A Project Report On

Credit Card Fraud Detection Using Adaboost And Majority Voting

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

Kukatpally, Hyderabad-500 085, Telangana, India

In partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

Information Technology

By

K.VAISHNAVI [16E31A1214]
Y.ABHINAYA [16E31A1231]

Under the guidance of

Mrs.K.B.V.K.Nagasree

Assistant Professor

IT Dept

Department of Information Technology

MAHAVEER INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to JNTU Hyderabad, Approved by AICTE)

Vyasapuri, Bandlaguda, Post: Keshavgiri, Hyderabad-500 005

2019-2020

MAHAVEER INSTITUTE OF SCIENCE ANDTECHNOLOGY

(Affiliated to JNTU Hyderabad, Approved by AICTE)

Vyasapuri, Bandlaguda, Post: Keshavgiri, Hyderabad-500005

CERTIFICATE

This is to certify that this project work report entitled **“CREDIT CARD FRAUD DETECTION USING ADABOOST AND MAJORITY VOTING”** which is being submitted by **K.VAISHNAVI[16E31A1214]** and **Y.ABHINAYA[16E31A1231]** in partial fulfillment for the award of the Degree of **Bachelor Of Technology** in **Information Technology,** affiliated of **Jawaharlal Nehru Technological University,** Hyderabad and is a record of the bonafide work carried out by them under our guidance during 2019-2020.

Signature of Project Guide

Mrs.K.B.V.K.Nagasree

(Assistant Professor)

Signature of Head of the Department

Dr.A.Nanda Gopal Reddy

External Examiner

Principal

Dr. K. S. S. S.N Reddy

ACKNOWLEDGMENT

We would like to express our deep felt appreciation and gratitude to **Mrs. K. B. V. K. Nagasree**, Assistant Professor, Department of IT our project guide, for her skillful guidance, constant supervision, timely suggestion, keen interest and encouragement in completing the individual seminar within the stipulated time.

We wish to express our gratitude to **Mr. V. Naresh Kumar Reddy,** Project Coordinator, who has shown keen interest and even rendered her valuable guidance in terms of suggestions and encouragement extended to us with an immense care and zeal.

We express our profound sense of gratitude to **Dr. A. Nanda Gopal Reddy** Head of Department, IT, who has served as a host of valuable corrections and for providing us time and amenities to complete this project.

We express our thanks to **Dr. K. S. S. S. N. Reddy**, Principal of our college and the management of **Mahaveer Institute of Science &Technology** for providing excellent academic environment in the college.

We wish to express our gratitude to the **Members of staff** and all others who helped us in more than one way. We would also like to thank the **Lab assistants and Programmers** for helping us through our project.

K.VAISHNAVI	**[16E31A1214]**
Y.ABHINAYA	**[16E31A1231]**

DECLARATION

We hereby declare that the project entitled "**Credit Card fraud Detection using AdaBoost and majority voting**" submitted to partial fulfillment of the requirements for award of the degree of Bachelor of Technology at **Mahaveer Institute of Science & Technology**, affiliated to **Jawaharlal Nehru Technology University, Hyderabad** in authentic work and has not been submitted to any other university institute for award of any degree.

K.Vaishnavi[16E31A1214]

Y.Abhinaya [16E31A1231]

ABSTRACT

Credit card fraud is a serious problem in financial services. Billions of dollars are lost due to credit card fraud every year. There is a lack of research studies on analyzing real-world credit card data owing to confidentiality issues. In this project, machine learning algorithms are used to detect credit card fraud. Standard models are firstly used. Then, hybrid methods which use AdaBoost and majority voting methods are applied.

To evaluate the model efficacy, a publicly available credit card data set is used. Then, a real-world credit card data set from a financial institution is analyzed. In addition, noise is added to the data samples to further assess the robustness of the algorithms. The experimental results positively indicate that the majority voting method achieves good accuracy rates in detecting fraud cases in credit cards.

CONTENTS

CHAPTER 3

SYSTEM ANALYSIS

CHAPTER 4

SYSTEM DESIGN

LIST OF FIGURES

TITLE **PAGE NO**

LIST OF SCREENS

TITLE **PAGE NO**

CHAPTER-1

INTRODUCTION

Fraud is a wrongful or criminal deception aimed to bring financial or personal gain . In avoiding loss from fraud, two mechanisms can be used: fraud prevention and fraud detection.

Fraud prevention is a proactive method, where it stops fraud from happening in the first place. On the other hand, fraud detection is needed when a fraudulent transaction is attempted by a fraudster. Credit card fraud is concerned with the illegal use of credit card information for purchases. Credit card transactions can be accomplished either physically or digitally. In physical transactions, the credit card is involved during the transactions.

In digital transactions, this can happen over the telephone or the internet. Cardholders typically provide the card number, expiry date, and card verification number through telephone or website. With the raise of e-commerce in the past decade, the use of credit cards has increased dramatically. The number of credit card transactions in 2011 in Malaysia were at about 320 million, and increased in 2015 to about 360 million. Along with the rise of credit card usage, the number of fraud cases have been constantly increased. While numerous authorization techniques have been in place, credit card fraud cases have not hindered effectively. Fraudsters favour the internet as their identity and location are hidden. The rise in credit card fraud has a big impact on the financial industry. The global credit card fraud in 2015 reached to a staggering USD $21.84 billion. Loss from credit card fraud affects the merchants, where they bear all costs, including card issuer fees, charges, and administrative charges.

Since the merchants need to bear the loss, some goods are priced higher, or discounts and incentives are reduced. Therefore, it is imperative to reduce the loss, and an effective fraud detection system to reduce or eliminate fraud cases is important. There have been various studies on credit card fraud detection. Machine learning and related methods are most commonly used, which include artificial neural networks, rule-induction techniques, decision trees, logistic regression, and support vector machines. These methods are used either standalone or by combining several methods together to form hybrid models. Total of twelve machine learning algorithms are used for detecting credit card fraud. The algorithms range from standard neural networks to deep learning models. They are evaluated using both benchmark and real world credit card data sets. In addition, the AdaBoost and majority voting methods are applied for forming hybrid models. To further evaluate the robustness and reliability of the models, noise is added to the real-world data set.

The key contribution of this project is the evaluation of a variety of machine learning models with a real-world credit card data set for fraud detection. While other researchers have used various methods on publicly available data sets, the data set used in this project is extracted from actual credit card transaction information over three months. Related studies on single and hybrid machine learning algorithms for financial applications is given. The experiments with both benchmark and real world credit card data sets are presented.

CHAPTER-2

LITERATURE SURVEY

1. Credit card detection using HMM(2010)

As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. They have model the sequence of operations in credit card transaction processing using a Hidden Markov Model (HMM) and show how it can be used for the detection of frauds. An HMM is initially trained with the normal behavior of a cardholder. If an incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, they try to ensure that genuine transactions are not rejected.

2. Credit card fraud detection: A Case Study(2015)

As fraudsters are increasing day by day. And fallacious transactions are done by the credit card and there are various types of fraud. So to solve this problem combination of technique is used like Genetic Algorithm, Behavior Based Technique and Hidden Markov Model. By this transaction is tested individually and whatever suits the best is further proceeded. And the foremost goal is to detect fraud by filtering the above techniques to get better result.

2.2 SOFTWARE ENVIRONMENT

2.2.1 JAVA TECHNOLOGY

About Java

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

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

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

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called java byte codes the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed.

API support all these kinds of programs

It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- Applets: The set of conventions used by applets.
- Networking: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.

- Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- Security: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- Software components: Known as JavaBeans, can plug into existing component architectures.
- Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages.
- The 100% Pure Java Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
- Write once, run anywhere: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

Java Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

2.2.2 TOMCAT SERVER

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java

Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Weblogic, is one of the popular application server).To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.

2.2.3 HTML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web.

The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Advantages

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

2.2.4 JAVASCRIPT

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser’s display accordingly.

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it.

Here are a few things we can do with JavaScript :

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser’s status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.

2.2.5 JDBC

JDBC is a Java API for executing SQL statements. JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database.

JDBC connectivity

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

- Perform connection and authentication to a database server.
- Manager transactions.
- Move SQL statements to a database engine for preprocessing and execution.
- Execute stored procedures.
- Inspect and modify the results from Select statements.

JDBC Driver Types

The JDBC drivers are fit into one of four categories:

- JDBC-ODBC bridge driver.
- Native-API partly-Java driver.
- JDBC-Net pure Java driver.
- Native-protocol pure Java driver.

2.2.6 JAVA SERVER PAGES (JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each others work.

Components

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

Steps in the execution of a JSP Application

- The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
- This request is transferred to the Java Web Server. At the server side Java Web Server receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
- JSP engine is program which can understands the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side.
- This Servlet is loaded in the memory and then it is executed and the result is given back to the Java Web Server and then it is transferred back to the result is given back to the Java Web Server and then it is transferred back to the client.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Three methods to detect fraud are presented. Firstly, clustering model is used to classify the legal and fraudulent transaction using data clusterization of regions of parameter value. Secondly, Gaussian mixture model is used to model the probability density of credit card user's past behavior so that the probability of current behavior can be calculated to detect any abnormalities from the past behavior. Lastly, Bayesian networks are used to describe the statistics of a particular user and the statistics of different fraud scenarios. The main task is to explore different views of the same problem and see what can be learned from the application of each different technique.

Disadvantages of Existing System

- There is no Majority Voting technique for credit card fraud detection.
- There is no Machine Learning Techniques in the existing system.

3.2 PROPOSED SYSTEM

Total of twelve machine learning algorithms are used for detecting credit card fraud. The algorithms range from standard neural networks to deep learning models. They are evaluated using both benchmark and real world credit card data sets. In addition, the AdaBoost and robustness and reliability of the models, noise is added to the real-world data set. The key contribution of this paper is the evaluation of a variety of machine learning models with a realworld credit card data set for fraud detection.

Advantages of Proposed System

- The system is very fast due to AdaBoost Technique.
- Effective Majority Voting techniques.

3.3 FEASIBILITY STUDY

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

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

3.3.1 OPERATIONAL FEASIBILITY

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

3.3.2 ECONOMICAL FEASIBILITY

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

3.3.3 TECHNICAL FEASIBILITY

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

3.4 SYSTEM REQUIREMENTS

3.4.1. SOFTWARE REQUIREMENTS:

Operating System	:	Windows 10/8/7/95/98/2000/XP
Application Server	:	Tomcat5.0/6.X/8.X
Front End	:	HTML
Scripts	:	JavaScript
Server side Script	:	Java Server Pages
Database Connectivity	:	Mysql Java
Version	:	jdk 1.8

