

# **1.INTRODUCTION**

## **1.1 MOTIVATION:**

The Internet of Things (IoT) is an important component of the new generation of information technology. It is widely used in many fields such as industrial control, cyber-physical systems, and military investigation through the techniques of intelligent perception, identification technology, and pervasive computing and provide communication between objects. It connects humans and objects, objects with objects, provides remote control, and controls intelligent networks in new ways through enabling technologies. An important branch of IoT is control, including human to object control and human control of machines, which is an important foundation for achieving intelligence. Spam is one of the most common forms of attack in mobile networks. Spammers pretend to be normal users and only send spam, and these are the users we aim to detect. A serious problem caused by spam is that links leading to viruses are selected by mistake and then users' personal information is stolen, or production control is interfered with. These malicious nodes communicate with each other and spammers hide in them as shown.

## **1.2 PROBLEM DEFINITION**

An industrial mobile network is crucial for industrial production in the Internet of Things. It guarantees the normal function of machines and the normalization of industrial production. However, this characteristic can be utilized by spammers to attack others and influence industrial production. Users who only share spams, such as links to viruses and advertisements, are called spammers. With the growth of mobile network membership, spammers have organized into groups for the purpose of benefit maximization, which has caused confusion and heavy losses to industrial production. It is difficult to distinguish spammers from normal users owing to the characteristics of multidimensional data. To address this problem, this paper proposes a Spammer Identification scheme based on Gaussian Mixture Model (SIGMM) that utilizes machine learning for industrial mobile

networks. It provides intelligent identification of spammers without relying on flexible and unreliable relationships. SIGMM combines the presentation of data, where each user node is classified into one class in the construction process of the model. We validate SIGMM by comparing it with the reality mining algorithm and hybrid FCM clustering algorithm using a mobile network dataset from a cloud server. Simulation results show that SIGMM outperforms these previous schemes in terms of recall, precision, and time complexity

### **The SIGMM model**

The SIGMM model is proposed and developed based on Gaussian Mixture Model, which focuses on spammer identification. The paper contains the following three main contributions. Based on the Gaussian Mixture Model, we propose a recognition process named the SIGMM model for classification without relying on users' unreliable relationships. SIGMM can label data automatically, which increases the precision of the model by expanding the training set. We use an industrial mobile network dataset from a cloud server to perform simulations. The results show that SIGMM performs better than two other models in terms of identifying spammers and reducing time complexity. For existing algorithms, there are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

## **1.3 OBJECTIVE OF PROJECT**

The following are the objectives of the project:

- ☐ To overcome the disadvantages of existing system. (Time complexity, and analysis through relationships).
- ☐ To identify spammers based on their behavior.

## **2. SYSTEM ANALYSIS**

### **2.1 INTRODUCTION**

#### **2.1.1 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

#### **2.1.2 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.

#### **2.1.3 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.

## **2.1.4 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.

## **2.2 REQUIREMENT SPECIFICATIONS**

### **2.2.1 HARDWARE REQUIREMENTS**

- ☐ System : Pentium IV 2.4 GHz.
- ☐ Hard Disk : 40 GB.
- ☐ Floppy Drive : 1.44 Mb.
- ☐ Monitor : 15 VGA Colour.
- ☐ Mouse : Logitech.
- ☐ Ram : 512 Mb.

### **2.2.2 SOFTWARE REQUIREMENTS**

- ☐ Operating system : Windows XP/7.
- ☐ Coding Language : JAVA/J2EE.
- ☐ IDE : Netbeans 7.4.
- ☐ Database : MYSQL.

## 2.3 EXISTING SYSTEM

Existing methods mainly depend on the relationships among users. However, owing to the development of intelligent recommendation mechanisms, user associations are not based on their true preferences or intentions. In the case where the users are not very clearly defined, a user might follow or be followed by an automatic link. Therefore many users represent fuzzy relationships.

## DISADVANTAGES

- The main goal of supervised learning is to learn a model from labeled training data that allows us to make predictions about unseen or future data
- In the RMA(reality mining algorithm) based on K-means in, the algorithm proposes a silhouette function which accepts the number of clusters as a parameter to judge the accuracy of clustering. Then it uses a matrix of means to record the mean silhouette values for each value of k and finally determines the best value of k. But the clustering result depends on the k centroids. Therefore it must consume extra time to determine the value of k.
- Recursive least squares (RLS) algorithm based on the reinforcement algorithm is proposed. It applies Q-Learning by choosing a policy which is the best selection for a specific user. But it starts from a random user and does the exploration within the network by friendship relationships, which restricts the scope of the exploration, and leads to decreased detection efficiency.

## 2.4 PROPOSED SYSTEM

Proposals from industry and academia discuss solutions for shielding against spam. Classification based on machine learning is a learning process for mapping data samples into two classes. However it has limitations. One is data imbalance, unlabeled data are present in a much larger amount than labeled data, which hinders direct model construction. Another limitation is multidimensional data, too many features can lead to over fitting. Hence intelligent feature selection is necessary. In this paper, we first investigate the characteristics of spammers and normal users in an industrial mobile

network. Then, the SIGMM model is proposed and developed based on Gaussian Mixture Model, which focuses on spammer identification.

## **ADAVANTAGES**

- Based on the Gaussian Mixture Model, we propose a recognition process named the SIGMM model for classification without relying on users' unreliable relationships.
- SIGMM can label data automatically, which increases the precision of the model by expanding the training set. It labels large amounts of unlabeled data based on a few labeled data and solves the problem of the imbalance between labeled data and unlabeled data.
- We use an industrial mobile network dataset from a cloud server to perform simulations. The results show that SIGMM performs better than two other models in terms of identifying spammers and reducing time complexity.

## **2.5 MODULES**

After careful analysis the system has been identified to have the following modules:

- User module
- Admin module

**USER MODULE :** In user module it consists of basic functionalities such as registering with their own credentials , searching for friends, sending and accepting the friend requests, uploading profile photo of their own, sharing both labeled and unlabeled data.

User is not given the access of viewing all contents and actions of the users.

**ADMIN MODULE :** In Admin module it consists of access to all the user specific operations, an admin views all the shared data between the users.

Admin's main activity is to find out whether the user is authorized user or spammer

This identification of normal user and spammer is done by using Gaussian mixture model

Under this model the spammer is identified and his activities are blocked by the admin

If the user is normal user and is characterized as the spammer then he can forward a request to admin providing necessary proofs that he/she is not a spammer and then on verifying the admin can unblock them.

## **2.5.1 MODULES FUNCTIONALITY DESCRIPTION:**

### **Supervised and Unsupervised Learning**

The main goal of supervised learning is to learn a model from labeled training data that allows us to make predictions about unseen or future data. Supervised refers to a set of samples where the desired output labels are already known. In the Spammer Selection algorithm based on Logistic Regression, a spammer classifier is built for an online network with some features as inputs, and the algorithm output is 1 if a spammer is suspected. The model is trained on a large training set, however, collection of labeled data is rather difficult because of the recent emphasis on the secrecy of user data. Using unsupervised learning techniques, we are able to explore the structure of our data to extract meaningful information without the guidance of a known outcome variable or reward function. A clustering algorithm is the main algorithm for unsupervised learning. Clustering is a technique that allows us to find groups of similar members. In the RMA based on K-means in, the algorithm proposes a silhouette function which accepts the number of clusters as a parameter to judge the accuracy of clustering. Then it uses a matrix of means to record the mean silhouette values for each value of k and finally determines the best value of k. But the clustering result depends on the k centroids. In, a prediction model based on Big Data analysis using a hybrid FCM clustering algorithm (HFCM) is proposed.

### **Feature scaling**

The data we obtained have the following two constraints. First, the labeled data are far fewer than the unlabeled data which severely decreases the precision of training. Second, there is large data noise that may cause incorrect factors in the parameters of the model. Data points that do not belong to any class are defined as data noise. The values of some data may greatly differ from the mean of samples. SIGMM reduces data noise by calculating the similarity among users to increase the precision of training.

In order to remove data noise from large datasets, the similarity is calculated first according to the vectors that describe user behaviors.

## Gaussian Mixture Model

The data are approximately subject to the Gaussian distribution. The mean and variance must be estimated for initializing the model. According to the probability density  $p(x|q)$ , we independently extract some samples to constitute the training sample set  $X$ . Parameter  $q$  represents the mean and variance of the dataset, and is estimated through the sample set  $X$ . Consider  $X = \{x_1; x_2; \dots; x_n\}$  as a set of extracted samples,  $x_i$  represents the  $i$ th user data, and  $n$  represents the number of samples. Because they are independent, the probability that  $x_i$  and  $x_j$  are extracted simultaneously is  $p(x_i|q) * p(x_j|q)$ . Similarly, the probability that  $n$  samples are extracted simultaneously is the product of their respective probabilities.

## Classification of SIGMM Model

A Gaussian mixture model is a probabilistic model for statistical learning. Through the estimation of the probability density distribution of samples, each Gaussian model represents a class. By matching samples with several Gaussian models to obtain probabilities, the class with the largest probability is chosen as the classification result. The data of the two classifications are clearly separated. The radius of the green one representing normal users is larger than that of the red one, because the number of normal users is large. Their behavior data are not similar to each other and deviate from the center. The radius of the red one is smaller than that of the green one because spammer behaviors for attacking others are similar. The radius and location of the two ellipsoids vary only slightly. Therefore, the model is stable.



# 2.6 SYSTEM ARCHITECTURE

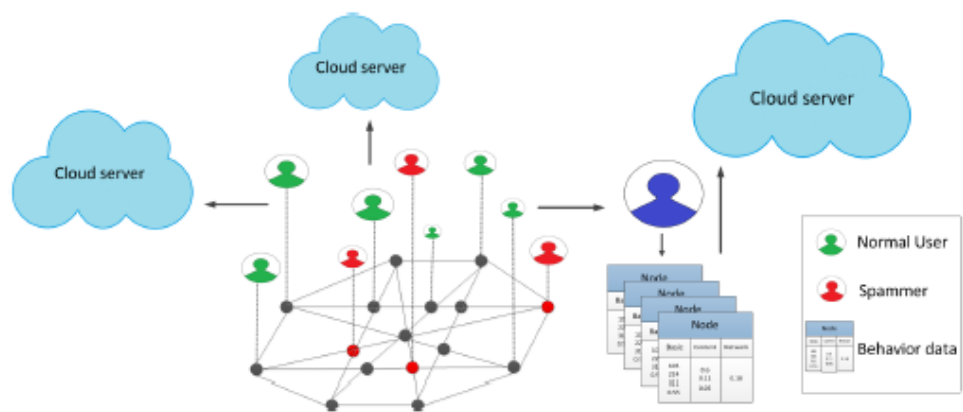


Fig1: System Architecture

## **3. SYSTEM DESIGN**

### **3.1 INTRODUCTION**

#### **REQUIREMENTS SPECIFICATION DOCUMENT**

According to Roger Pressman in Software Engineering: A Practitioner's Approach (McGraw-Hill Publications) [SEPA–1997], the requirement specification document is produced at the end of Analysis of the system. This document is a very comprehensive document & contains all the User requirements & Analysis diagrams. The Requirements are broadly divided into two groups:

1. Functional requirements
2. Non-functional requirements

#### **FUNCTIONAL REQUIREMENTS**

The main purpose of functional requirements within the requirement specification document is to define all the activities or operations that take place in the system. These are derived through interactions with the users of the system. According to this system each project should be maintain unique project\_code in the database.

