A

Project Report

On

A Multi-perspective Fraud Detection Method for Multi-Participant E-commerce Transactions

Submitted in partial fulfillment of the requirements for the award of Degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

by

A.MAHALAKSHMI	**(227R5A6706)**
S.SHASHANK	**(217R1A6752)**
P.SAI CHANDU REDDY	**(217R1A6744)**

Under the Guidance of

Dr. M. KISHORE KUMAR

Professor CSE(DATA SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(DATA SCIENCE)

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by
AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the
UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2021-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled **"A MULTI-PERSPECTIVE FRAUD DETECTION METHOD FOR MULTI PARTICIPANT E-COMMERCE TRANSACTIONS"** being submitted by **A.MAHALAKSHMI (227R5A6706), S.SHASHANK (217R1A6752) & P.SAI CHANDU REDDY(217R1A6744)** in partial fulfillment of the requirements for the award of the degree of BTech in **Computer Science and Engineering (Data Science)** to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024- 25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. M. Kishore Kumar Professor CSE(DS) INTERNAL GUIDE	**Dr. K. Murali** HOD CSE(DS)

EXTERNAL EXAMINER

Submitted for viva voice Examination held on____________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr.M.Kishore Kumar,** ProfessorCSE(DS) for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Mr. Sanjib Kumar Nayak, Mr. Y. Ashok Kumar, Ms. N. Soujanya, Mr. Bhookya Ramesh, Mrs. V. Sandya** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Murali,** Head, Department of Computer Science and Engineering (Data Science) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

A .MAHALAKSHMI	**(227R5A6706)**
S.SHASHANK	**(217R1A6752)**
P.SAI CHANDU REDDY	**(217R1A6744)**

ABSTRACT

Detection and prevention of fraudulent transactions in e-commerce platforms have always been the focus of transaction security systems. However, due to the concealment of e-commerce, it is not easy to capture attackers solely based on the historic order information. Many researches try to develop technologies to prevent the frauds, which have not considered the dynamic behaviors of users from multiple perspectives. This leads to an inefficient detection of fraudulent behaviors. To this end, this paper proposes a novel fraud detection method that integrates machine-learning and process mining models to monitor real-time user behaviors. First, we establish a process model concerning the B2C e-commerce platform, by incorporating the detection of user behaviors. Second, a method for analyzing abnormalities that can extract important features from event logs is presented. Then, we feed the extracted features to a Support Vector Machine (SVM) based classification model that can detect fraud behaviors. We demonstrate the effectiveness of our method in capturing dynamic fraudulent behaviors in e-commerce systems through the experiments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SCREENSHOTS

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

The project “A Multi-Perspective Fraud Detection Method for Multi-Participant E-Commerce Transactions” focuses on developing a Multi-Perspective Fraud Detection Method (MPFDM) to enhance security in e-commerce transactions involving multiple participants. Fraud detection in such environments is challenging due to the anonymity of users and the complexity of interactions between buyers, sellers, and intermediaries. This project aims to address these challenges by integrating machine learning, process mining, and behavioral analytics to detect fraudulent activities in real time.

The project will involve several key components. First, data collection and preprocessing will be conducted using e-commerce transaction records, including order history, user interactions, and event logs. These datasets will be cleaned and processed to extract meaningful features that indicate suspicious activities. Next, a process mining model will be developed to analyze user behavior and transaction flows, helping to establish normal patterns and detect deviations indicative of fraud. Additionally, an anomaly analysis framework will be implemented to extract critical features from event logs, which will then be used in a Support Vector Machine (SVM)-based classification model for fraud detection.

To validate the effectiveness of the proposed method, the model will be tested using real-world e-commerce datasets. Performance will be evaluated using key metrics such as accuracy, precision, recall, and F1-score, with comparisons made against traditional fraud detection techniques. The expected outcome is a robust and scalable fraud detection framework capable of identifying evolving fraud patterns in multi-participant transactions.

Despite its advantages, the system may require fine-tuning for different e-commerce platforms and datasets. Future improvements could include integrating deep learning models for enhanced detection capabilities and real-time fraud prevention mechanisms to further strengthen security in online transactions.

1.2 PROJECT PURPOSE

The purpose of the project “A Multi-Perspective Fraud Detection Method for Multi-Participant E-Commerce Transactions ” is to develop an advanced Multi-Perspective Fraud Detection Method (MPFDM) to enhance the security of e-commerce transactions involving multiple participants, such as buyers, sellers, and intermediaries. Traditional fraud detection systems primarily focus on individual user behaviors and historical transaction data, making them ineffective in identifying complex fraud schemes that involve multiple entities. This project aims to address these limitations by integrating machine learning, process mining, and behavioral analytics to detect fraudulent activities in real-time.

The proposed method seeks to improve fraud detection by analyzing user interactions from multiple perspectives, rather than relying solely on static transaction data. By leveraging process mining models, the system can monitor and track user behaviors dynamically, identifying unusual patterns that may indicate fraud. Additionally, event log analysis will be used to extract key features from transaction histories, which will then be fed into a Support Vector Machine (SVM)-based classification model to detect anomalies more accurately.