3.4.2. HARDWARE REQUIREMENTS:

RAM	:	4GB
Hard Disk	:	57 GB
Processor	:	intel core i3

CHAPTER- 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

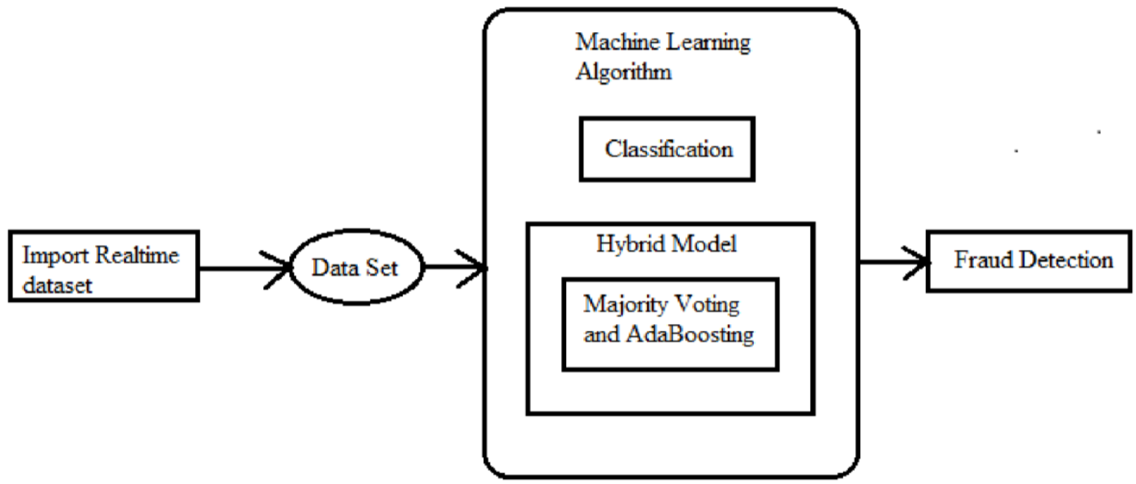


Fig 4.1.1 Architecture

4.2 DATA FLOW DIAGRAM:

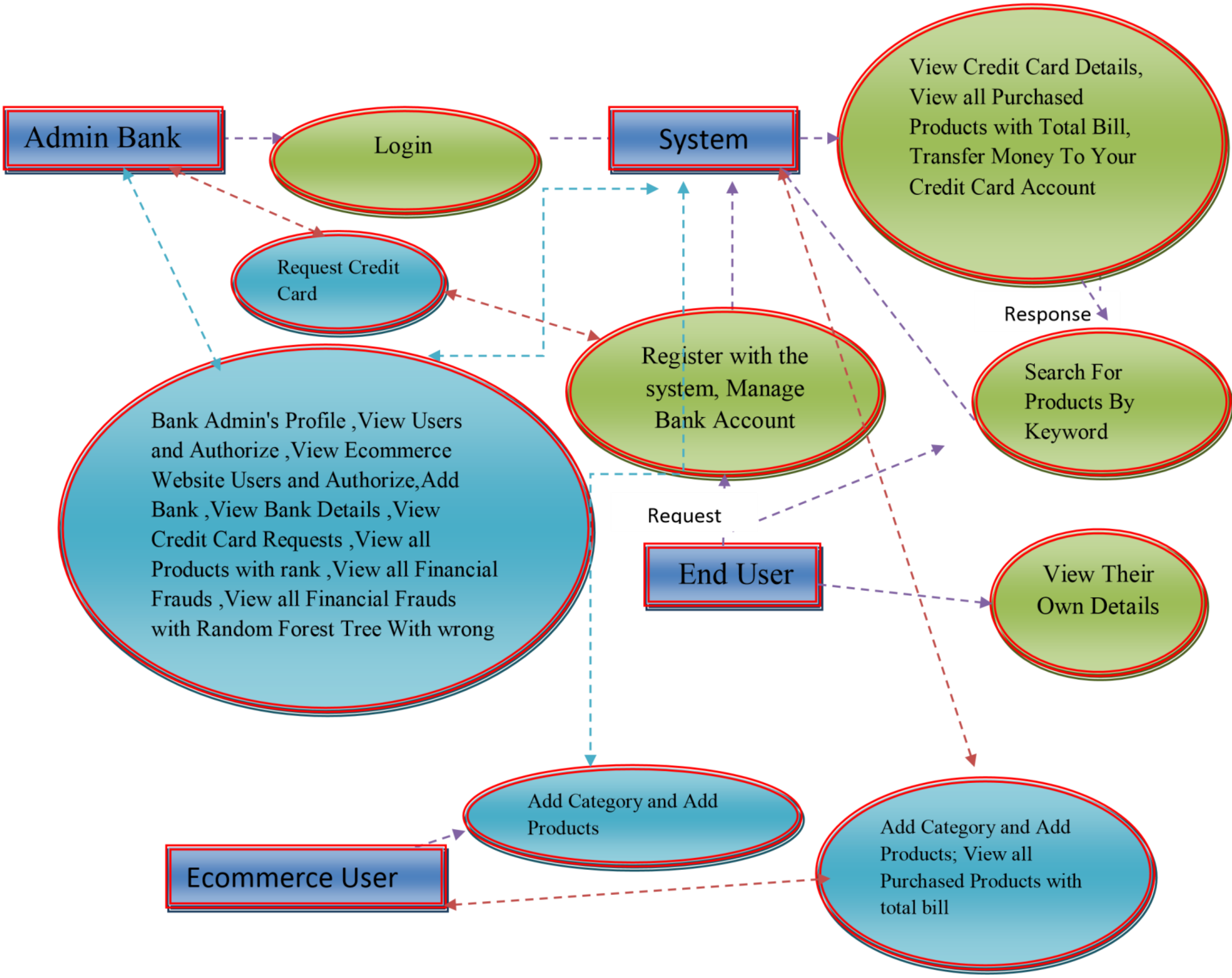
1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

- ### 4.2.1 DATA FLOW DIAGRAM

4.3 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 goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model


Admin Bank
Login
System
View Credit Card Details, View all Purchased Products with Total Bill, Transfer Money To Your Credit Card Account
Request Credit Card
Response
Register with the system, Manage Bank Account
Search For Products By Keyword
Bank Admin's Profile ,View Users and Authorize ,View Ecommerce Website Users and Authorize,Add Bank ,View Bank Details ,View Credit Card Requests ,View all Products with rank ,View all Financial Frauds ,View all Financial Frauds with Random Forest Tree With wrong
Request
End User
View Their Own Details
Add Category and Add Products
Ecommerce User
Add Category and Add Products; View all Purchased Products with total bill

and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

4.3.1 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.

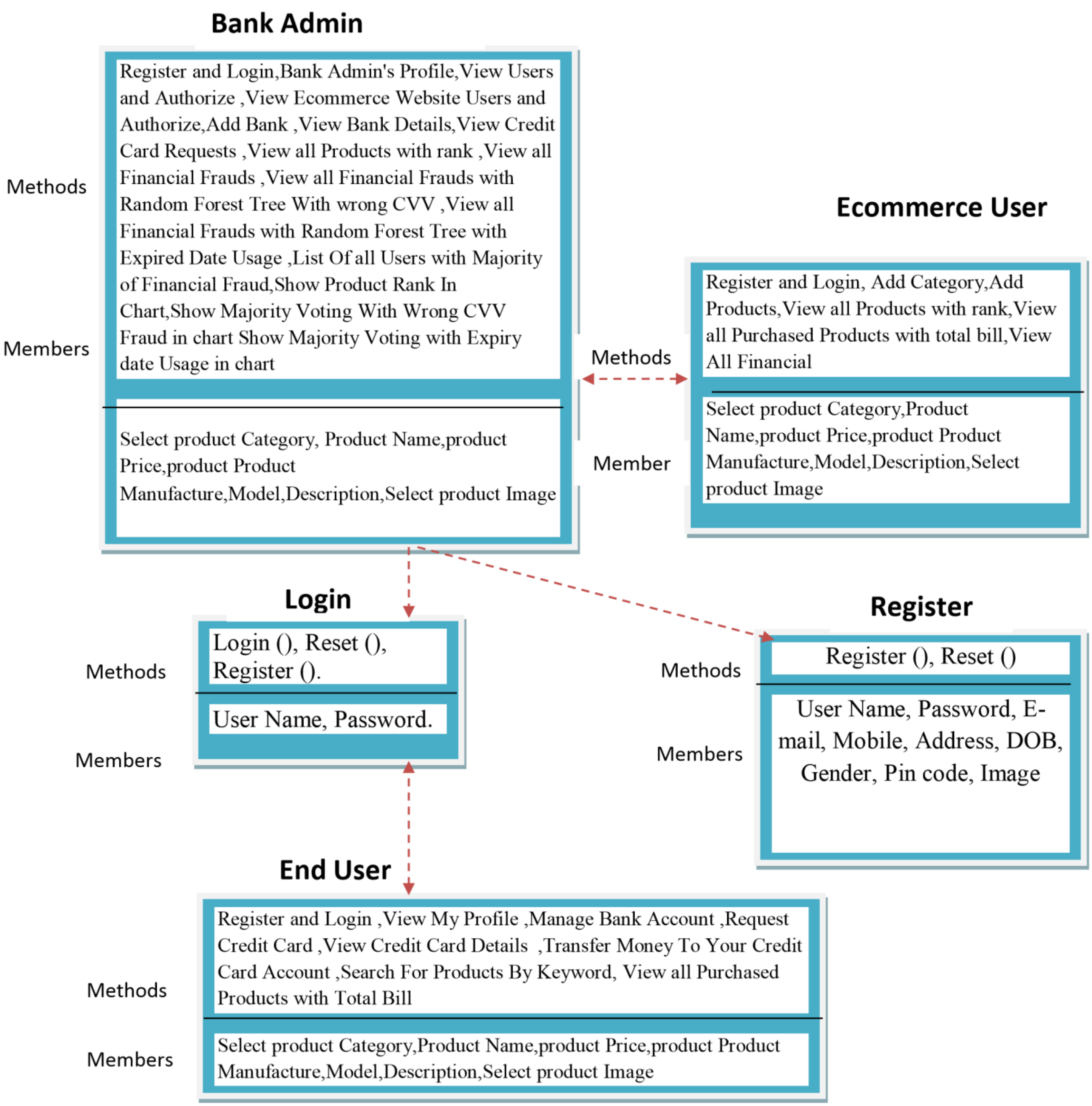

Bank Admin
Methods
Register and Login,Bank Admin's Profile,View Users and Authorize ,View Ecommerce Website Users and Authorize,Add Bank ,View Bank Details,View Credit Card Requests ,View all Products with rank ,View all Financial Frauds ,View all Financial Frauds with Random Forest Tree With wrong CVV ,View all Financial Frauds with Random Forest Tree with Expired Date Usage ,List Of all Users with Majority of Financial Fraud,Show Product Rank In Chart,Show Majority Voting With Wrong CVV Fraud in chart Show Majority Voting with Expiry date Usage in chart
Members
Select product Category, Product Name,product Price,product Product Manufacture,Model,Description,Select product Image
Ecommerce User
Methods
Register and Login, Add Category,Add Products,View all Products with rank,View all Purchased Products with total bill,View All Financial
Member
Select product Category,Product Name,product Price,product Product Manufacture,Model,Description,Select product Image
Login
Methods
Login (), Reset (), Register ().
Members
User Name, Password.
Register
Methods
Register (), Reset ()
Members
User Name, Password, E-mail, Mobile, Address, DOB, Gender, Pin code, Image
End User
Methods
Register and Login ,View My Profile ,Manage Bank Account ,Request Credit Card ,View Credit Card Details ,Transfer Money To Your Credit Card Account ,Search For Products By Keyword, View all Purchased Products with Total Bill
Members
Select product Category,Product Name,product Price,product Product Manufacture,Model,Description,Select product Image