#### **NON-FUNCTIONAL REQUIREMENTS**

The non-functional requirements consist of:

1. Analysis, Design & Data requirements.
2. Constraints.
3. Guidelines.
4. Validation Criteria.

## 3.2 ARCHITECTURE & TECHNOLOGIES USED

After analyzing the Requirements, the Architecture chosen for the System to be implemented is decided as the Multi-tier Architecture. This architecture is chosen because it separates the Development of the System into independent modules or tiers. Generally complex projects use this, as the work in each tier can be taken up by specialists in each field.

### 3.2.1 JAVA TECHNOLOGY

Java technology is both a programming language and a platform.

#### **The Java Programming Language**

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

- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

#### **The 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 Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a

combination of the operating system and 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* (Java VM)
- The *Java Application Programming Interface* (Java API).

### 3.2.2 JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

#### JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. ***SQL Level API***

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

2. ***SQL Conformance***

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. ***JDBC must be implemental on top of common database interfaces***

The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. ***Provide a Java interface that is consistent with the rest of the Java system***

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. ***Keep it simple***

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6. ***Use strong, static typing wherever possible***

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

### ***7. Keep the common cases simple***

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

### **3.2.3 JSP**

Java Servlet and JSPs are server-side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.

#### **Web Container**

Tomcat is a web container, when a request is made from Client to web server, it passes the request to web container and it's web container job to find the correct resource to handle the request (servlet or JSP) and then use the response from the resource to generate the response and provide it to web server. Then web server sends the response back to the client.

When web container gets the request and if it's for servlet then container creates two Objects `HttpServletRequest` and `HttpServletResponse`. Then it finds the correct servlet based on the URL and creates a thread for the request. Then it invokes the servlet `service()` method and based on the HTTP method `service()` method invokes `doGet()` or `doPost()` methods. Servlet methods generate the dynamic page and write it to response. Once servlet thread is complete, container converts the response to HTTP response and send it back to client.

Some of the important work done by web container are:

- ☐ Communication Support
- ☐ Lifecycle and Resource Management
- ☐ Multithreading Support
- ☐ JSP Support
- ☐ Miscellaneous Task

### **3.3 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### **GOALS:**

The Primary goals in the design of the UML are as follows:

- ☐ Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- ☐ Provide extendibility and specialization mechanisms to extend the core concepts.
- ☐ Be independent of particular programming languages and development process.
- ☐ Provide a formal basis for understanding the modeling language.
- ☐ Encourage the growth of OO tools market.
- ☐ Support higher level development concepts such as collaborations, frameworks, patterns and components.
- ☐ Integrate best practices.

