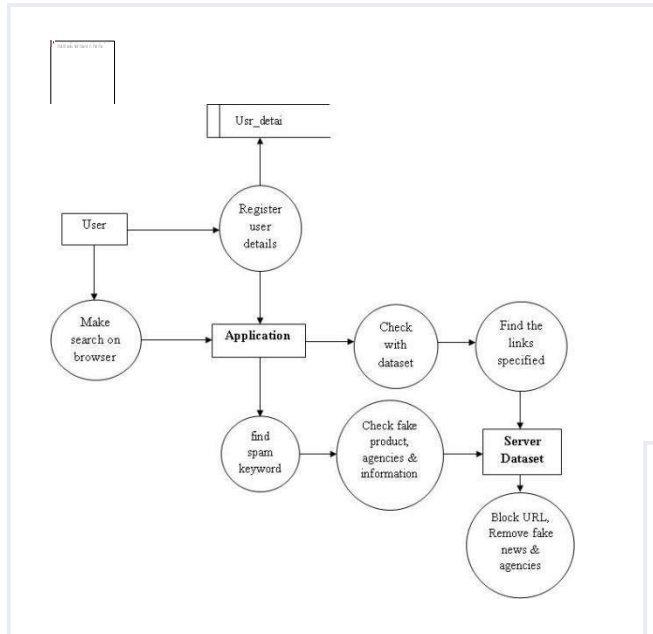
we perform the classification by finding the plane that differentiates the two classes very well.



symbols and notation to describe various entities and their relationships.

Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation. Visualizing each

element makes it easy to identify inefficiencies and produce the best possible system.

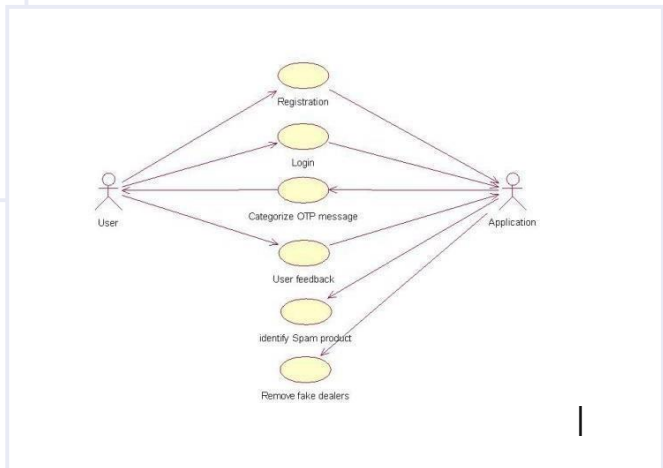


USE case diagram

A use case is a list of steps that define interaction between an actor (a human who interacts with the system or an external system) and the system itself. Use case diagrams depict the specifications of a use case and model the functional units of a system. These diagrams help development teams understand the requirements of their system, including the role of human interaction therein and the differences between various use cases. A use case diagram might display all use cases of

the system, or just one group of use cases with similar functionality.

1. To begin a use case diagram, add an oval shape to the center of the drawing.
2. Type the name of the use case inside the oval.
3. Represent actors with a stick figure near the diagram, then use lines to model relationships between actors and use cases.



Results and Discussions

The final result that we achieve in this Project is that Spam Keywords will be identified and those posts will be discarded.

Phishing URLs are detected and will not allow those URLs to get loaded.

Fake Products and Fake Agencies are identified

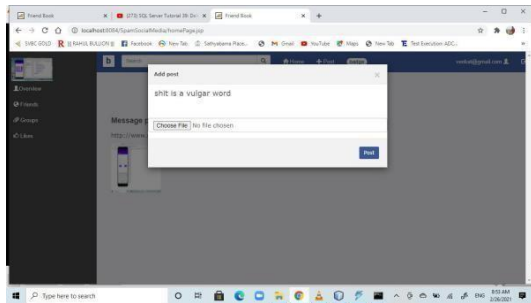
CONCLUSION

We Have deployed social media like the web pages to deduct the malicious URLs and dealers so as to ensure proper and genuine product sales through social media we also identify the fake dealers and fake products through this system. We deploy the android based application to clustering the messages based on keywords.