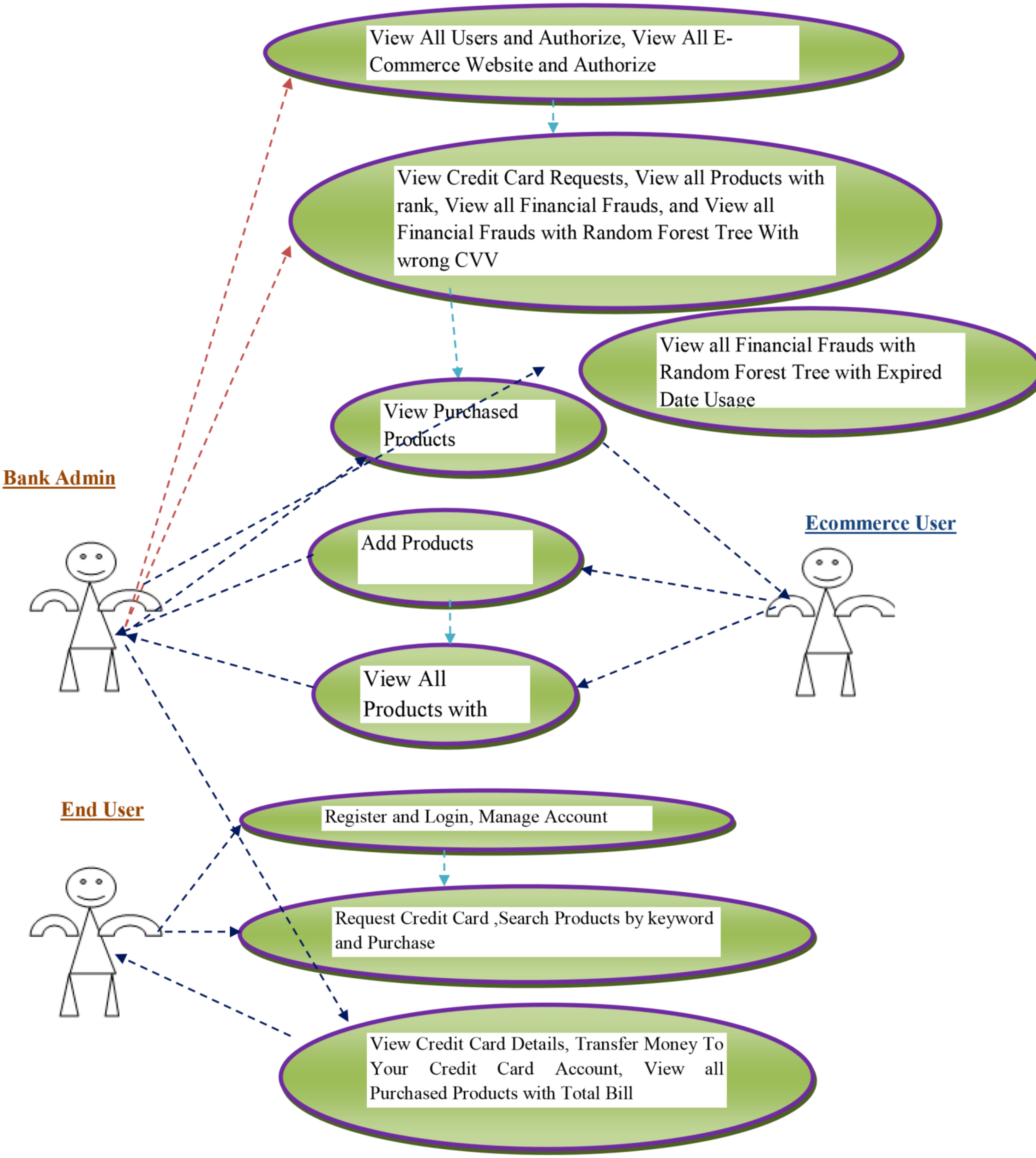
Fig 4.3.1 Class Diagram

4.3.2 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.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

Fig 4.3.2 Use Case Diagram


View All Users and Authorize, View All E-Commerce Website and Authorize
View Credit Card Requests, View all Products with rank, View all Financial Frauds, and View all Financial Frauds with Random Forest Tree With wrong CVV
View all Financial Frauds with Random Forest Tree with Expired Date Usage
View Purchased Products
Bank Admin
Add Products
Ecommerce User
View All Products with
End User
Register and Login, Manage Account
Request Credit Card ,Search Products by keyword and Purchase
View Credit Card Details, Transfer Money To Your Credit Card Account, View all Purchased Products with Total Bill

4.3.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

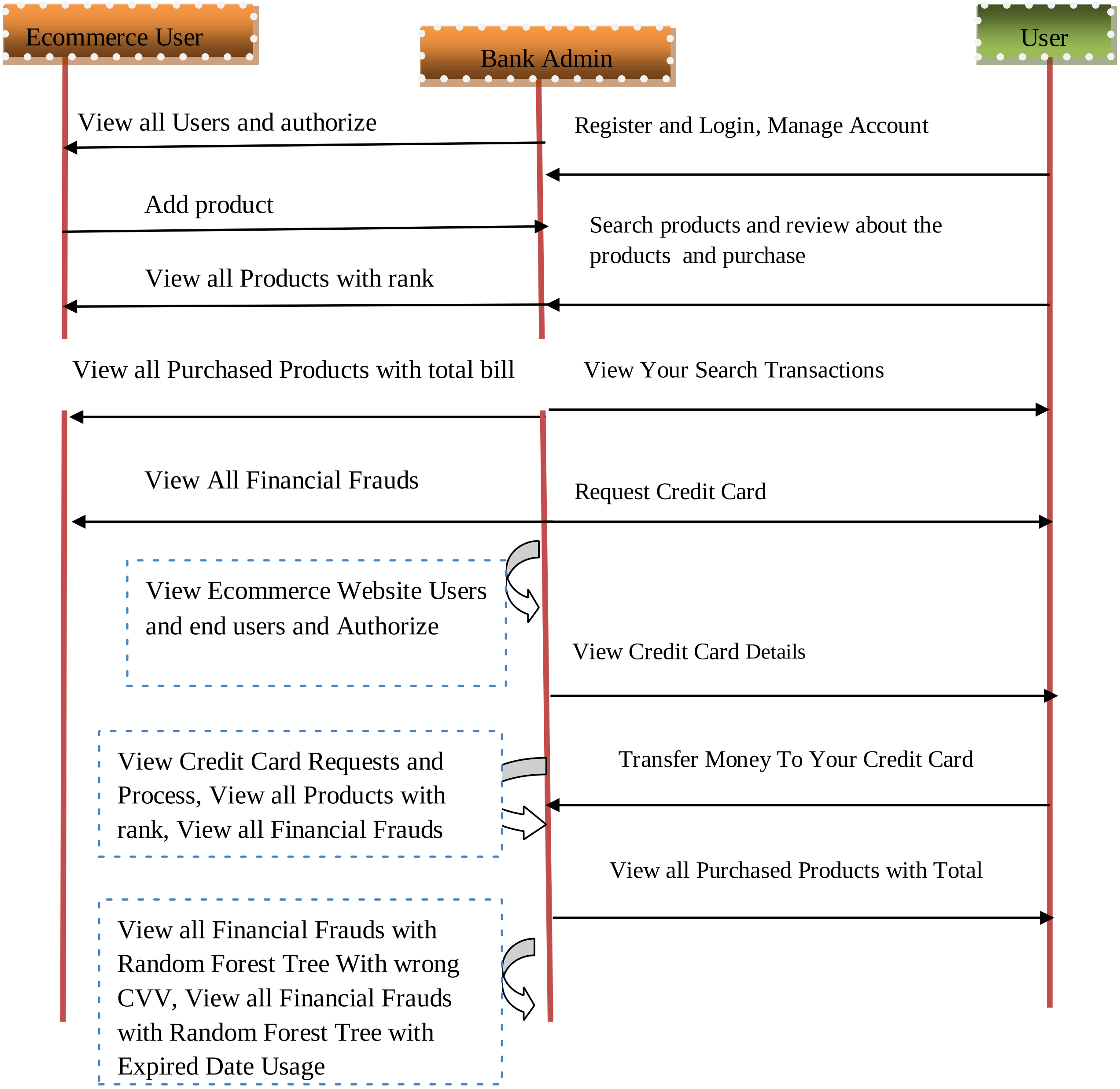

Ecommerce User
Bank Admin
User
View all Users and authorize
Register and Login, Manage Account
Add product
Search products and review about the products and purchase
View all Products with rank
View all Purchased Products with total bill
View Your Search Transactions
View All Financial Frauds
Request Credit Card
View Ecommerce Website Users and end users and Authorize
View Credit Card Details
View Credit Card Requests and Process, View all Products with rank, View all Financial Frauds
Transfer Money To Your Credit Card
View all Purchased Products with Total
View all Financial Frauds with Random Forest Tree With wrong CVV, View all Financial Frauds with Random Forest Tree with Expired Date Usage

Fig 4.3.3 Sequence Diagram

CHAPTER-5

IMPLEMENTATION AND CODING

5.1 MODULES:

- Bank Admin
- E-Commerce user
- End user

5.2 MODULES DESCRIPTION

Bank Admin

In this module, the Admin has to login by using valid user name and password. After login successful he can do some operations such as Bank Admin's Profile ,View Users and Authorize ,View Ecommerce Website Users and Authorize, Add Bank ,View Bank Details

,View Credit Card Requests, View all Products with rank ,View all Financial Frauds ,View all Financial Frauds with Random Forest Tree With wrong CVV ,View all Financial Frauds with Random Forest Tree with Expired Date Usage ,List Of all Users with Majority of Financial Fraud ,Show Product Rank In Chart ,Show Majority Voting With Wrong CVV Fraud in chart ,Show Majority Voting with Expiry date Usage in chart.

View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

View Chart Results

Show Product Rank In Chart, Show Majority Voting With Wrong CVV Fraud in chart, Show Majority Voting with Expiry date Usage in chart.

Ecommerce User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like, Add Category, Add Products, View all Products with rank, and View all Purchased Products with total bill, View All Financial Frauds.

End User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like, View My Profile, Manage Bank Account,

Request Credit Card, View Credit Card Details, Transfer Money to Your Credit Card Account, Search for Products by Keyword, View all Purchased Products with Total Bill.

5.3 SAMPLE CODE

Sample code for Index Page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Home Page</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />
```

```
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />


<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-quicksand.js"></script>


<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>


<script type="text/javascript" src="js/script.js"></script>


<script type="text/javascript" src="js/coin-slider.min.js"></script>


<style type="text/css">


<!-- .style1 {

font-size: 24px;

color: #FF0000;


}


.style3 { color:

#FF0000; font-

weight: bold;


}


-->


</style>


</head>
```

```
<body>

<div class="main">

<div class="header">

<div class="header_resize">
<div class="logo">

<h1 align="center"><a href="index.html" class="style1">Credit card fraud detection   using

AdaBoost and majority voting</a></h1>

</div>

<div class="menu_nav">

<ul>

<li class="active"><a href="index.html"><span>Home Page</span></a></li>

<li><a href="AdminLogin.jsp"><span>Bank Admin</span></a></li>

<li><a href="EcommerceLogin.jsp"><span>Ecommerce</span></a></li>

<li><a href="UserLogin.jsp"><span>User</span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">
```

```
<div id="coin-slider"> <a href="#"><img src="images/slide1.jpg" width="935" height="285"
alt="" /> </a> <a href="#"><img src="images/slide2.jpg" width="935" height="285" alt="" />

</a> <a href="#"><img src="images/slide3.jpg" width="935" height="285" alt="" /> </a> </div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>
</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">

<h2 align="center"><span class="style1"><span class="style3">Credit card fraud detection using
AdaBoost and majority voting</span></h2>

<p align="center"><img src="images/Home.jpg" width="629" height="335" /></p>

<p align="justify" class="style3">Credit card fraud is a serious problem in financial services.
Billions of dollars are lost due to credit card fraud every year. There is a lack of research studies
on analyzing real-world credit card data owing to confidentiality issues. In this paper, machine
learning algorithms are used to detect credit card fraud. Standard models are firstly used. Then,
hybrid methods which use AdaBoost and majority voting methods are applied. To evaluate the
model efficacy, a publicly available credit card data set is used. Then, a real-world credit card data
set from a financial institution is analyzed. In addition, noise is added to the data samples to further
```

```
assess the robustness of the algorithms. The experimental results positively indicate that the
majority voting method achieves good accuracy rates in detecting fraud cases in credit cards.</p>

<p align="center">&nbsp;</p>

</div>

</div>

<div class="sidebar">

<div class="searchform">

<form id="formsearch" name="formsearch" method="post" action="#">
<span>

<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80"
value="Search our ste:" type="text" />

</span>

<input name="button_search" src="images/search.gif" class="button_search" type="image" />