### 3.3.1 USE CASE DIAGRAM:

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

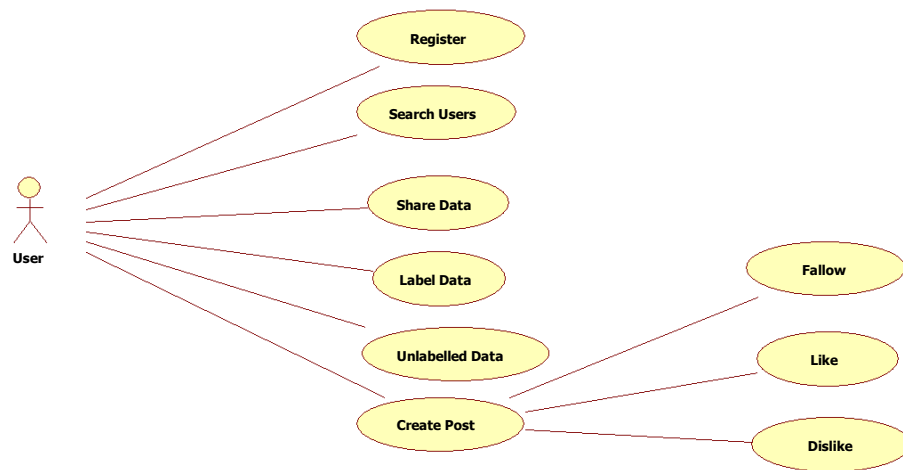


Fig 2: User Use case

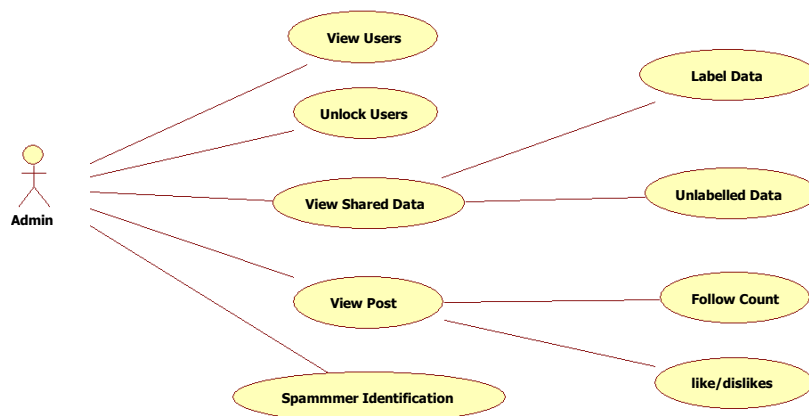


Fig 3: Admin use case



### 3.3.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

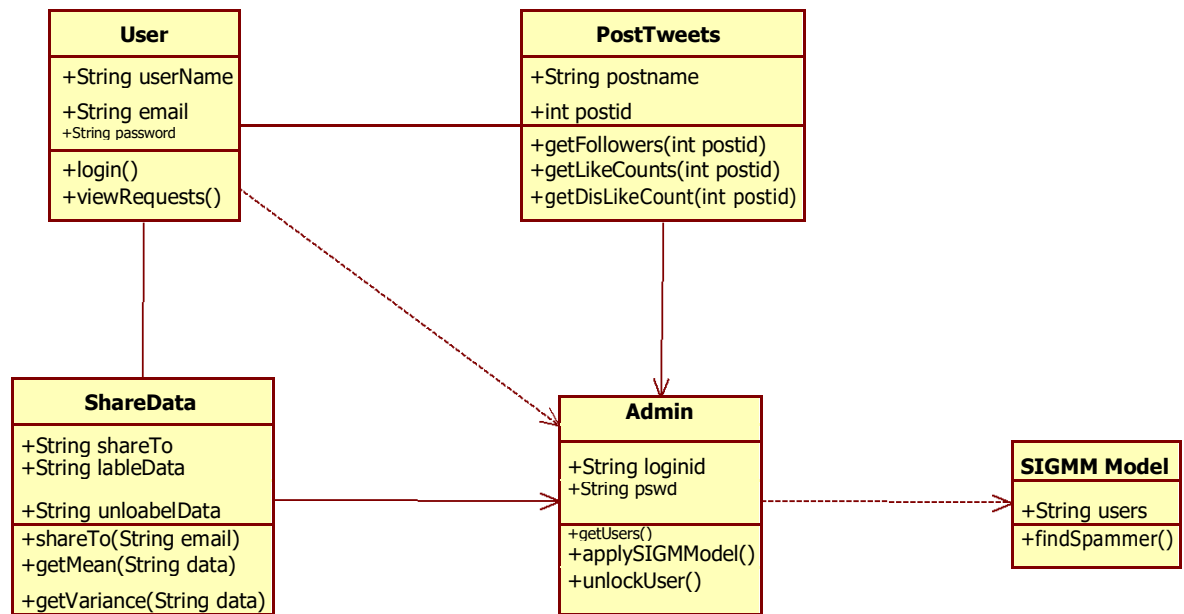


Fig 4: Class diagram

### 3.3.3 SEQUENCE DIAGRAM

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

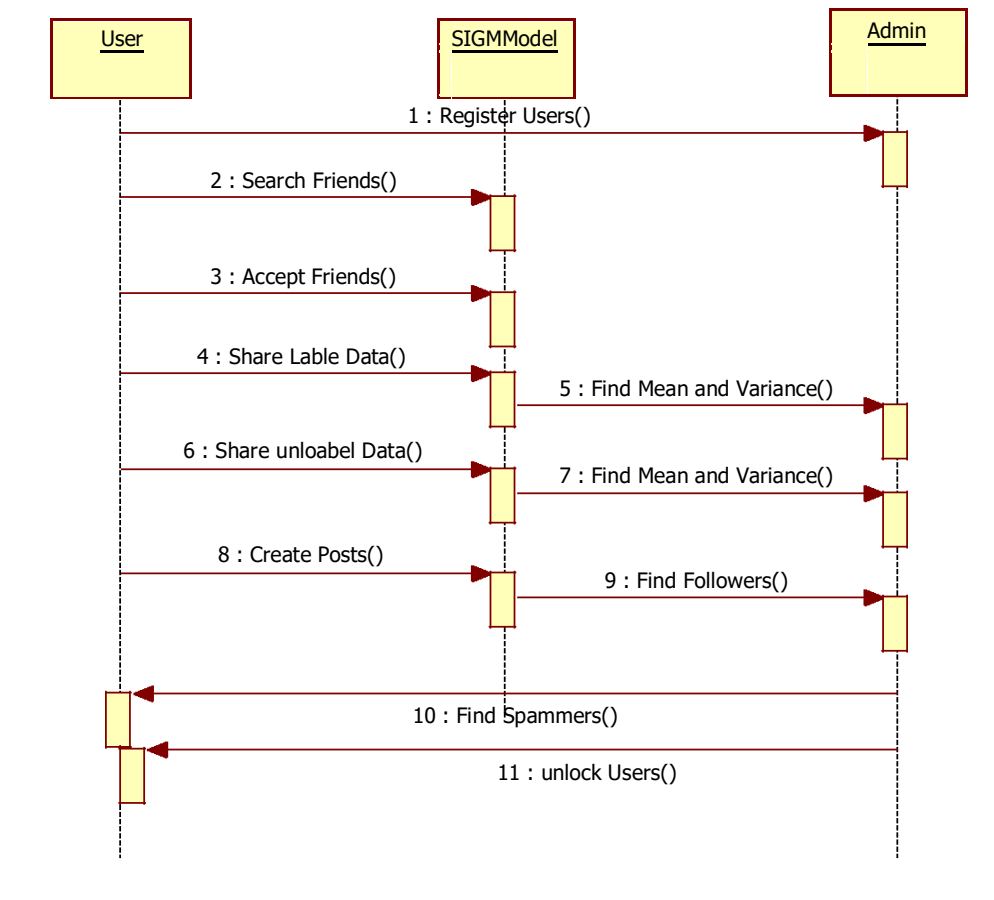


Fig 5: Sequence diagram to unlock user

### 3.3.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

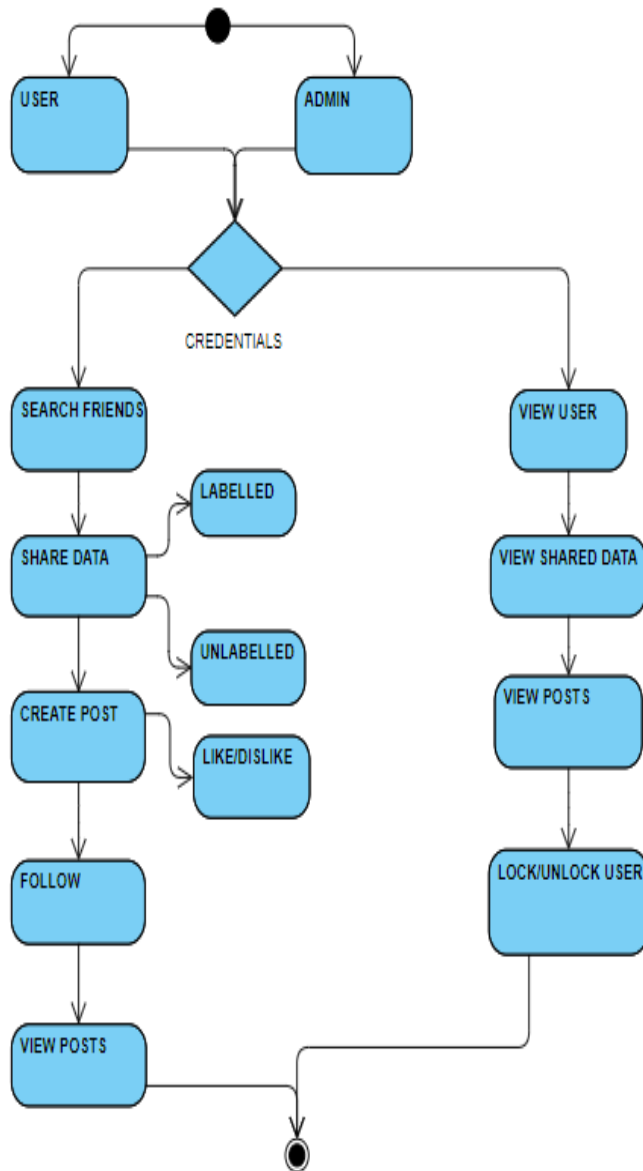


Fig6: Activity diagram for workflow of User and Admin

## 3.4 DATABASE

Data dictionary consists of descriptions of all the data used in the system. It consists of logical characteristics of the data stored including name, description, aliases, contents and organization. Data dictionary serves as the basis for identifying database requirements during system design. It is used to manage the detail in large systems, to communicate a common meaning for all the system elements, to document the features of the system, to locate errors and omissions in the system. Data dictionary consists of two types of descriptions for the data flowing through the system – data elements and data structures. Data elements are grouped together to make up the data structure. The description of data elements consists of data names, data description, aliases, and length and data values. The data structures describe sequence relationship, selection relationship, iteration relationship and operational relationship.

### 3.4.1 MySQL:

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQLWebsite (<http://www.mysql.com/>) provides the latest information about MySQL software. MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- ☐ MySQL databases are relational.
- ☐ MySQL software is Open Source.
- ☐ The MySQL Database Server is very fast, reliable, scalable, and easy to use.
- ☐ A large amount of contributed MySQL software is available.

### 3.4.2 DATABASE TABLES

Mysql> desc followings;

Field	Type	Null	Key	Default
id	int(11)	No	Pri	NULL
postname	varchar(500)	No		
postid	int(11)	Yes	Mul	NULL
postcreatedby	varchar(50)	No		
creatermail	varchar(50)	No		
followername	varchar(50)	No		
followemail	varchar(50)	No		
followingtimestamp	varchar(50)	No		
cdate	date	Yes		NULL

Table 1 : Followings table

Mysql> desc friendstable;

Field	Type	Null	Key	Default
fromemail	varchar(60)	No	Pri	
fromuser	varchar(60)	No		
toemail	varchar(60)	No	Pri	
touser	varchar(60)	No		
status	varchar(60)	No		
rdate	date	No		

Table 2: Friend stable

Mysql> desc labledataset;

Field	Type	Null	Key	Default
id	int(11)	No	Pri	NULL
fromuseremail	varchar(60)	No		
fromusername	varchar(60)	No		
touseremail	varchar(60)	No		
tousername	varchar(60)	No		
message	varchar(60)	No		
sharedate	date	No		
meanval	double	No		
variance	double	No		
stdDev	double	No		
median	double	No		
likelihoodfunction	double	Yes		NULL

Table 3: Labeledataset table

Mysql> desc likedislike;

Field	Type	Null	Key	Default
id	int(11)	No	Pri	NULL
postame	Varchar(500)	No		
postid	Int(11)	Yes	Nul	NULL
postcreatedby	varchar(50)	No		
creatermail	varchar(50)	No		
followername	varchar(50)	No		
likestatus	varchar(50)	No		
followemail	varchar(50)	No		
followingtimestamp	varchar(50)	No		
cdate	date	Yes		NULL

Table 4: Likedislike table

Mysql> desc unlabelledataset;

Field	Type	Null	Key	Default
id	int(11)	No	Pri	NULL
fromuseremail	varchar(80)	No		
fromusername	varchar(80)	No		
touseremail	varchar(80)	No		
tousername	varchar(80)	No		
message	longblob	No		
sharedate	date	No		
meanval	double	No		
variance	double	No		
stdDev	double	No		
median	double	No		
likelihoodfunction	double	No		
filename	varchar(100)	No		

Table 5 : Unlabeldataset table

Mysql> desc userpics;

Field	Type	Null	Key	Default
profilepic	longblob	Yes		NULL
email	varchar(50)	No	Pri	

Table 6 : Userpics table

Mysql> desc userposts;

Field	Type	Null	Key	Default
id	int(11)	No	Pri	NULL
postname	varchar(500)	No	Uni	
creatername	varchar(60)	No		
useremail	varchar(60)	No		
postdate	date	No		
ctimestamp	varchar(60)	No		

Table 7: Userposts table

Mysql> desc userregisters;

Field	Type	Null	Key	Default
username	varchar(50)	No		
firstname	varchar(50)	No		
lastname	varchar(50)	No		
email	varchar(50)	No	Pri	
pswd	varchar(50)	No		
mobile	varchar(50)	No		
dob	varchar(50)	No		
gender	varchar(50)	No		
address	varchar(500)	No		
spamstatus	varchar(50)	No		

Table 8: Userregisters table



### 3.5 DATABASE ARCHITECTURE

A Database Management system is not always directly available for users and applications to access and store data in it. A Database Management system can be centralized(all the data stored at one location), decentralized(multiple copies of database at different locations) or hierarchical, depending upon its architecture.

Database Architecture is logically of two types:

- 2-tier DBMS architecture
- 3-tier DBMS architecture

#### 2-tier DBMS Architecture

2-tier DBMS architecture includes an Application layer between the user and the DBMS, which is responsible to communicate the user's request to the database management system and then send the response from the DBMS to the user.

An application interface known as ODBC(Open Database Connectivity) provides an API that allow client side program to call the DBMS. Most DBMS vendors provide ODBC drivers for their DBMS.

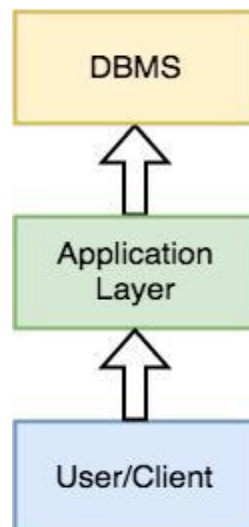


Fig7: 2-tier DBMS Architecture

Such an architecture provides the DBMS extra security as it is not exposed to the End User directly. Also, security can be improved by adding security and authentication checks in the Application layer too.

### 3-tier DBMS Architecture

3-tier DBMS architecture is the most commonly used architecture for web applications. It is an extension of the 2-tier architecture.

In 3-tier architecture, an additional Presentation or GUI Layer is added, which provides a graphical user interface for the End user to interact with the DBMS. For the end user, the GUI layer is the Database System, and the end user has no idea about the application layer and the DBMS system.

If you have used MySQL, then you must have seen PHPMysqlAdmin, it is the best example of a 3-tier DBMS architecture.

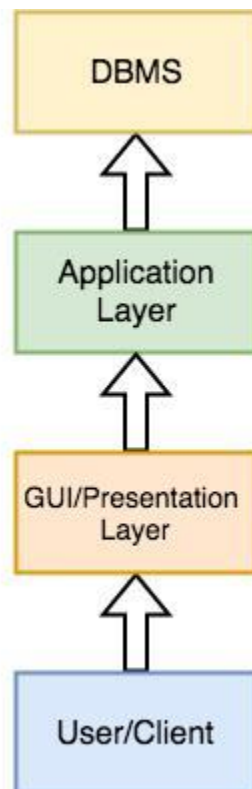


Fig 8: 3-tier DBMS Architecture

## **4. IMPLEMENTATION AND RESULTS**

### **4.1 INTRODUCTION**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

In project implementation or project execution, we put it all together. Project planning is complete, as detailed as possible, yet providing enough flexibility for necessary changes. In a customer-contractor relationship, the contract is signed, based on the right decisions about the contract structures, and including clauses for change and claim management.

Now we apply all the tools we prepared in order to keep ourselves in control of the project. As project managers and sub-project managers we have to make sure that we

- ☐ together with all our team members,
- ☐ take action, in-line with the plan and / or contract
- ☐ record and document all the work, work results, special events, decisions about changes, implementation of changes, etc.
- ☐ analyze, communicate, report, and document status and results of action, in-line with the plan and / or contract
- ☐ take decision if and what kind of change we need, in case any result (or action) is not as required

In project implementation, we manage implementation of all our project plans, following the triple constraint:

- Project Scope Management
- Project Time Management
- Project Cost Management

In case we are applying the classical planning approach or the critical chain method, our focus will be on the comparison of actual results with required results.