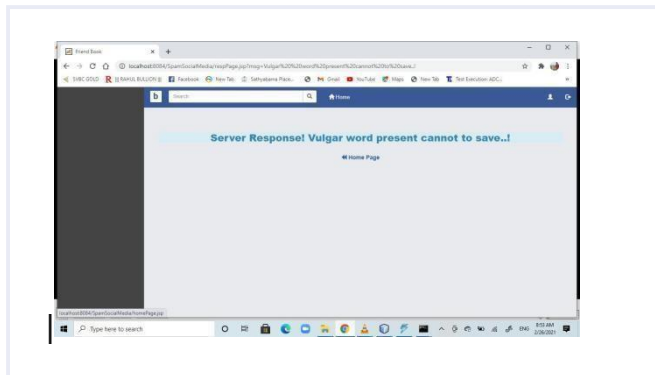
OUTPUTS and SCREENSHOTS

VULGAR words detection and deletion |

User enters a VULGAR word

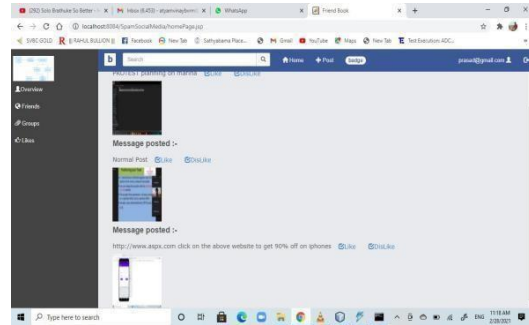


Server detecting and deleting words|

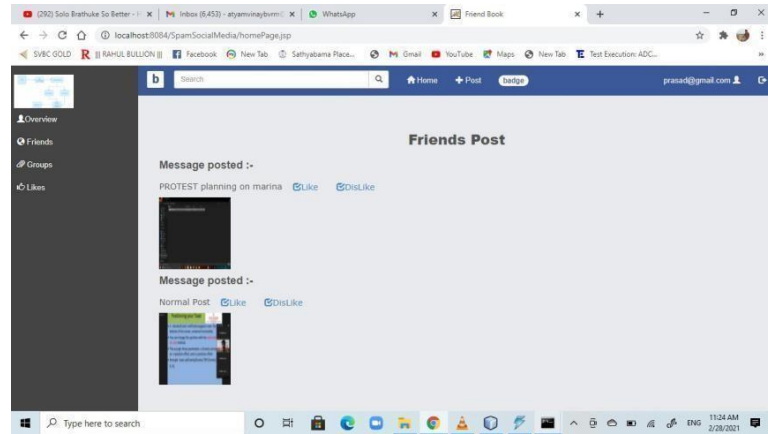


URL detection(based on feedback)

user posted a picture



Link has been deleted due to dislikes



REFERENCES:

- 1. S. Heron. Technologies for spam detection. Network Security, Vol. 2009: Pages 11 – 15, 2009.
- 2. D. Wang, D. I rani, and C. Pu. A study on the evolution of email spam over fifteen years. In the ninth International Conference on Collaborative Computer Networking, Applications and Work

sharing (Collaborate com), pages from 1 to 10. IEEE, 2013.

Skyscraper Lab, “Spam and Phishing Statistics Report Q1-2014”. [Online]. Available: [HTTP://USA.skyscraper.com/counterinsurgency/threats/spam-statistics-report-](http://USA.skyscraper.com/counterinsurgency/threats/spam-statistics-report-q1-2014/)

q1-2014/. Accessed: 2016- 06-2.

Skyscraper Lab, “Spam and Phishing Statistics for 2016”. [Online]. Available: [HTTP://WWW.skyscraper.com/about/preseason/2016kaspersky-lab-reports-significant-increase-in-malicious-spam-emails-in-q1-2016](http://WWW.skyscraper.com/about/preseason/2016kaspersky-lab-reports-significant-increase-in-malicious-spam-emails-in-q1-2016). Accessed: 2016-06- 2.

P. Wood, B. Attorney, K. Chandragupta, S. Wallace, and K. Haley. Semanteme internet security threat report. Vol. 21: pages 27–36, 2016.

G. Cormack. Email spam filtering: A systematic review. Foundations and Trends in Information Retrieval, Vol. 1: Pages 335–455, 2007.

A. Satirist, A. Alabama, M. Hobbs, J. Kawabata, and A. Ahab. “Trends the Crime Toolkit Development” in Networking Security Technologies: Design and Applications. IGI Global. ch2, pages 28-43, 2014.

B. Stone-Gross, T. Hold, G. Strine, and G. Avignon. the underground economy of the spam: The postmasters perspective of coordinating large-scale spam campaigns. USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), Vol. 11: Page 4, 2011.

M. Crawford T. Sharpshooter J. Prussia. “survey of the review spam detection using the machine learning techniques”. Journal Of Big Data, Vol. 2:pages 23, 2015.

M. Shilly-shally, A. Saracen, M.

Mejia, N. Tawbi, and F. Martinelli. Effective and fast clustering of spam emails based on structural similarity. In the International Symposium on Foundations and Practice of Security, pages 195 to 211. Springer, 2015.

PLAGIARISM REPORT