</form>

</div>

<div class="clr"></div>

<div class="gadget">

<h2 class="star"><span>Sidebar</span> Menu</h2>

<div class="clr"></div>
```

```
<ul class="sb_menu">

<li><a href="index.html">Home</a></li>

<li><a href="AdminLogin.jsp">Bank Admin</a></li>

<li><a href="EcommerceLogin.jsp">Ecommerce</a></li>

<li><a href="UserLogin.jsp">User</a></li>

</ul>

</div>

 </div>

<div class="clr"></div>

</div>
</div>

<div class="footer"></div>

</div>

<div align=center></div>

</body>

</html>
```

Sample code for Admin Login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>BankAdminLoginPage</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-quicksand.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">
<!-- .style1 {

font-size: 24px;

color: #FF0000;


}


.style3 {color: #333333}
```

```
.style4 {color: #CC6600}

.style6 {color: #FF0000; font-weight: bold; }

-->

</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="logo">

<h1 align="center"><a href="index.html" class="style1">Credit card fraud detection using
AdaBoost and majority voting</a></h1>

</div>

<div class="menu_nav">

<ul>
<li class="active"><a href="index.html"><span>Home Page</span></a></li>

<li><a href="AdminLogin.jsp"><span>Bank Admin</span></a></li>

<li><a href="EcommerceLogin.jsp"><span>Ecommerce</span></a></li>
```

```
<li><a href="UserLogin.jsp"><span>User</span></a></li>

<li><a href="#"><span>Contact Us</span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div  id="coin-slider">  <a  href="#"><img  src="images/slide1.jpg"  width="935"  height="285"
alt="" /> </a> <a href="#"><img src="images/slide2.jpg" width="935" height="285" alt="" />
</a> <a href="#"><img src="images/slide3.jpg" width="935" height="285" alt="" /> </a> </div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">
<h2 class="style4" style="color:#CC6600">Welcome To Bank Admin Login..</h2>
```

```
<div class="clr"></div>

<div class="post_content">

<p align="center"><img src="images/Login.png" width="237" height="133" /></p>

<form id="form1" name="form1" method="post"
action="Authentication.jsp?type=<%="bankadmin"%>">

<table width="464" border="0" cellspacing="2" cellpadding="2">

<tr>

<td width="170" height="33"><div align="justify" class="style6">Select Bank
(required)</div></td>

<td width="274"><label>

<select name="bank">

<option>Select</option>

<option>SBI Bank</option>

<option>Karnataka Bank</option>

<option>Corporation Bank</option>

<option>Canara Bank</option>

<option>Indian Bank</option>

</select>
```

```
</label></td>

</tr>

<tr>
<td width="200" height="46" align="center"><div align="justify" class="style6"><span
class="style34">

<label for="name">Bank Admin Name (required)</label>

</span></div> </td>

<td width="253"><input id="name" name="adminid" class="text" /></td>

</tr>

<tr>

<td height="30" align="center"><div align="justify" class="style6">

<label for="pass">Password (required)</label>

</div></td>

<td><input type="password" id="pass" name="pass" class="text" /></td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>
```

```
</tr>

<tr>

<td>&nbsp;</td>

<td><input name="imageField" type="submit"  class="LOGIN" id="imageField" value="Login"
/>

<span class="style3">New Bank Admin?</span><span class="style4"><a
href="AdminRegister.jsp" class="style4"> <b>Register</b> </a></span></td> </tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

</form>

</div>

<div class="clr"></div>

</div>

</div>

<div class="sidebar">
```

```
<div class="searchform">

<form id="formsearch" name="formsearch" method="post" action="#">

<span>

<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80" value="Search our ste:" type="text" />

</span>

<input name="button_search" src="images/search.gif" class="button_search" type="image" />

</form>

</div>
<div class="clr"></div>

<div class="gadget">

<h2 class="star"><span>Sidebar</span> Menu</h2>

<div class="clr"></div>

<ul class="sb_menu">

<li><a href="index.html">Home</a></li>

<li><a href="AdminLogin.jsp">Bank Admin</a></li>

<li><a href="CompanyLogin.jsp">Ecommerce</a></li>

<li><a href="UserLogin.jsp">User</a></li>
```

```
</ul>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="footer"></div>

</div>

<div align=center></div>

</body>

</html>
```

Sample code for E-commerce login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>EcommerceLoginPage</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-quicksand.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!-- .style1 {

font-size: 24px;

color: #FF0000;


}


.style3 {color: #333333}
.style4 {color: #CC6600}


.style5 {color: #FF0000}


.style7 {color: #FF0000; font-weight: bold; }


-->
```

```
</style>

</head>

<body>

<div class="main">

<div class="header">

<div class="header_resize">

<div class="logo">

<h1 align="center"><a href="index.html" class="style1">Credit card fraud detection using AdaBoost and majority voting</a></h1>

</div>

<div class="menu_nav">

<ul>

<li class="active"><a href="index.html"><span>Home Page</span></a></li>

<li><a href="AdminLogin.jsp"><span>Bank Admin</span></a></li>

<li><a href="EcommerceLogin.jsp"><span>Ecommerce</span></a></li>

<li><a href="UserLogin.jsp"><span>User</span></a></li>

<li><a href="#"><span>Contact Us</span></a></li>
</ul>
```

```
</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"> <a href="#"><img src="images/slide1.jpg" width="935" height="285"
alt="" /> </a> <a href="#"><img src="images/slide2.jpg" width="935" height="285" alt="" />
</a> <a href="#"><img src="images/slide3.jpg" width="935" height="285" alt="" /> </a> </div>

<div class="clr"></div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">

<h2 class="style4" style="color:#CC6600">Welcome To Ecommerce User Login..</h2>

<div class="clr"></div>

<div class="post_content">
```

```
<p><img src="images/Login.png" width="227" height="151" /></p>
<form id="form1" name="form1" method="post"
action="Authentication.jsp?type=<%="euser"%>">

<table width="464" border="0" cellspacing="2" cellpadding="2">

<tr>

<td width="197" height="46" align="center"><span class="style34">

<label for="name"><span class="style4 style3">Select Ecommerce Website</span> </label>

</span> </td>


<td width="253"><select id="s2" name="esite" style="width:150px;" class="text">

<option>--Select--</option>

<option>Amazon</option>

<option>Flipkart</option>

<option>Snapdeal</option>

<option>ebay</option>

</select></td>

</tr>

<tr>

<td width="197" height="46" align="center"><span class="style34"> <span class="style7">
```

```
<label for="name">User Name (required)</label>

</span> </span> </td>

<td width="253"><input id="name" name="euserid" class="text" /></td> </tr>
<tr>

<td height="40" align="center" class="style5 style34"><span class="style5"><strong>Password
(required)</strong></span></td>

<td><input type="password" id="pass" name="pass" class="text" /></td>

</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td>&nbsp;</td>

<td><span class="style16">

<input name="imageField" type="submit"  class="LOGIN" id="imageField" value="Login" />

<span class="style5"> New User?</span></span><a href="EcommerceRegister.jsp"
class="style30"><strong> Register </strong></a></td>
```

```
</tr>

<tr>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

</form>
</div>

<div class="clr"></div>

</div>

</div>

<div class="sidebar">

<div class="searchform">

<form id="formsearch" name="formsearch" method="post" action="#">

<span>

<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80"
value="Search our ste:" type="text" />

</span>
```

```
<input name="button_search" src="images/search.gif" class="button_search" type="image" />

</form>

</div>

<div class="clr"></div>

<div class="gadget">

<h2 class="star"><span>Sidebar</span> Menu</h2>

<div class="clr"></div>

<ul class="sb_menu">

<li><a href="index.html">Home</a></li>



</ul>

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="footer"></div>
```

```
</div>

<div align=center></div>

</body>

</html>
```

Sample code for user login

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>UserLoginPage</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-quicksand.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>
```

```
<style type="text/css">

<!-- .style1 {

font-size: 24px;

color: #FF0000;


}

.style3 {color: #333333}

.style4 {color: #CC6600}

.style5 {color: #FF0000}

.style9 {color: #FF0000; font-weight: bold; }

.style10 {font-weight: bold}

-->

</style>

</head>

<body>

<div class="main">
<div class="header">

<div class="header_resize">

<div class="logo">
```

```
<h1 align="center"><a href="index.html" class="style1">Credit card fraud detection using
AdaBoost and majority voting</a></h1>

</div>

<div class="menu_nav">

<ul>

<li class="active"><a href="index.html"><span>Home Page</span></a></li>

<li><a href="AdminLogin.jsp"><span>Bank Admin</span></a></li>

<li><a href="EcommerceLogin.jsp"><span>Ecommerce</span></a></li>

<li><a href="UserLogin.jsp"><span>User</span></a></li>

<li><a href="#"><span>Contact Us</span></a></li>

</ul>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"> <a href="#"><img src="images/slide1.jpg" width="935" height="285"
alt="" /> </a> <a href="#"><img src="images/slide2.jpg" width="935" height="285" alt="" />
</a> <a href="#"><img src="images/slide3.jpg" width="935" height="285" alt="" /> </a> </div>

<div class="clr"></div>
</div>
```

```
<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">

<h2 class="style4" style="color:#CC6600">Welcome To User Login..</h2>

<p class="style4" style="color:#CC6600"><img src="images/Login.png" width="228" height="145" /></p>

<div class="clr"></div>

<div class="post_content">

<form id="form1" name="form1" method="post" action="Authentication.jsp?type=<%="user"%>">

<table width="570" border="0" align="left" cellpadding="2" cellspacing="2">

<tr>

<td width="185" align="center" valign="middle"><label for="email">

<div align="center" class="style4 style3">Select Bank (required)</div>
```

```
</label></td>

<td><p align="left">

<select name="bank">
<option>Select</option>

<option>SBI Bank</option>

<option>Karnataka Bank</option>

<option>Corporation Bank</option>

<option>Canara Bank</option>

<option>Indian Bank</option>

</select></p></td>

</tr>

<tr>

<td height="46" align="center" valign="middle"><span class="style5 style35"><strong>

<label for="label">

 </label>

</strong></span>

<span class="style9"><label for="label">

</span><div align="center" class="style5 style35"><strong>User Name
```

```
(required)</strong></div>

<span class="style9"></label>

<label for="name"></label>

</span></td>

<td width="371"><input id="name" name="userid" class="text" /></td>

</tr>
<tr>

<td height="40" align="center" valign="middle"><span class="style5 style35"><strong>

<label for="email">

</label>

</strong></span>

<span class="style9">

<label for="email">

</span>

<div align="center" class="style5 style35"><strong>Password (required)</strong></div>

<span class="style9">

</label>

</span></td>
```

```
<td><input type="password" id="pass" name="pass" class="text" /></td>

</tr>

<tr>

<td height="47">&nbsp;</td>

<td><input name="imageField" type="submit"  class="LOGIN" id="imageField" value="Login"
/>

<span class="style38 style5 style3">New User?</span><span class="style10"><strong><a

href="UserRegister.jsp" class="style30"> Register </a></strong></span></td> </tr>


<tr>
<td height="49"></td>

</tr>

</table>

</form>

</div>

<div class="clr"></div>

</div>

</div>

<div class="sidebar">

<div class="searchform">
```

```
<form id="formsearch" name="formsearch" method="post" action="#">

<span>

<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80" value="Search our ste:" type="text" />

</span>

<input name="button_search" src="images/search.gif" class="button_search" type="image" />

</form>

</div>

<div class="clr"></div>

<div class="gadget">

<h2 class="star"><span>Sidebar</span> Menu</h2>
<div class="clr"></div>

<ul class="sb_menu">

<li><a href="index.html">Home</a></li>

</ul>

</div>

</div>

<div class="clr"></div>
```