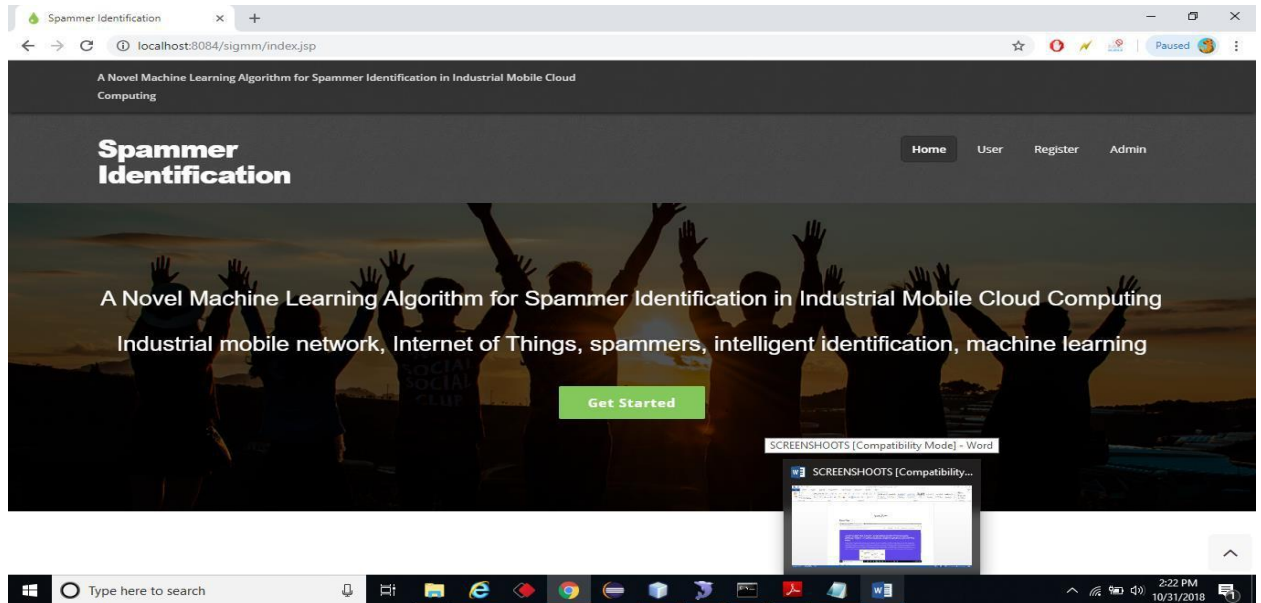


Fig 9: Project Implementation process Diagram

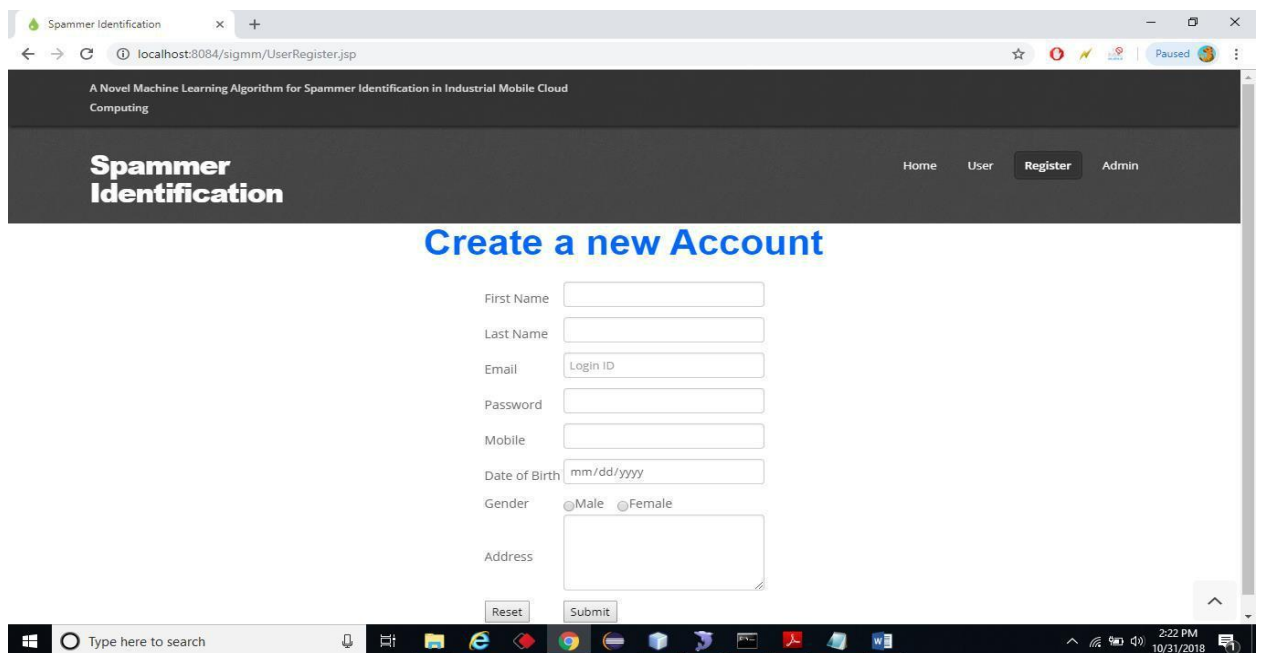
The most powerful platform for this comparison in order to analyze, communicate, and decide work progress, problems, and necessary changes are project meetings in which we apply the planned project controlling tools.

- ☐ kick-off meetings
- ☐ regular status meetings
- ☐ special status meetings
- ☐ risk analysis workshops (as part of our risk management strategy)
- ☐ problem solving workshops
- ☐ project management review meetings