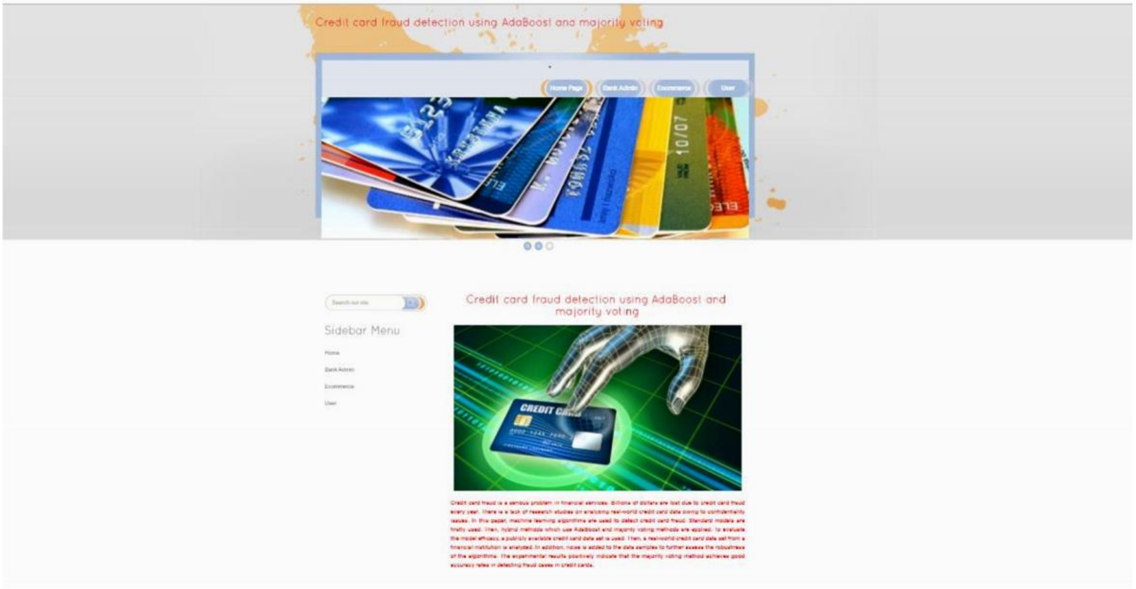
```
</div>

</div>

<div class="footer"></div>

</div>

<div align=center></div>

</body>

</html>
```

CHAPTER-6

OUTPUT SCREENS

Screen No: 1

Screen Title: Home Page



Credit card fraud detection using AdaBoost and majority voting

Credit card fraud detection using AdaBoost and majority voting

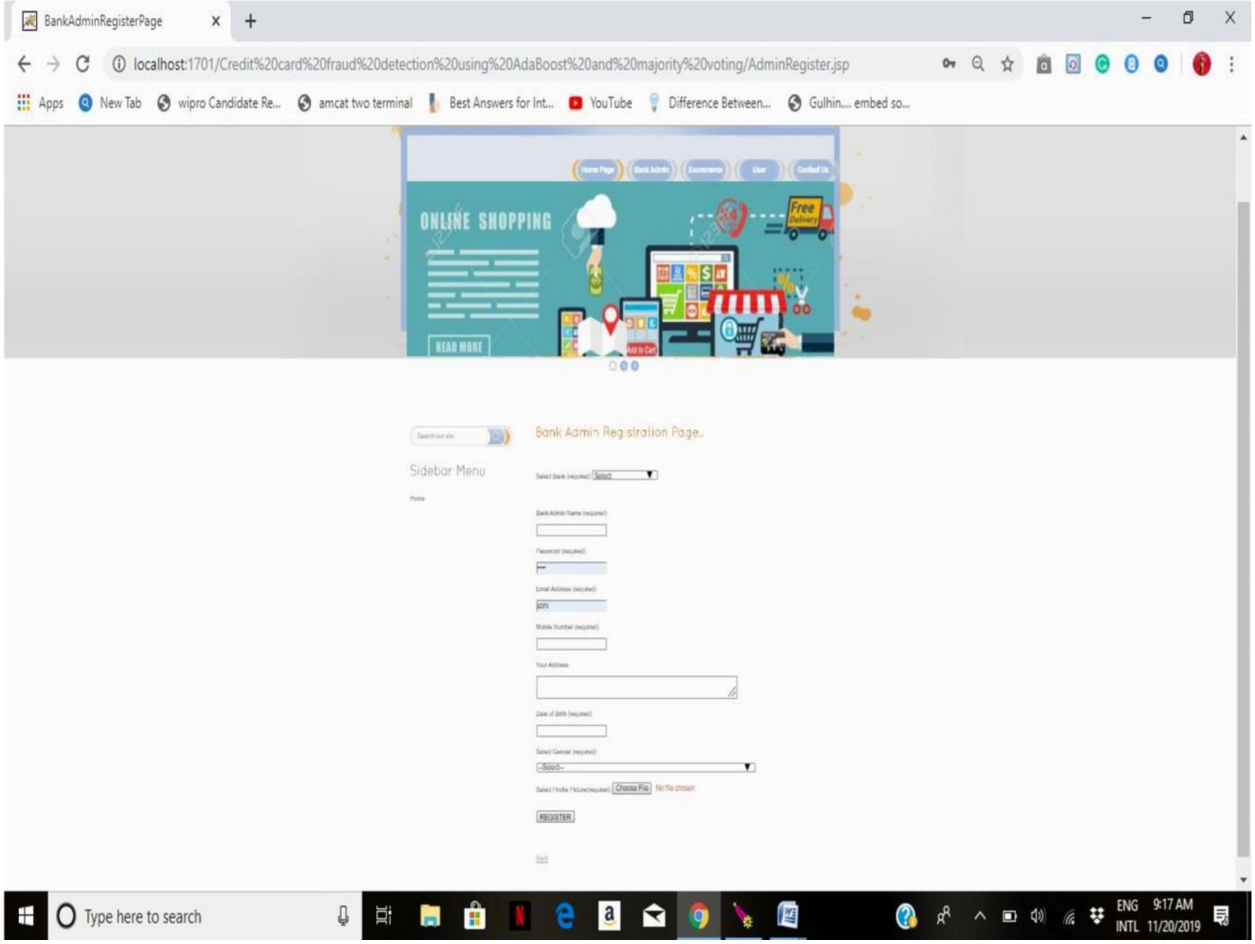
Credit card fraud is a serious problem in financial services. Billions of dollars are lost due to credit card fraud every year. There is a lack of research studies on analyzing real-world credit card data owing to confidentiality issues. In this paper, machine learning algorithms are used to detect credit card fraud. Standard models are firstly used. Then, hybrid methods which use AdaBoost and majority voting methods are applied. To evaluate the model efficacy, a publicly available credit card data set is used. Then, a real-world credit card data set from a financial institution is analyzed. In addition, noise is added to the data samples to further assess the robustness of the algorithms. The experimental results positively indicate that the majority voting method achieves good accuracy rates in detecting fraud cases in credit cards.

Home Page Description:

Home page contains menus such as homepage, admin, ecommerce, user is used for admin, ecommerce user and user login. Sign up page is used for registration of new users.

Screen No:2

Screen Title: Admin Registration Page

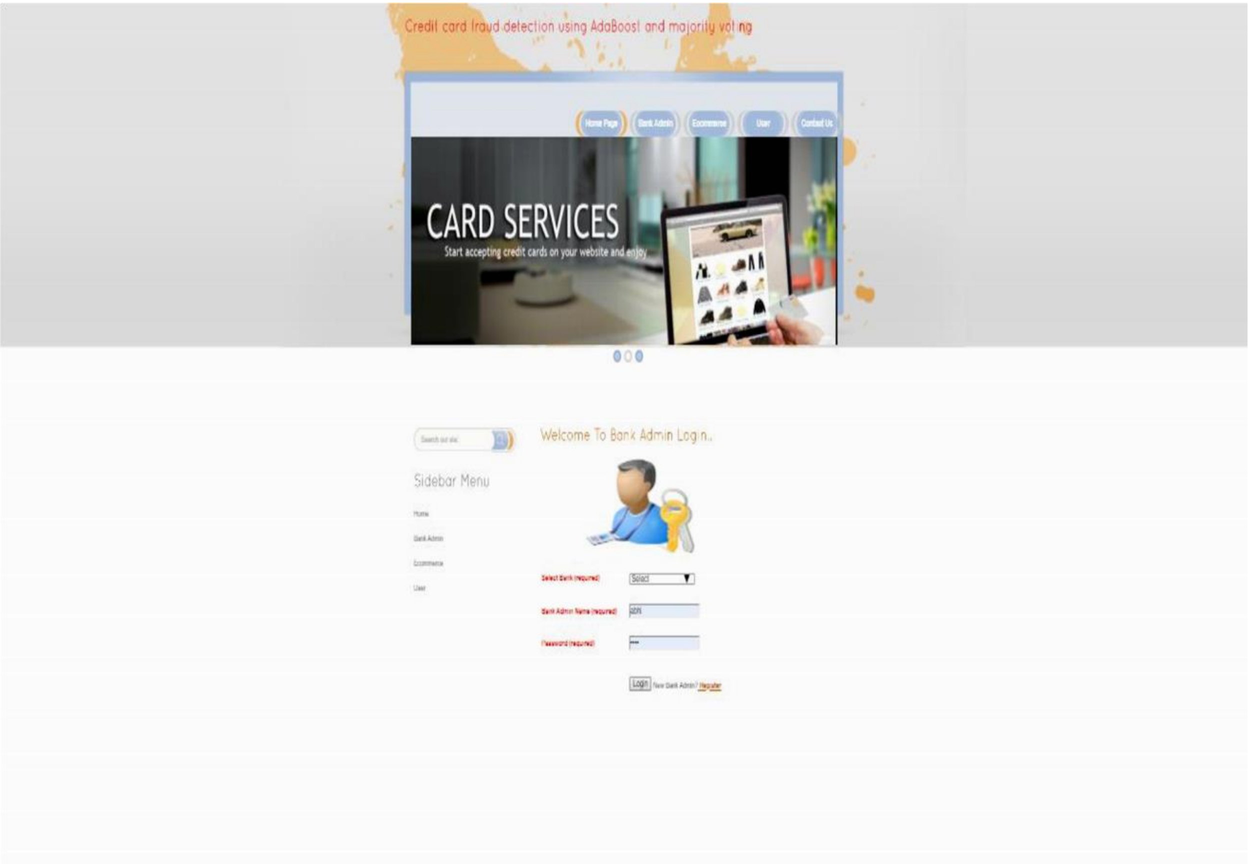


Admin Registration Page Description

Here we can register if we don't have account. We should enter all the details without fail, or else it will not register.

Screen No:3

Screen Title: Admin Login Page



Credit card fraud detection using AdaBoost and majority voting

Welcome To Bank Admin Login..

Select Bank (required) Select

Bank Admin Name (required) abhi

Password (required) ****

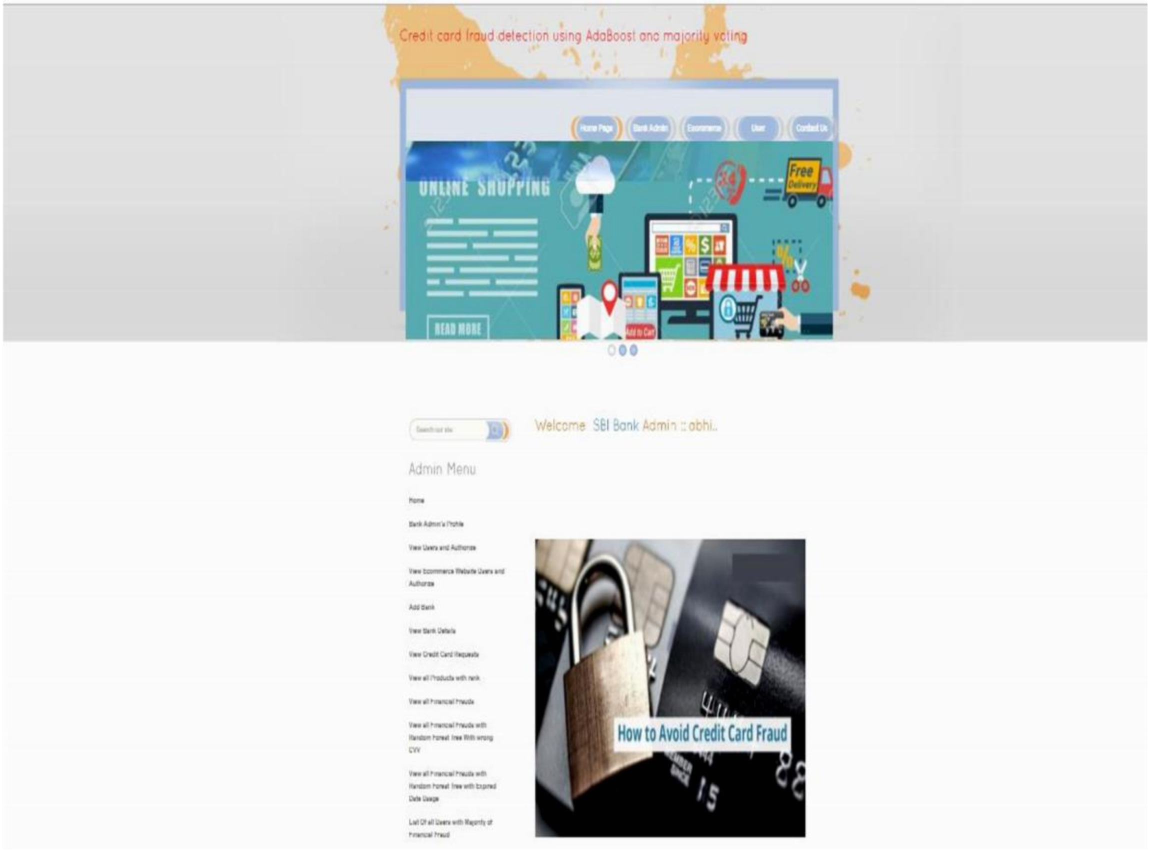
New Bank Admin? Register

Admin login page Description:

If we want to login to the admin page, here we can login if the login credentials are valid .If the login credentials are wrong it will shows a message that the details you have entered are not valid.

Screen No: 4

Screen Title: Admin Home Page



Credit card fraud detection using AdaBoost anc majority voting

Welcome SBI Bank Admin :: abhi..

Admin Menu

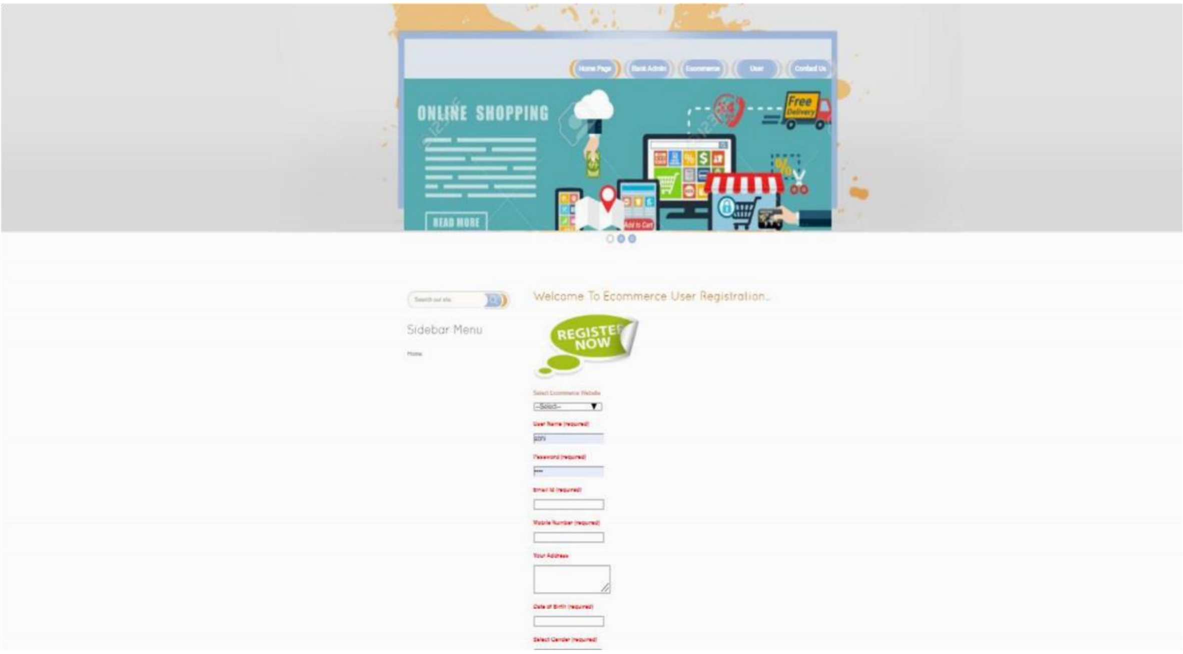
- Home
- Bank Admin's Profile
- View Users and Authorize
- View Ecommerce Website Users and Authorize
- Add Bank
- View Bank Details
- View Credit Card Requests
- View all Products with rank
- View all Financial Frauds
- View all Financial Frauds with Random Forest Tree With wrong CVV
- View all Financial Frauds with Random Forest Tree with Expired Date Usage
- List Of all Users with Majority of Financial Fraud

Admin Home Page Description

Admin home page contains various sub modules like Admin’s profile ,view users and authorize, Add bank, View bank details, Request credit card etc.,

Screen No: 5

Screen Title:E commerce Registration Page



Welcome To Ecommerce User Registration..

Select Ecommerce Website

--Select--

User Name (required)

Password (required)

Email Id (required)

Mobile Number (required)

Your Address

Date of Birth (required)

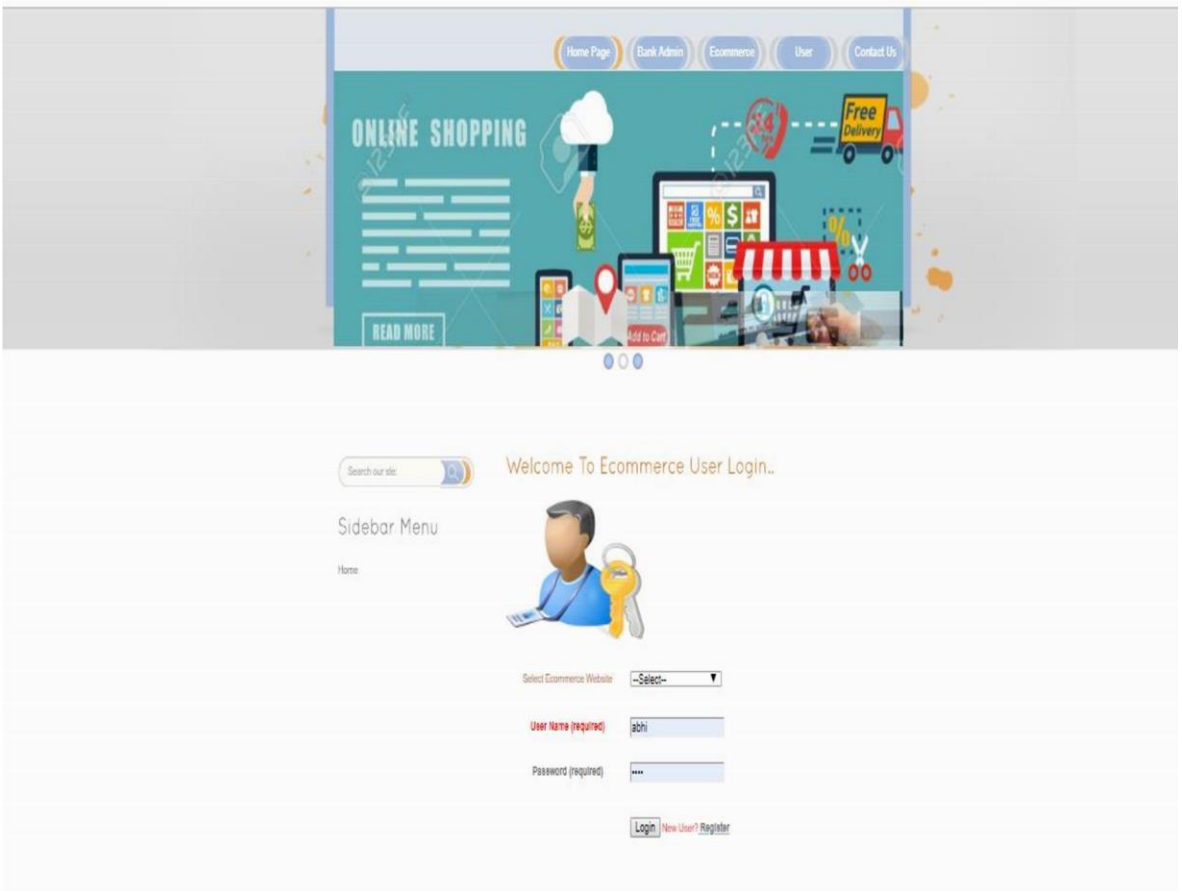
Select Gender (required)

E-commerce Registration Page Description

Here we can register if we don’t have account. We should enter all the details without fail, or else it will not register.

Screen No: 6

Screen Title: E commerce Login Page



Welcome To Ecommerce User Login..

Select Ecommerce Website --Select--

User Name (required) abhi

Password (required) ****

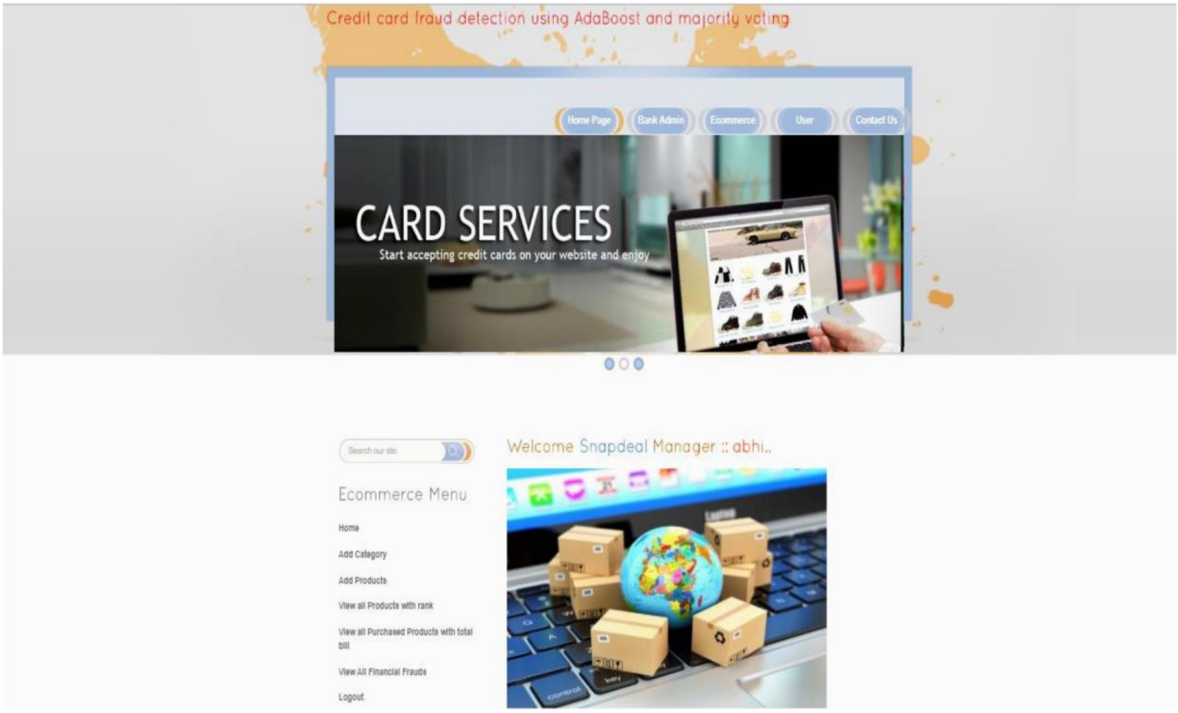
Login New User? Register

E-commerce Login Page Description

This is the login page of E-commerce user. If we want to login to the E-commerce page, here we can login if the login credentials are valid .If the login credentials are wrong it will shows a message that the details you have entered are not valid.