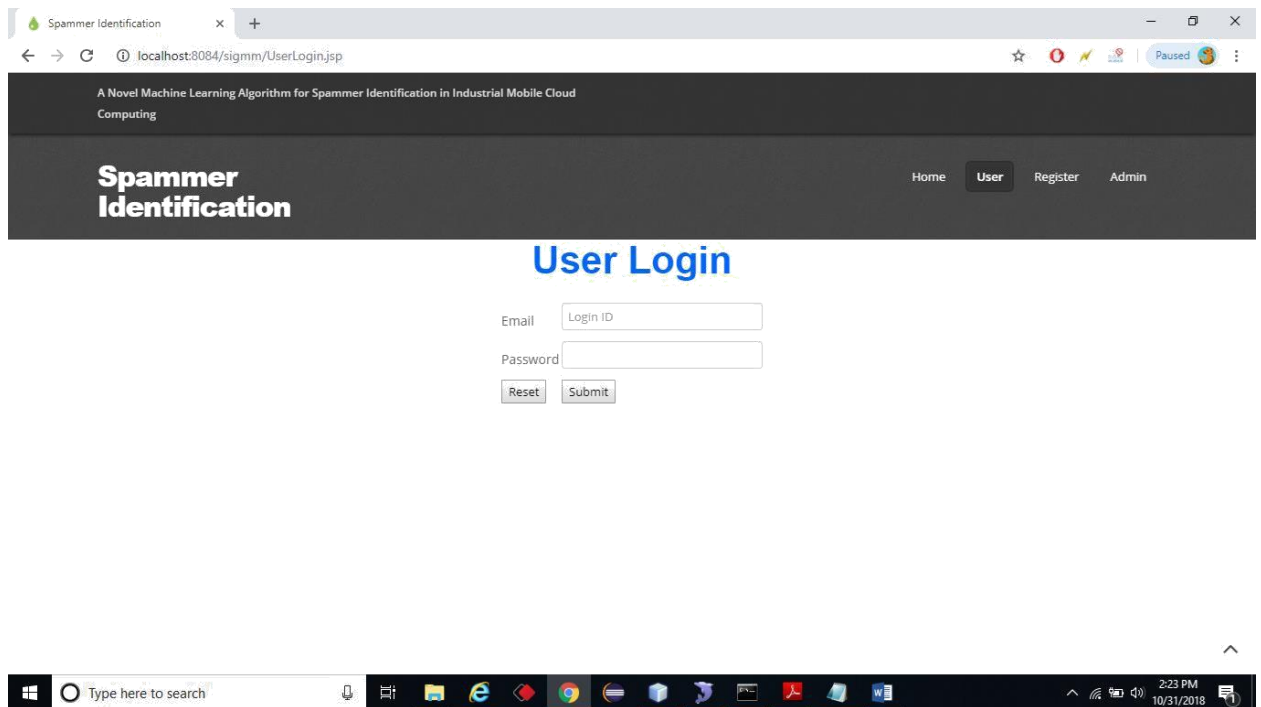
## 4.2 SCREENS



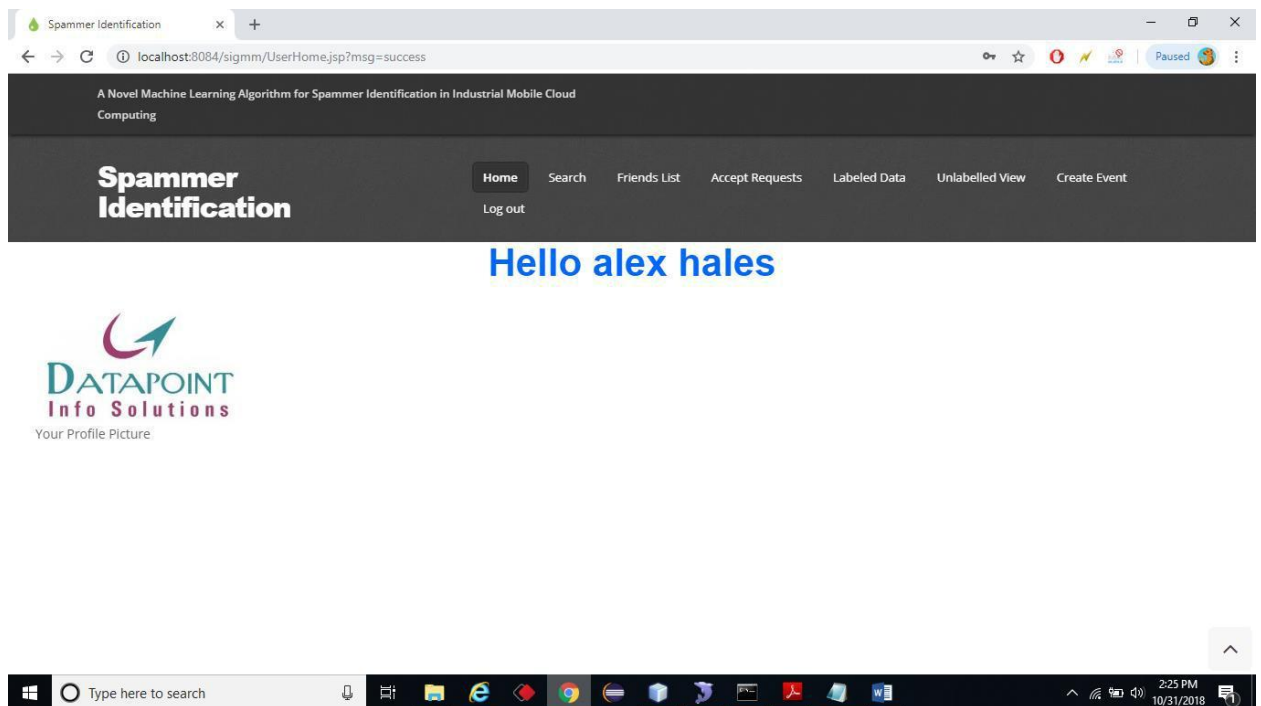
Screen 1: Home Page



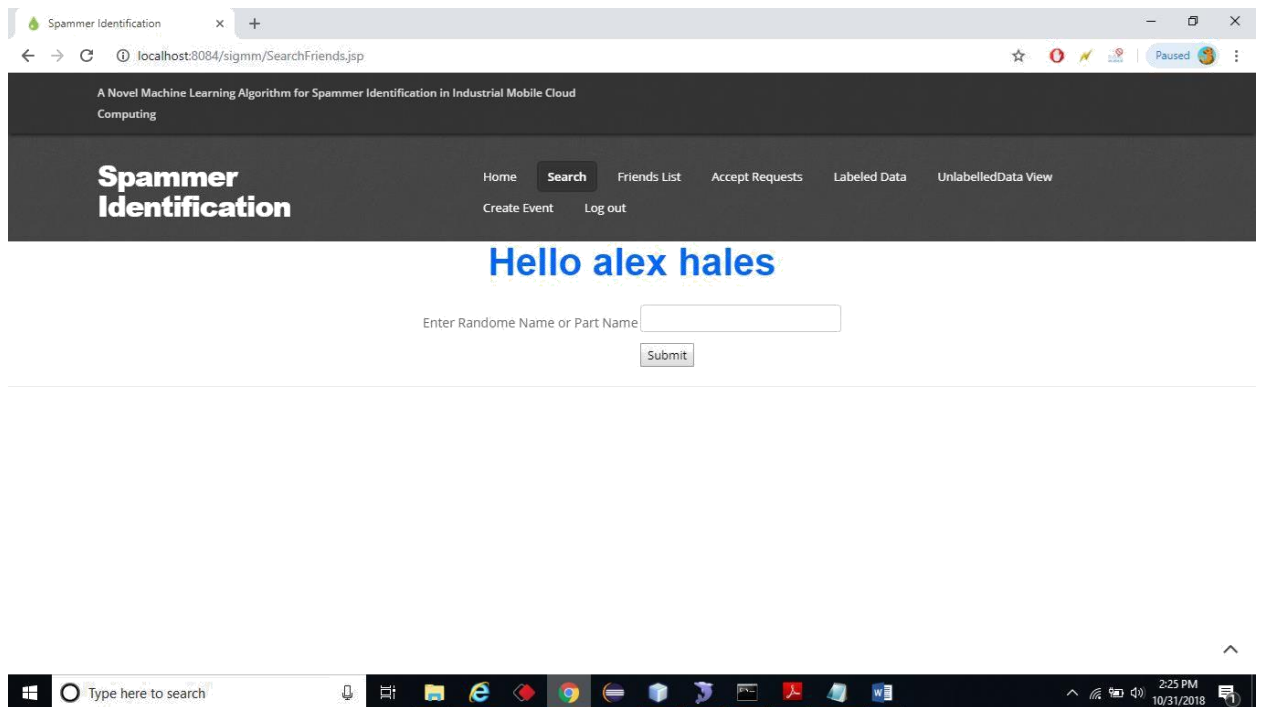
Screen 2 : User Registration



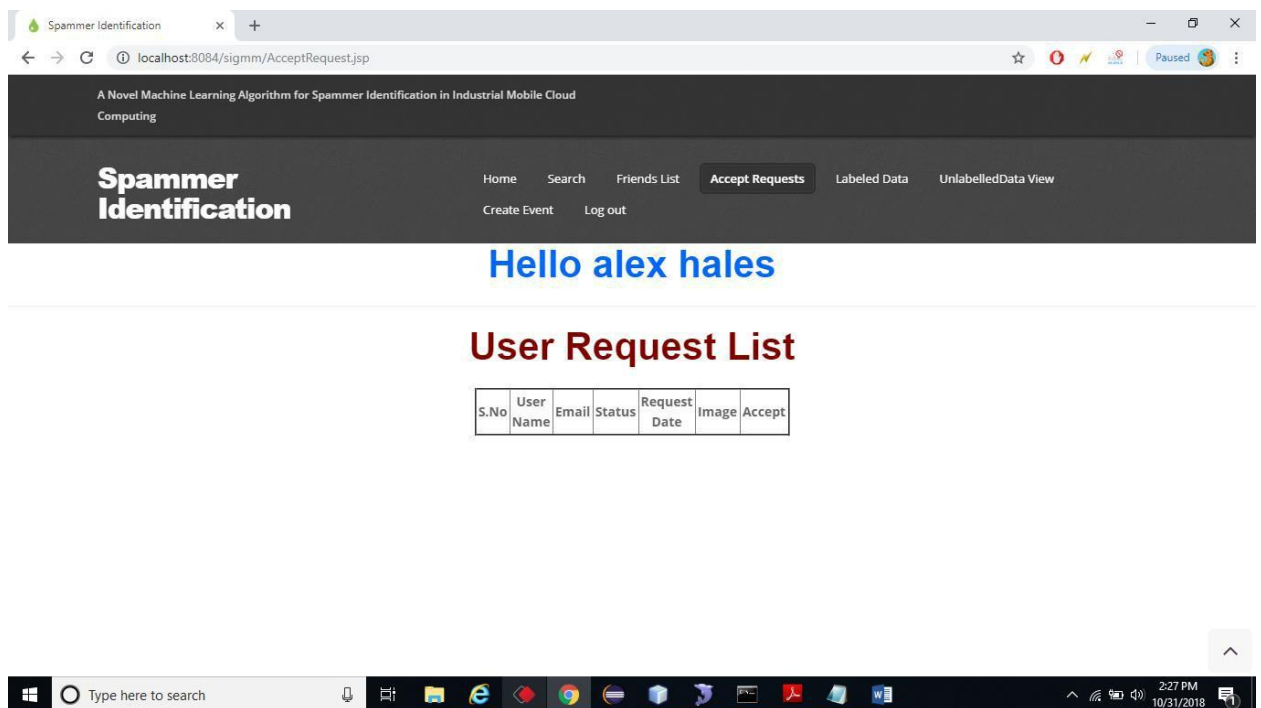
Screen 3 : User Login



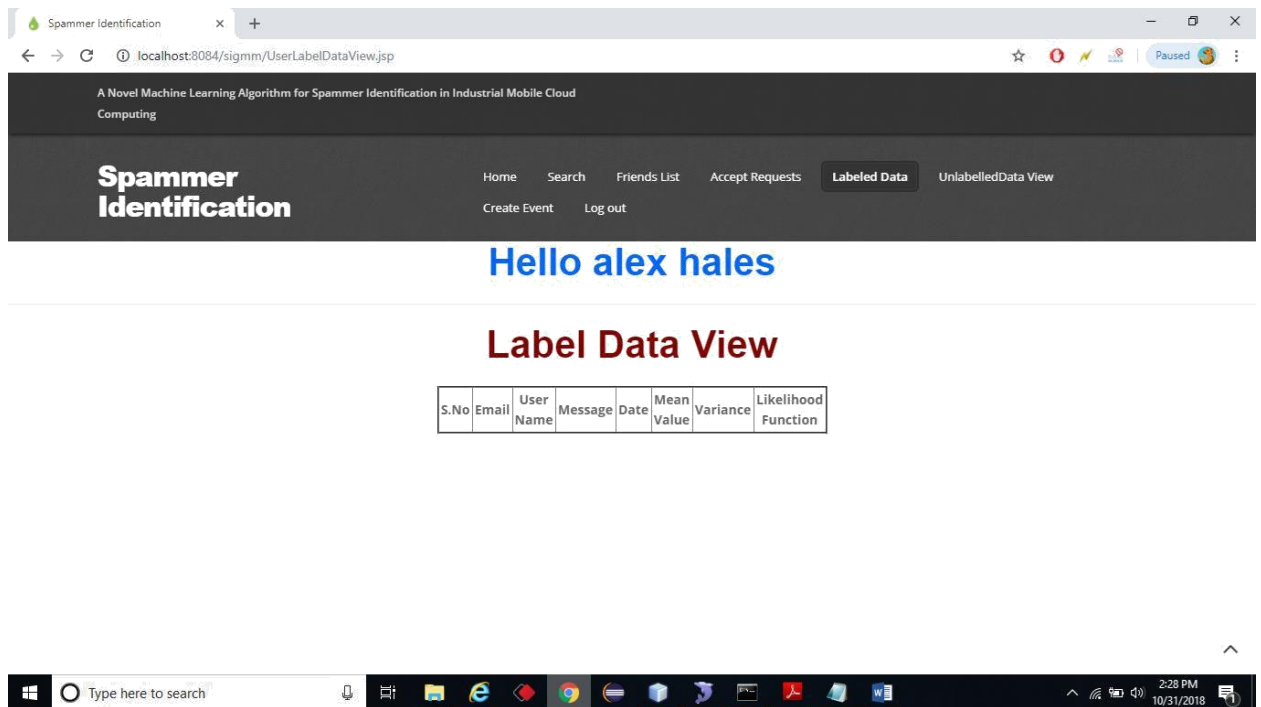
Screen 4 : User Homepage



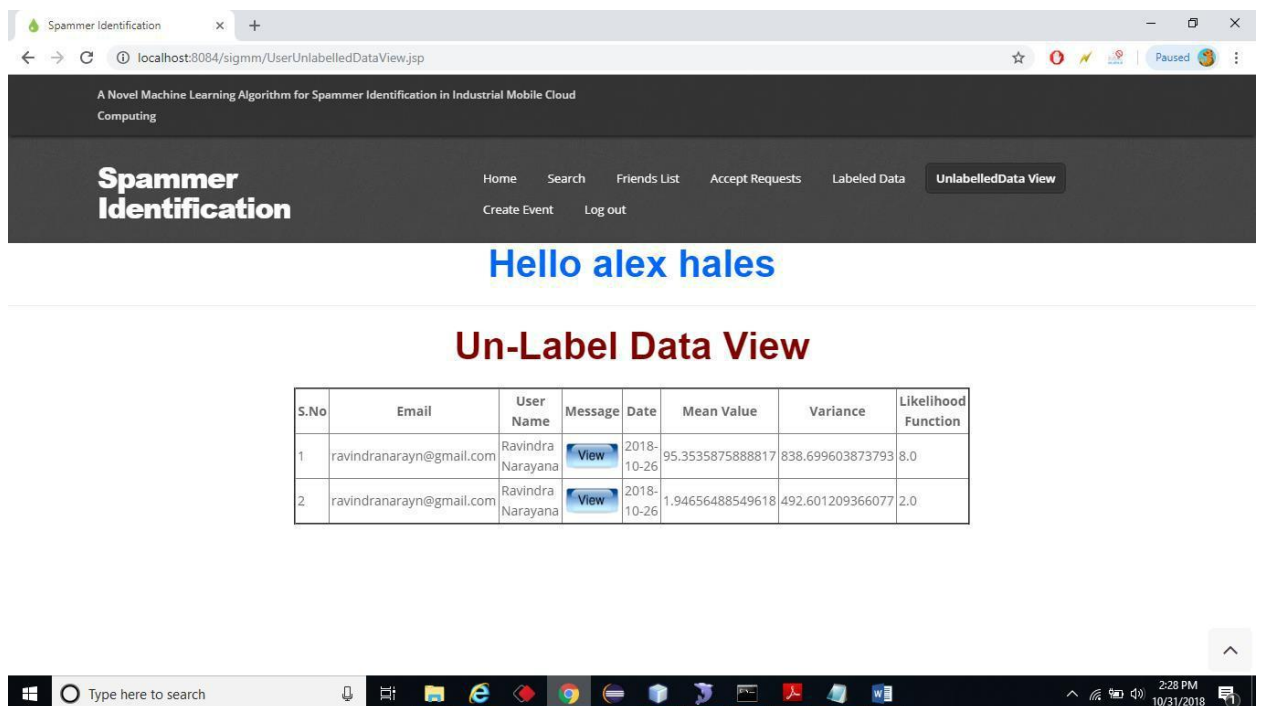
Screen 5: friends Search



Screen 6 : User Request List



Screen 7 : Label data view



Screen 8: Unlabel data view



Spammer Identification

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

**Spammer Identification**

Home Search Friends List Accept Requests Labeled Data Unlabelled View **Create Event**