Screen No: 7

Screen Title: E commerce Home Page


Credit card fraud detection using AdaBoost and majority voting
Home Page
Bank Admin
Ecommerce
User
Contact Us
CARD SERVICES
Start accepting credit cards on your website and enjoy
Search our site
Ecommerce Menu
Home
Add Category
Add Products
View all Products with rank
View all Purchased Products with total bill
View All Financial Frauds
Logout

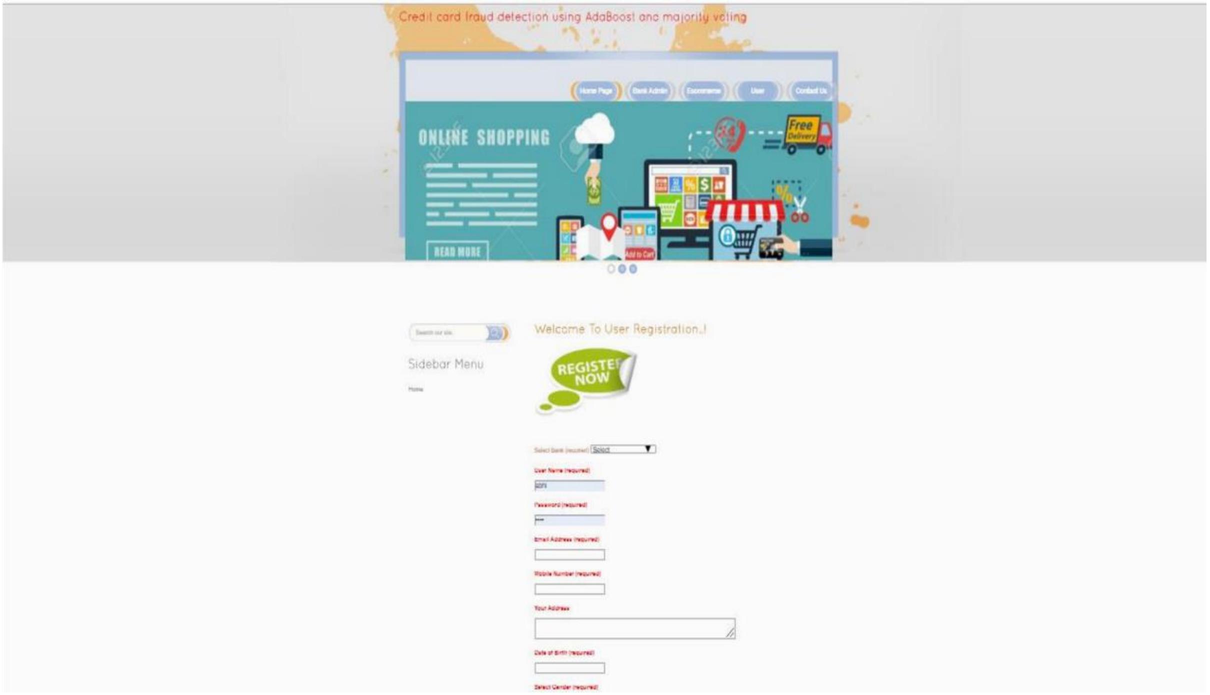
Welcome Snapdeal Manager :: abhi..

E-commerce Home Page Description

E-commerce home page contains sub modules like Add category, Add products, View all Purchases etc.,

Screen No: 8

Screen Title: User Registration page



Welcome To User Registration..!

Select Bank (required) Select

User Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

Your Address

Date of Birth (required)

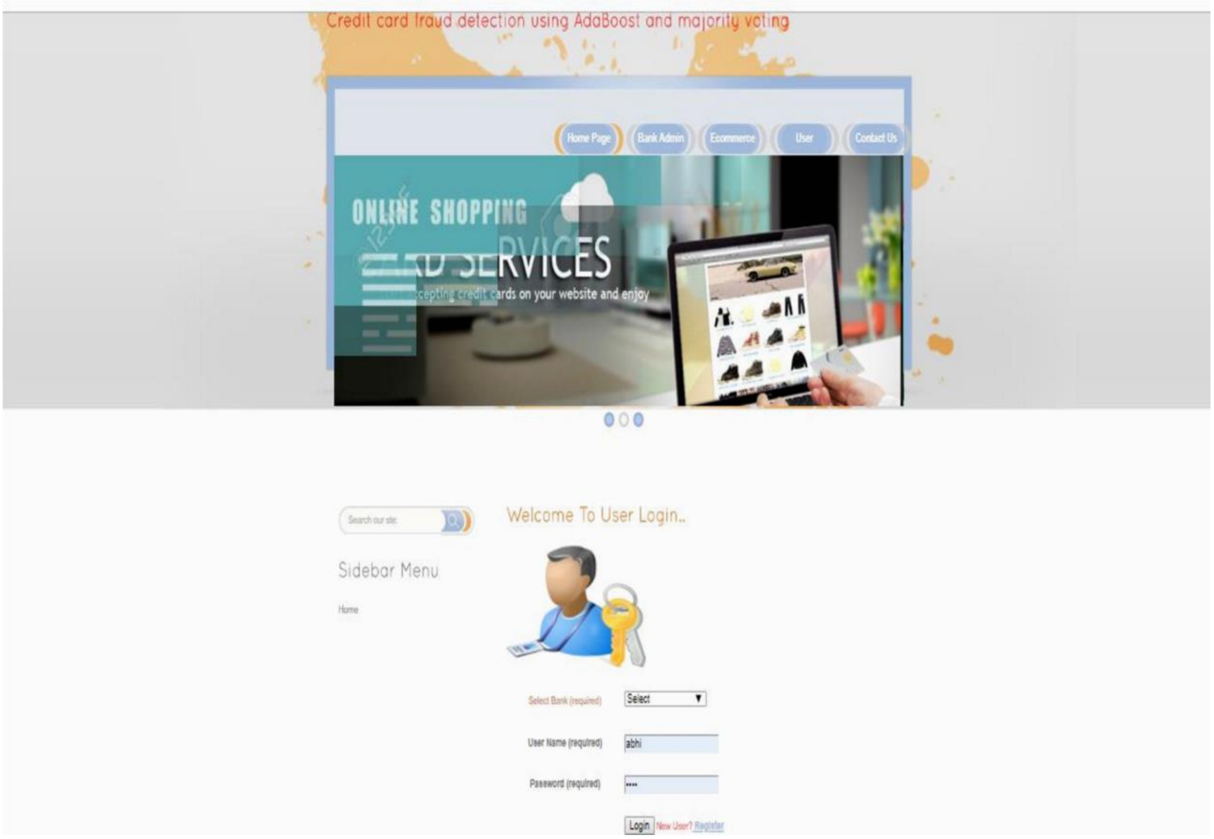
Select Gender (required)

User Registration page Description

Here we can register if we don’t have account. We should enter all the details without fail, or else it will not register.

Screen No: 9

Screen Title: User Login Page



Select Bank (required) Select

User Name (required) abhi

Password (required) ••••

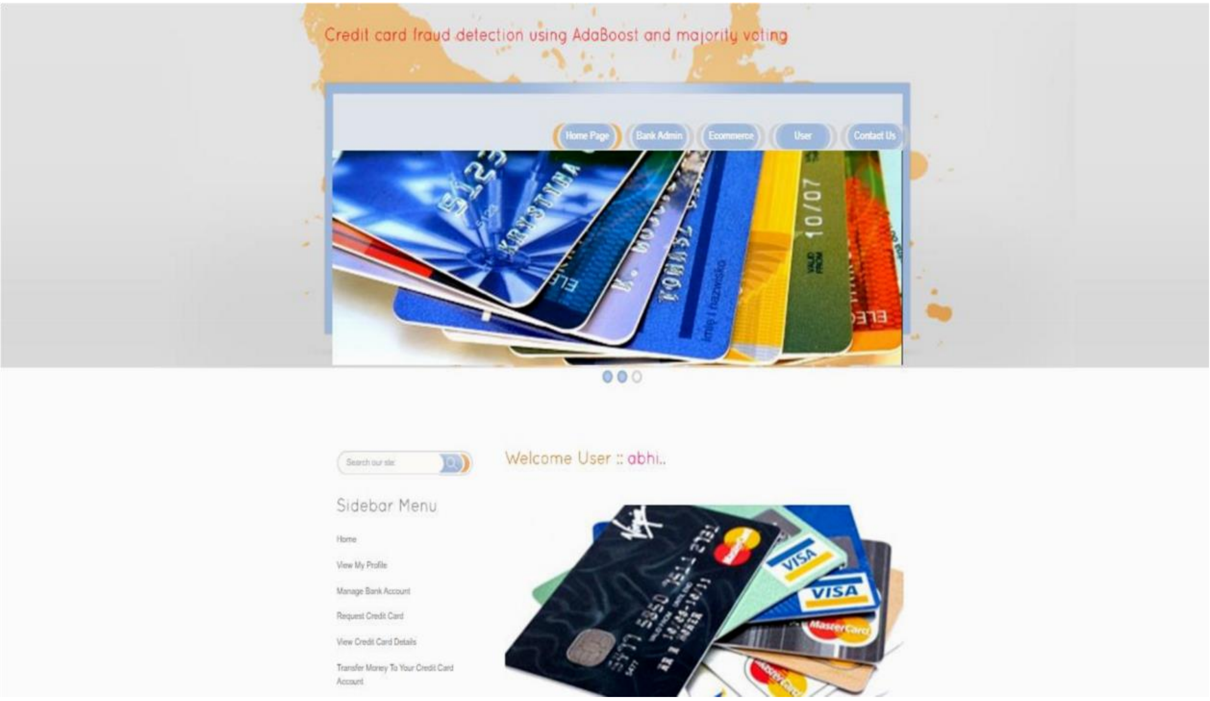
Login New User? Register

User Login Page Description

This is the login page of user. If we want to login to the User page, here we can login if the login credentials are valid .If the login credentials are wrong it will shows a message that the details you have entered are not valid.