Log out

## Hello alex hales

### Creating Event

Write Something here\*

Date

### Posted Comments

S.No	Post Name	Posted by	Times Stamp	Fallow	Like	Dislike
1	The RSA Algorithm is first public key encryption to the world	alex hales	Mon, 29 Oct 2018 15:47:35 IST	Fallow 7	Like 5	Dislike 3
2	The Spammer in industrial mobile is a latest trend in the market..	Akhila Konakanchi	Mon, 29 Oct 2018 16:55:26 IST	Fallow 5	Like 5	Dislike 2

Screen 9 : Create event

Spammer Identification

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

**Spammer Identification**

Home **Search** Friends List Accept Requests Labeled Data UnlabelledData View

Create Event Log out

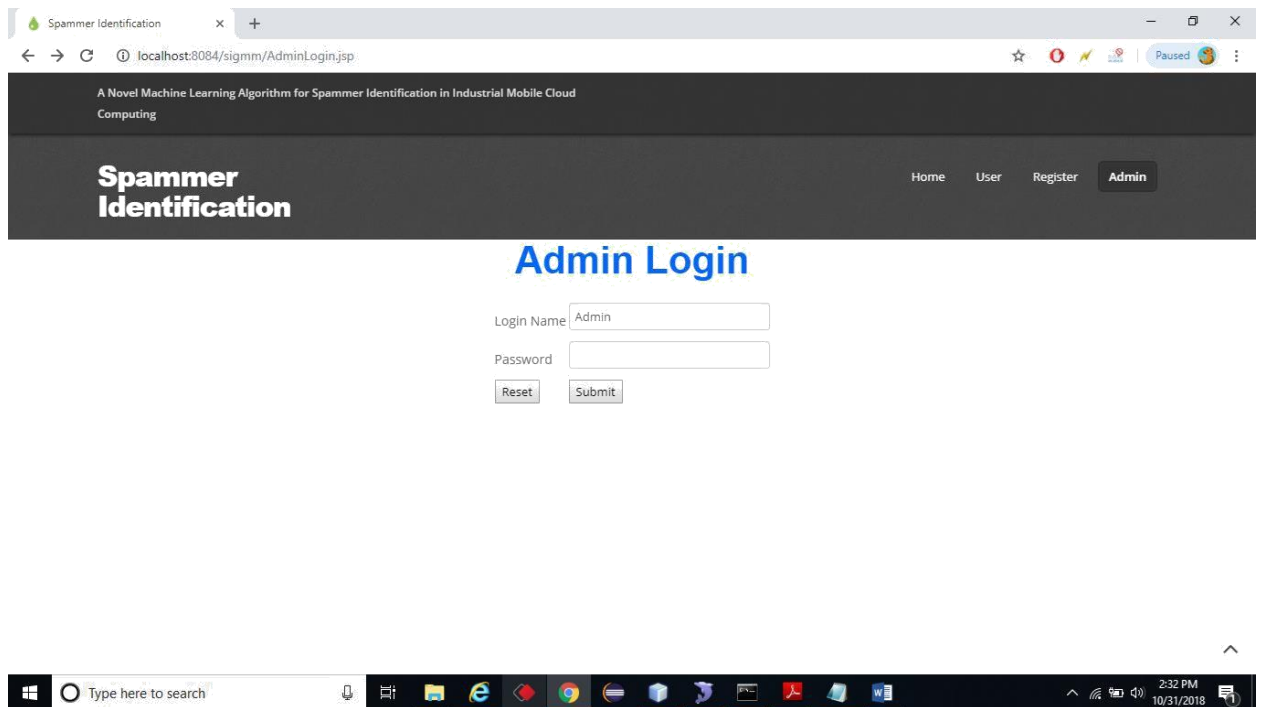
## Hello alex hales

Enter Random Name or Part Name

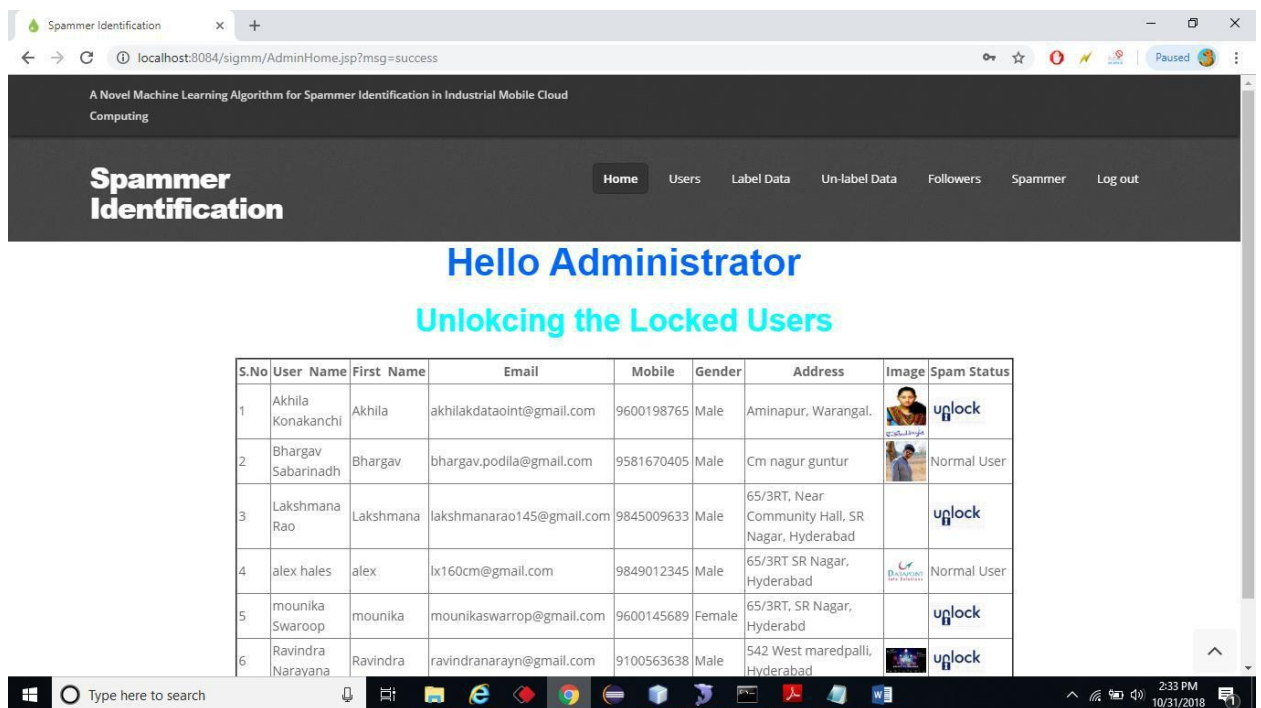
### Search Results

S.No	User Name	First Name	Email	Mobile	Date of Birth	Gender	Address	Image	Send Request
1	Akhila Konakanchi	Akhila	akhilakdataoint@gmail.com	9600198765	1995-02-07	Male	Aminapur, Warangal.		<a href="#">Send Request</a>
2	Bhargav Sabarinadh	Bhargav	bhargav.podila@gmail.com	9581670405	1996-08-02	Male	Cm nagur guntur		<a href="#">Send Request</a>
5	Ravindra	Ravindra	ravindranarayn@gmail.com	9100563638	1995-	Male	542 West maredpalli		<a href="#">Send</a>

Screen 10: Search results



Screen 11 : Admin Login



Screen 12: Locked users list







Spammer Identification

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

Home **Users** Label Data Un-label Data Followers Spammer Log out

## Hello Administrator

### Viewing the Register Users

S.No	User Name	First Name	Email	Mobile	Date of Birth	Gender	Address	Image
1	Akhila Konakanchi	Akhila	akhilakdataoint@gmail.com	9600198765	1995-02-07	Male	Aminapur, Warangal.	
2	Bhargav Sabarinadh	Bhargav	bhargav.podila@gmail.com	9581670405	1996-08-02	Male	Cm nagur guntur	
3	Lakshmana Rao	Lakshmana	lakshmanarao145@gmail.com	9845009633	1995-02-07	Male	65/3RT, Near Community Hall, SR Nagar, Hyderabad	
4	alex hales	alex	lx160cm@gmail.com	9849012345	2002-01-07	Male	65/3RT SR Nagar, Hyderabad	
5	mounika Swaroop	mounika	mounikaswaroop@gmail.com	9600145689	1994-01-11	Female	65/3RT, SR Nagar, Hyderabad	
6	Ravindra Narayana	Ravindra	ravindranarayn@gmail.com	9100563638	1995-02-07	Male	542 West maredpalli, Hyderabad	

Screen 13 : All users

Spammer Identification

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

Home Users **Label Data** Un-label Data Followers Spammer Log out

## Hello Administrator

### Label Data View

S.No	Email	User Name	Shared TO	Message	Date	Mean Value	Variance	Likelihood Function
1	lx160cm@gmail.com	alex hales	Akhila Konakanchi	Hi akhila how are you doing in machine learning concepts	2018-10-24	93.8928571428571	800.533766233766	10.0
2	lx160cm@gmail.com	alex hales	Ravindra Narayana	Hello Ravindranadh, when you are coming to male, here maldevs very good beach.	2018-10-24	91.974358974359	859.012321012321	5.0

Screen 14 : Label data view from admin

Spammer Identification

localhost:8084/sigmm/AdminUnLabelledDatView.jsp

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

**Spammer Identification**

Home Users Label Data **Un-label Data** Followers Spammer Log out

## Hello Administrator

### Un-Label Data View

S.No	Email	User Name	Shared To	Message	Date	Mean Value	Variance	Likelihood Function
1	ravindranarayn@gmail.com	Ravindra Narayana	alex hales	<a href="#">View</a>	2018-10-26	95.3535875888817	838.699603873793	8.0
2	ravindranarayn@gmail.com	Ravindra Narayana	alex hales	<a href="#">View</a>	2018-10-26	1.94656488549618	492.601209366077	2.0