Screen No: 10

Screen Title: User Home Page


Credit card fraud detection using AdaBoost and majority voting

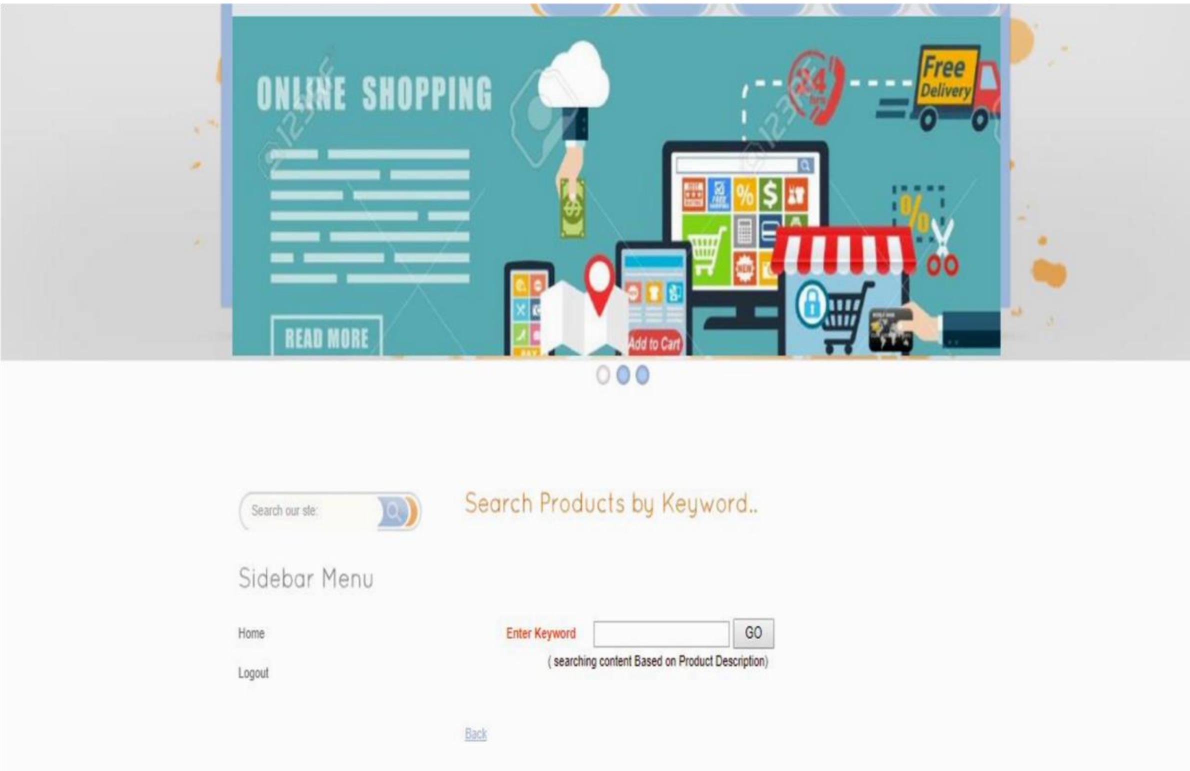
Welcome User :: abhi..

User Home Page Description

User home page contains sub modules like View my profile, Manage Bank Account, View Credit Card Details etc.,

Screen No: 11

Screen Title: Purchase Products



Search Products by Keyword..

Enter Keyword

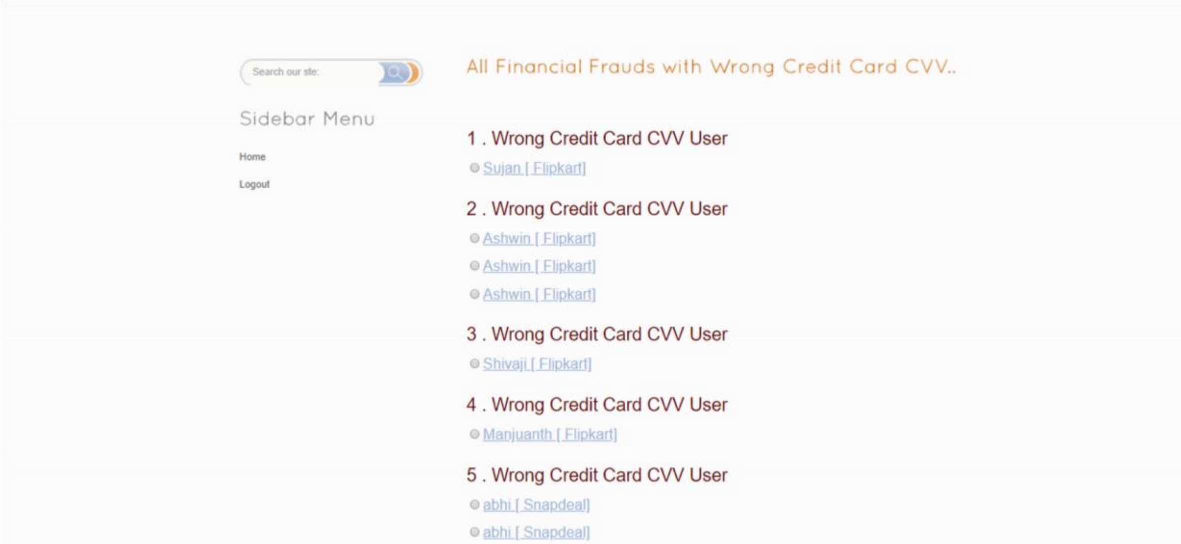
(searching content Based on Product Description)

Purchase Products Description

Here user can purchase products according to their wish,they can search the product by the product Description.

Screen No: 12

Screen Title: Fraud with CVV number



All Financial Frauds with Wrong Credit Card CVV..

1 . Wrong Credit Card CVV User

- Sujan [Flipkart]

2 . Wrong Credit Card CVV User

- Ashwin [Flipkart]
- Ashwin [Flipkart]
- Ashwin [Flipkart]

3 . Wrong Credit Card CVV User

- Shivaji [Flipkart]

4 . Wrong Credit Card CVV User

- Manjuanth [Flipkart]

5 . Wrong Credit Card CVV User

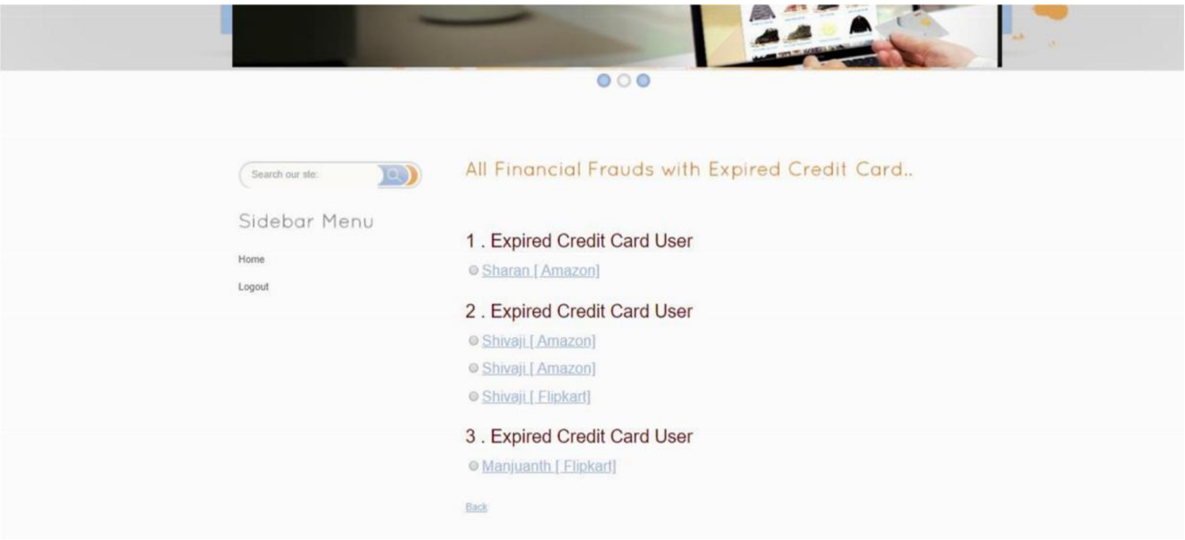
- abhi [Snapdeal]
- abhi [Snapdeal]

Fraud with CVV number Description

These are the frauds which has done when entering wrong CVV number.

Screen No: 13

Screen Title: Fraud with Expired Card number



All Financial Frauds with Expired Credit Card..

1 . Expired Credit Card User

- Sharan [Amazon]

2 . Expired Credit Card User

- Shivaji [Amazon]
- Shivaji [Amazon]
- Shivaji [Flipkart]

3 . Expired Credit Card User

- Manjuanth [Flipkart]

Fraud with Expired Card number Description

These are the frauds which has done with the expired card.

CHAPTER-7

SYSTEM TESTING

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

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

7.1 TYPES OF TESTS

Unit testing

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

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually 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 in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

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

Unit Testing:

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

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.

7.2 TESTING METHODOLOGIES

The following are the Testing Methodologies:

- Unit Testing.

- Integration Testing.
- User Acceptance Testing. ○ Output Testing. ○ Validation Testing.

Unit Testing

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

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

Integration Testing

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

- **Top Down Integration**

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

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

- **Bottom-up Integration**

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

The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure.
- The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

7.2.1 USER ACCEPTANCE TESTING

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

7.2.2 OUTPUT TESTING

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

7.2.3 VALIDATION CHECKING

Validation checks are performed on the following fields.

Text Field:

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

Numeric Field:

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

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

Preparation of Test Data

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

Using Live Test Data:

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

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

Using Artificial Test Data:

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

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

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

7.3 USER TRAINING

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

7.4 MAINTAINENCE

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

The coding and designing is simple and easy to understand which will make maintenance easier.

Testing strategy:

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

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

System testing:

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

Unit testing:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module. In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

7.5 INPUT DESIGN AND OUTPUT DESIGN

7.5.1 INPUT DESIGN

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

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

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

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

7.5.2 OUTPUT DESIGN

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

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

CHAPTER- 8

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

A study on credit card fraud detection using machine learning algorithms has been presented in this paper. A number of standard models which include NB, SVM, and DL have been used in the empirical evaluation. A publicly available credit card data set has been used for evaluation using individual (standard) models and hybrid models using AdaBoost and majority voting combination methods. The MCC metric has been adopted as a performance measure, as it takes into account the true and false positive and negative predicted outcomes. The best MCC score is 0.823, achieved using majority voting. A real credit card data set from a financial institution has also been used for evaluation. The same individual and hybrid models have been employed. A perfect MCC score of 1 has been achieved using AdaBoost and majority voting methods. To further evaluate the hybrid models, noise from 10% to 30% has been added into the data samples. The majority voting method has yielded the best MCC score of 0.942 for 30% noise added to the data set. This shows that the majority voting method is stable in performance in the presence of noise.

Future enhancement

For future work, the methods studied in this project will be extended to online learning models. In addition, other online learning models will be investigated. The use of online learning will enable rapid detection of fraud cases, potentially in real-time. This in turn will help detect and prevent fraudulent transactions before they take place, which will reduce the number of losses incurred every day in the financial sector.

9. BIBLIOGRAPHY

[1] Y. Sahin, S. Bulkan, and E. Duman, “A cost-sensitive decision tree approach for fraud detection,” Expert Systems with Applications, vol. 40, no. 15, pp. 5916–5923, 2013.

[2] A. O. Adewumi and A. A. Akinyelu, “A survey of machine-learning and nature-inspired based credit card fraud detection techniques,” International Journal of System Assurance Engineering and Management, vol. 8, pp. 937–953, 2017.

[3] A. Srivastava, A. Kundu, S. Sural, A. Majumdar, “Credit card fraud detection using hidden Markov model,” IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 1, pp. 37– 48, 2008.

[4] The Nilson Report (October 2016) [Online]. Available:
https://www.nilsonreport.com/upload/content_promo/The_Nilson_R eport_10-17-2016.pdf

[5] J. T. Quah, and M. Sriganesh, “Real-time credit card fraud detection using computational intelligence,” Expert Systems with Applications, vol. 35, no. 4, pp. 1721–1732, 2008.

[6] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C., “Data mining for credit card fraud: A comparative study,” Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[7] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, “Detection of financial statement fraud and feature selection using data mining.

[8] online: https://www.kaggle.com/dalpozz/creditcardfraud

[9] P. Ravisankar, V. Ravi, G. R. Rao, and I. Bose, “Detection of financial statement fraud and feature selection using data mining

[10] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, “Association rules applied to credit card fraud detection,” Expert Systems with Applications, vol. 36, no. 2, pp. 3630–3640, 2009.

[11] E. Duman and M. H. Ozcelik, “Detecting credit card fraud by genetic algorithm and scatter search,” Expert Systems with Applications, vol. 38, no. 10, pp. 13057–13063, 2011.