Windows taskbar: 2:35 PM 10/31/2018

Screen 15 : Unlabel data view from admin

Spammer Identification

localhost:8084/sigmm/FollowersTopics.jsp

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

**Spammer Identification**

Home Users Label Data Un-label Data **Followers** Spammer Log out

## Hello Administrator

### View All Users Post

S.No	Post Description	Creator Name	Creator Email	Date	Timestamp	Followers	Likes	Dislikes
1	The RSA Algorithm is first public key encryption to the workd	alex hales	lx160cm@gmail.com	2018-10-29	Mon, 29 Oct 2018 15:47:35 IST	<a href="#">Followers</a> 7	<a href="#">Likes</a> 5	<a href="#">Dislikes</a> 3
2	The Spammer in industrial mobile is a latest trend in the market..	Akhila Konakanchi	akhilakdataoint@gmail.com	2018-10-29	Mon, 29 Oct 2018 16:55:26 IST	<a href="#">Followers</a> 5	<a href="#">Likes</a> 5	<a href="#">Dislikes</a> 2
3	Yesterday rohit sharama played very well.He was very nice player.	Akhila Konakanchi	akhilakdataoint@gmail.com	2018-10-30	Tue, 30 Oct 2018 14:29:09 IST	<a href="#">Followers</a> 6	<a href="#">Likes</a> 3	<a href="#">Dislikes</a> 2

Windows taskbar: 2:35 PM 10/31/2018

Screen 16: All users posts

Spammer Identification x +







localhost:8084/sigmm/Spammers.jsp

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

**Spammer Identification** Home Users Label Data Un-label Data Followers **Spammer** Log out

## Hello Administrator

### Spammer Detection

S.No	User Name	First Name	Email	Mobile	Gender	Address	Image	Spam Status
1	Akhila Konakanchi	Akhila	akhilakdataoint@gmail.com	9600198765	Male	Aminapur, Warangal.		spammer unlock
2	Bhargav Sabarinadh	Bhargav	bhargav.podila@gmail.com	9581670405	Male	Cm nagur guntur		spammer unlock
3	Lakshmana Rao	Lakshmana	lakshmanarao145@gmail.com	9845009633	Male	65/3RT, Near Community Hall, SR Nagar, Hyderabad		spammer unlock
4	alex hales	alex	lx160cm@gmail.com	9849012345	Male	65/3RT SR Nagar, Hyderabad		Normal User
5	mounika Swaroop	mounika	mounikaswaroop@gmail.com	9600145689	Female	65/3RT, SR Nagar, Hyderabad		spammer unlock
	Ravindra					542 West maredpalli		spammer

Type here to search

2:36 PM 10/31/2018

Screen 17: Spammer detection

## **5. TESTING AND VALIDATION**

### **5.1 INTRODUCTION**

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

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

### **5.2 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 satisfaction, as shown by successfully 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 the software tester has knowledge of the inner workings, structure and language of the software, or at least its 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. 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.

## **5.2.1 UNIT TESTING**

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

### **Test strategy and approach**

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



### **Test objectives**

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

### **Features to be tested**

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

## **5.2.2 INTEGRATION TESTING**

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

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

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

## **5.2.3 ACCESSING TESTING**

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

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

## 5.3 TESTING METHODOLOGIES

**Black box Testing:** is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

**White box Testing:** is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing.

### Test Planning

Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.

This document involves the scope of testing,

- ☐ Objective of testing,
- ☐ Areas that need to be tested,
- ☐ Areas that should not be tested,
- ☐ Scheduling Resource Planning,
- ☐ Areas to be automated, various testing tools Used....

### Test Development

Test case Development (check list)

Test Procedure preparation. (Description of the Test cases)

**Test Execution:** 1. Implementation of test cases. Observing the result.

**Result Analysis:** 1. Expected value: is nothing but expected behavior of application.  
2. Actual value: is nothing but actual behavior of application

**Bug Tracing:** Collect all the failed cases, prepare documents.

**Reporting:** Prepare document (status of the application)

## **5.4 DESIGN OF TEST CASES AND SCENARIOS**

A Test Case is defined as a set of actions executed to verify a particular feature or functionality of the software application. A test case is an indispensable component of the Software Testing Lifecycle that helps validate the AUT (Application Under Test).

### **Test Scenario Vs Test Case**

Test scenarios are rather vague and cover a wide range of possibilities. Testing is all about being very specific.

For a Test Scenario: Check Login Functionality there many possible test cases are:

Test Case 1: Check results on entering valid User Id & Password

Test Case 2: Check results on entering Invalid User ID & Password

Test Case 3: Check response when a User ID is Empty & Login Button is pressed, and many more.

Testcase ID	Test Case	Desired Result	Actual Result	Validations( Pass/Fail)	Security Level		
					H	M	L
001USRLOG	The user leaves username textbox as Blank	Error message “Username cannot be empty” displays	Display the error message	pass	M		
002USRLOG	The user enters some value in the username textbox and leave the password textbox as Blank	Error message “Password cannot be empty” displays	Display the error message	pass	H		
003USRLOG	User enters username as “abcd” and password as “xxxx”	Error message “Invalid username password combination” displays	Display the error message	pass	M		
004USRLOG	User enters username as “testuser” and password as “password” and clicks the Submit button	The user is able to see the “user details screen”	Display the user page	pass	H		

Table 9: Testcases for user login

Testcase ID	Test Case	Desired Result	Actual Result	Validations (Pass/Fail)	Security Level		
					H	M	L
001 ADLOG	The admin leaves username textbox as Blank	Error message “Username cannot be empty” displays	Display the error message	pass	M		
002 ADLOG	The admin enters some value in the username textbox and leave the password textbox as Blank	Error message “Password cannot be empty” displays	Display the error message	pass	H		
003 ADLOG	admin enters username as “abcd” and password as “xxxx”	Error message “Invalid username password combination” displays	Display the error message	pass	M		
004 ADLOG	Admin enters username as “admin” and password as “password” and clicks the Submit button	The admin is able to see the details on the screen	Display the admin page	pass	H		

Table 10 : Testcases for Admin login

## 5.5 VALIDATION

- It is a dynamic mechanism of testing and validating the actual product.
- It always involves executing the code and it uses methods like Black Box Testing, White Box Testing, and non-functional testing.
- It checks whether the software meets the requirements and expectations of a customer.
- It can find bugs that the verification process cannot catch.
- Target is an actual product.
- With the involvement of testing team validation is executed on software code.
- It comes after verification.

## **6. CONCLUSION**

In order to solve the malicious attack problem in industrial mobile networks and reduce the computational complexity of using large cloud server datasets, this paper proposes SIGMM, a spammer identification model based on the Gaussian Mixture Model. We extract features related to labels from originally labeled data in a given dataset containing both labeled and unlabeled data, and visualize the data to add labels to the unlabeled data.

According to the characteristics of data presentation, each user data belongs to one distribution. Multidimensional features are divided into three groups, and SIGMM separates the two distributions based on these features. Finally, we performed simulations to evaluate the performance of SIGMM. The results show that even if the relationships among users are not taken into account, it can implement classification. Our work is based on binary classification, whereas in large networks, the types of users are varied and complex. Our future work will extend the categories of users to multi-classifications such as celebrity, advertiser, hacker, etc.

## **7. REFERENCES**

- [1] J. Miranda, N. Makitalo, J. Garcia-Alonso, J. Berrocal, T. Mikkonen, C. Canal, and J. M. Murillo, “From the internet of things to the internet of people,” *IEEE Internet Computing*, vol. 19, no. 2, pp. 40–47, 2015.
- [2] T. Qiu, A. Zhao, F. Xia, W. Si, and D. O. Wu, “Rose:Robustness strategy for scale-free wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2944–2959, 2017.
- [3] L. Yao, Q. Z. Sheng, and S. Dustdar, “Web-based management of the internet of things,” *IEEE Internet Computing*, vol. 19, no. 4, pp. 60–67, 2015.
- [4] T. Qiu, R. Qiao, and D. O. Wu, “Eabs: An event-aware backpressure scheduling scheme for emergency internet of things,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 72–84, 2017.
- [5] T. Qiu, K. Zheng, H. Song, M. Han, and B. Kantarci, “A local-optimization emergency scheduling scheme with self-recovery for smart grid,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3195–3205, 2017.
- [6] S. Lu, V. H. Nascimento, J. Sun, and Z. Wang, “Sparsityaware adaptive link combination approach over distributed networks,” *Electronics Letters*, vol. 50, no. 18, pp. 1285–1287, 2014.
- [7] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao, “Spammer behavior analysis and detection in user generated content on social networks,” in *IEEE International Conference on Distributed Computing Systems*, May. 16- 18, 2012, pp. 305–314.



## **ANNEXURE - I : CODE OF PROJECT**

### **DBConnection.java**

```
package com.sigmm.db;
import java.sql.Connection;
import java.sql.DriverManager;

/**
 *
 * @author Ramu
 * Maloth */
public class DBConnection {
    public static Connection con = null;
    public static Connection getConnection(){
        try {
            DriverManager.registerDriver(new
com.mysql.jdbc.Driver());
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/sigmm","
root","root");
            if(con!=null){
                return con;
            }
            else{return con;}
        } catch (Exception e) {
            System.out.println("Database Error "+e.getMessage());
        }
        return con;
    }
}
```

## **UserRegisterAction.java**

```
package com.sigmm.action;

/**
 *
 * @author Ramu
 * Maloth */
public class UserRegisterAction extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter(); String
        fname = request.getParameter("fname");
        String lname = request.getParameter("lname");
        String email = request.getParameter("email");
        String pswd = request.getParameter("pswd");
        String mobile = request.getParameter("mobile");
        String dob = request.getParameter("dob");
        String gender = request.getParameter("gender");
        String address = request.getParameter("address");
        String userName = fname+" "+lname;
        Connection con = null;
        PreparedStatement ps = null;

        try {
            con = DBConnection.getConnection();
            String sqlQuery = "insert into
userregisters(username,firstname,lastname,email,pswd,mobile,dob,gender,address,spamstatus) values(?,?,?,?,?,?,?,?,?,?)";
            ps = con.prepareStatement(sqlQuery);
            ps.setString(1, userName);
            ps.setString(2, fname);
            ps.setString(3, lname);
```

```

        ps.setString(4, email);
        ps.setString(5, pswd);
        ps.setString(6, mobile);
        ps.setString(7, dob);
        ps.setString(8, gender);
        ps.setString(9, address);
        ps.setString(10, "NormalUser");
        int no = ps.executeUpdate();

        if(no>0) {

response.sendRedirect("UserRegister.jsp?msg=success");
            }else{
                response.sendRedirect("UserRegister.jsp?msg=faield");
            }

        } catch (Exception e) {
            System.out.println("User register Error
"+e.getMessage());
            response.sendRedirect("UserRegister.jsp?msg=faield");
        }
    }

    @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);
    }

}

```

### **CreatingEvent.java**

```

package com.sigmm.action;

/**
 *
 * @author Ramu
 * Maloth */
public class CreatingEvent extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)

        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String eventPost = request.getParameter("event");
        String postdate = request.getParameter("postdate");
        String email = request.getParameter("email");
        String username = request.getParameter("username");

        final DateFormat currentTimeandDate = new
SimpleDateFormat("E, dd MMM yyyy HH:mm:ss z");
        Date currentDate = new Date();
        String currentTime =
currentTimeandDate.format(currentDate);
        Connection con = null;
        PreparedStatement ps = null;

        try {

```

```

        java.sql.Date cDate = new
java.sql.Date(new java.util.Date().getTime());
        con = DBConnection.getConnection();
        String sqlQuery = "insert into
userposts(postname,creatername,useremail,postdate,ctimestamp)
values(?,?,?,?,:)";

        ps = con.prepareStatement(sqlQuery);
        ps.setString(1, eventPost);
        ps.setString(2, username);
        ps.setString(3, email);
        ps.setDate(4, cDate);
        ps.setString(5, currentTime); int no
        = ps.executeUpdate();
        if(no > 0){

response.sendRedirect("UserCreateEvent.jsp?msg=success");
        }else{

response.sendRedirect("UserCreateEvent.jsp?msg=faield");
        }
        }catch(Exception ex){
            ex.printStackTrace();
            System.out.println("Error at "+ex.getMessage());

response.sendRedirect("UserCreateEvent.jsp?msg=faield"); }
        finally {
            out.close();
        }
    }

    @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);
    }

}

```

#### **UnLabeledDataShareAction.java**

```

package com.sigmm.action;
import com.sigmm.db.DBConnection;
import com.sigmm.utility.GetFileExtension;
import com.sigmm.utility.StatisticsApproach;
import java.awt.image.BufferedImage; import
java.io.BufferedReader; import
java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import javax.imageio.ImageIO;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part; /**

```

\*

```

    * @author Ramu
    Maloth */
    @MultipartConfig
    public class UnLabeledDataShareAction extends HttpServlet {

        @Override
        protected void doPost(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
            double mean = 0.0;
            double variance = 0.0;
            double stdDev = 0.0;
            double median = 0.0;
            double titaValue = 0.0;
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            String fromEmail = request.getParameter("fromEmail");
            String fromUserName =
request.getParameter("fromUserName");
            String toUserEmail =
request.getParameter("toUserEmail");
            String toUserName = request.getParameter("toUserName");
            Part filePart = request.getPart("file");
            String fileName = getFileName(filePart);
            InputStream is = filePart.getInputStream();
            InputStream img = filePart.getInputStream();
            java.sql.Date currentDate = new java.sql.Date(new
java.util.Date().getTime());
            String fileExtension =
GetFileExtension.getFileExtension(fileName);

            if(fileExtension.equalsIgnoreCase(".txt")){
                String str = "";
                StringBuffer buf = new StringBuffer();

```

```

        BufferedReader reader = new BufferedReader(new
InputStreamReader(is));
        if (is != null) {
            while ((str = reader.readLine()) != null)
                { buf.append(str + "\n");
                }
        }
        String data = buf.toString();
        double []arr = new double[data.length()];
        System.out.println("The ascii value of each character is:
");
        for(int i=0;i<arr.length;i++){
            arr[i] = data.charAt(i); // assign the integer value
of character i.e ascii
            System.out.print(" "+arr[i]);
        }
        System.out.println("");
        StatisticsApproach algorithm = new
StatisticsApproach(arr);

        mean = algorithm.getMean();
        variance = algorithm.getVariance();
        stdDev = algorithm.getStdDev();
        median = algorithm.median();
        out.println("Mean =
"+mean+"\nvariance="+variance+"\nStandard
Variance="+stdDev+"\nMedian="+median);
        double result = Math.round(mean)/Math.round(stdDev);

        double xyz = Math.round(result) * Math.round(stdDev);
        // out.println("\nResults "+xyz);
        titaValue = Math.round(mean) - Math.round(xyz);
        out.println("tita Value "+titaValue);

    }else{

```



```

        BufferedImage buffImage = ImageIO.read(is);
        // double mean =
StatisticsApproach.meanofAnImage(buffImage);
        // out.println("Mean "+mean);
        double []data =
StatisticsApproach.getImageVariancea(buffImage);
        System.out.println("Data "+data.length);
        StatisticsApproach algorithm = new
StatisticsApproach(data);
        mean = algorithm.getMean();
        variance = algorithm.getVariance();
        stdDev = algorithm.getStdDev();
        median = algorithm.median();
        out.println("Mean =
"+mean+"\nvariance="+variance+"\nStandard
Variance="+stdDev+"\nMedian="+median);
        double result = Math.round(mean)/Math.round(stdDev);

        double xyz = Math.round(result) * Math.round(stdDev);
        // out.println("\nResults "+xyz);
        titaValue = Math.round(mean) - Math.round(xyz);
        out.println("tita Value "+titaValue);
    }

    Connection con = null;
    PreparedStatement ps = null;

    try {
        con = DBConnection.getConnection();
        String sqlQuery = "insert into
unlabelledataset(fromuseremail,fromusername,touseremail,tosuernam
e,message,sharedate,meanval,variance,stdDev,median,likelihoodfunc
tion,filename) values(?,?,?,?,?,?,?,?,?,?,?,?,?)";
        ps = con.prepareStatement(sqlQuery);

```

```

        ps.setString(1, fromEmail);
        ps.setString(2, fromUserName);
        ps.setString(3, toUserEmail);
        ps.setString(4, toUserName);
        ps.setBinaryStream(5, img);
        ps.setDate(6, currentDate);
        ps.setDouble(7, mean);
        ps.setDouble(8, variance);
        ps.setDouble(9, stdDev);
        ps.setDouble(10, median);
        ps.setDouble(11, titaValue);
        ps.setString(12, fileName);
        int no = ps.executeUpdate();
        if(no>0){
            out.println("<html>");
            out.println("<body>");
            out.println("<h2>Your Messages Posted you can close
this window</h2>");
            out.println("<button type=\"button\"
onclick=\"javascript:window.close()\">Close</button><br/>");
            out.println("</body>");
            out.println("</html>");
        }
        catch (Exception e) {
            System.out.println("Error "+e.getMessage());
        }
    }

    private String getFileName(Part filePart) {
        for (String cd : filePart.getHeader("content-
disposition").split(";")) {
            if (cd.trim().startsWith("filename")) {
                return cd.substring(cd.indexOf('=') + 1).trim()
                    .replace("\"", "");
            }
        }
    }

```

```

        }
        return null;
    }
}

```

### **LabeledDataShareAction.java**

```

package com.sigmm.action;
import com.sigmm.db.DBConnection;
import com.sigmm.utility.StatisticsApproach;
import java.io.IOException;
import java.io.PrintWriter; import
java.sql.Connection; import
java.sql.PreparedStatement; import
javax.servlet.ServletException; import
javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 *
 * @author Ramu
 * Maloth */
public class LabeledDataShareAction extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String fromEmail = request.getParameter("fromEmail");
        String fromUserName =
request.getParameter("fromUserName");

        String toUserEmail = request.getParameter("toUserEmail");
        String toUserName = request.getParameter("toUserName");

```

```

        String message = request.getParameter("shareMessage");

        java.sql.Date currentDate = new
        java.sql.Date(new java.util.Date().getTime());

        Connection con = null;
        PreparedStatement ps = null;

        try {

            double []arr = new double[message.length()];
            System.out.println("The ascii value of each character is:
");
            for(int i=0;i<arr.length;i++){
                arr[i] = message.charAt(i); // assign the integer
                value of character i.e ascii
                System.out.print(" "+arr[i]);
            }

            System.out.println("");
            StatisticsApproach algorithm = new
            StatisticsApproach(arr);

            double mean = algorithm.getMean(); double
            variance = algorithm.getVariance(); double
            stdDev = algorithm.getStdDev(); double
            median = algorithm.median();

            // out.println("Mean =
            "+mean+"\nvariance="+variance+"\nStandard
            Variance="+stdDev+"\nMedian="+median);

            double result = Math.round(mean)/Math.round(stdDev);

            double xyz = Math.round(result) * Math.round(stdDev);
            // out.println("\nResults "+xyz);

            double titaValue = Math.round(mean) - Math.round(xyz);
            System.out.println("tita Value "+titaValue);

```

```

        /*
            double finalTita =          mean-titaValue;
long spamStatus =
Math.round(Math.abs(1/titaValue));
out.println("\nTita Value "+spamStatus); */

        con = DBConnection.getConnection();
        String sqlQuery = "insert into
labeledataset(fromuseremail,fromusername,touseremail,tosuename,me
ssage,sharedate,meanval,variance,stdDev,median,likelihoodfunction
)
values(?,?,?,?,?,?,?,?,?,?,?,?,?)";

        ps = con.prepareStatement(sqlQuery);
        ps.setString(1, fromEmail);
        ps.setString(2, fromUserName);
        ps.setString(3, toUserEmail);
        ps.setString(4, toUserName);
        ps.setString(5, message);
        ps.setDate(6, currentDate);
        ps.setDouble(7, mean);
        ps.setDouble(8, variance);
        ps.setDouble(9, stdDev);
        ps.setDouble(10, median);
        ps.setDouble(11,titaValue); int no =
ps.executeUpdate();
        if(no>0){
            out.println("<h2>Your Messages Posted you can close
this window</h2>");
            out.println("<button type=\"button\"
onclick=\"javascript:window.close()\">Close</button><br/>");
        }
    }catch(Exception ex){
        System.out.println("Labeled Data share Error
"+ex.getMessage());
    } finally {
        out.close();
    }
}

```

```

        }
    }

    @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);
    }

}

```

### **EuclideanDist.java**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sigmm.utility;

/**
 *
 * @author Ramu
 * Maloth */
import java.util.Random;

```

```

public class EuclideanDist {
    public static void main(String[] args) {
        EuclideanDist euc = new EuclideanDist();
        Random rnd = new Random();

        //      int N = Integer.parseInt(args[0]);

        //double[] a = new double[N]; //88.53658536585365
        // double[] b = new double[N]; //103.0

        double[] a = new double[88]; //88.53658536585365
        double[] b = new double[103]; //103.0

        euc.print(euc.init(a, rnd));
        euc.print(euc.init(b, rnd));
        System.out.println(euc.distance(a, b));
    }

    private double[] init(double[] src, Random rnd)
    { for (int i = 0; i < src.length; i++) {
        src[i] = rnd.nextDouble();
    }
    return src;
}

    private double distance(double[] a, double[] b)
    { double diff_square_sum = 0.0;
    for (int i = 0; i < a.length; i++) {
        diff_square_sum += (a[i] - b[i]) * (a[i] - b[i]);
    }
    return Math.sqrt(diff_square_sum);
}

    private void print(double[] x) {

```

```
        for (int j = 0; j < x.length; j++) {  
            System.out.print(" " + x[j] + " ");  
        }  
        System.out.println();  
    }  
}
```