The ultimate goal of this project is to provide a scalable, efficient, and adaptable fraud detection framework for modern e-commerce platforms. By enhancing fraud detection accuracy, the system will help reduce financial losses, improve transaction security, and build trust among e-commerce participants. Furthermore, the project aims to contribute to the ongoing development of intelligent fraud detection systems that can adapt to emerging threats and fraudulent tactics in the digital marketplace.

1.3 PROJECT FEATURES

The project “A Multi-Perspective Fraud Detection Method for Multi-Participant E-Commerce Transactions” introduces several key features to enhance fraud detection in multi-participant e-commerce transactions. Unlike traditional methods that focus on individual transactions, MPFDM adopts a multi-perspective analysis, examining interactions among buyers, sellers, and intermediaries to identify collusion and fraudulent behavior.

Additionally, MPFDM integrates machine learning-based fraud detection, utilizing a Support Vector Machine (SVM) to classify transactions as fraudulent or legitimate based on extracted behavioral and transactional features. The system further enhances fraud detection through event log analysis, extracting key indicators such as transaction frequency, payment patterns, and order anomalies to identify suspicious activities. A graph-based relationship mapping feature is also included, allowing the visualization of user connections and the detection of collusive fraud networks.

To ensure adaptability, the system includes adaptive learning capabilities, allowing it to continuously improve by updating fraud detection models based on evolving fraudulent techniques. Furthermore, MPFDM is scalable and integrable, making it suitable for large-scale e-commerce platforms while allowing seamless integration with existing fraud detection frameworks. Finally, the model's effectiveness is rigorously tested using real-world e-commerce datasets, with performance measured through accuracy, precision, recall, and F1-score to optimize detection efficiency. By incorporating these advanced features, MPFDM offers a comprehensive, intelligent, and scalable fraud detection solution for securing multi-participant e-commerce transactions.

2. SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.1 PROBLEM DEFINITION

Fraud in e-commerce transactions is a significant challenge, particularly in multi-participant environments where buyers, sellers, and intermediaries interact. Traditional fraud detection systems primarily analyze individual behaviors and historical data, making them ineffective in identifying sophisticated fraud schemes such as collusion, fake orders, and account takeovers. Existing fraud detection models, which are often rule-based or statistical, struggle to adapt to emerging fraudulent techniques, leading to high false positives and false negatives. These inefficiencies result in financial losses, reduced platform trust, and negative customer experiences. To overcome these limitations, a multi-perspective approach is required to analyze interactions across multiple participants and detect fraud in real-time. By integrating machine learning, process mining, and behavioral analytics, fraudulent activities can be identified more accurately. This project proposes a Multi-Perspective Fraud Detection Method (MPFDM) to monitor user behavior, extract key transaction features, and detect anomalies dynamically.

The core problem lies in the lack of an integrated, comprehensive approach that can analyze fraud signals across various perspectives, including user activity, transactional behavior, device usage, behavioral biometrics, and network relationships. Therefore, there is a pressing need to develop a multi-perspective fraud detection method that holistically combines these dimensions to enhance detection capabilities. Such a system must be scalable, adaptive to new fraud patterns, and capable of real-time decision-making to effectively mitigate fraud in modern, multi-channel e-commerce environments.

Fraudsters exploit this gap by spreading their actions across multiple accounts, devices, and channels, making it difficult to trace and flag malicious behavior using siloed approaches.

2.2 EXISTING SYSTEM

Current fraud detection systems in e-commerce primarily rely on rule-based methods, statistical analysis, and machine learning models that focus on individual transaction patterns. These systems typically analyze historical transaction data, user profiles, and predefined fraud indicators to detect suspicious activities. While effective for basic fraud detection, they struggle to identify complex fraud schemes involving multiple participants, such as collusive fraud, fake reviews, and coordinated attacks.

Most existing methods operate on a single-perspective approach, meaning they assess fraud risk based on isolated user behaviors rather than considering interactions between buyers, sellers, and intermediaries. Additionally, traditional rule-based systems require manual updates and predefined thresholds, making them rigid and ineffective against evolving fraud tactics. Machine learning models, while more adaptive, often lack real-time monitoring capabilities and fail to capture the dynamic nature of user interactions.

Another major limitation is the reliance on static datasets, which only provide insights into past fraudulent activities. This prevents early detection of emerging fraud patterns, leading to high false positives (legitimate users being flagged) and high false negatives (fraudsters evading detection). Furthermore, existing systems do not leverage process mining techniques to analyze transaction workflows, making it difficult to identify deviations from normal behavior.

Due to these shortcomings, e-commerce platforms face challenges in accurately detecting and preventing fraud in multi-participant transactions. To overcome these limitations, a more intelligent, multi-perspective, and real-time fraud detection system is needed to enhance security and trust in digital marketplaces.

2.2.1 LIMITATIONS OF EXISTING SYSTEMS

Following are the disadvantages of existing system:

Fraud mode one - an order is tempered by a malicious actor: The malicious actor may deceive the victim merchant by sending a fake formal payment order to the cash server.

Fraud mode two - subcontract the order: The victim pays the malicious actor’s order instead of his order. To achieve their goals, the malicious actors impersonate the duties of sellers and buyers.

2.3 PROPOSED SYSTEM

The proposed system combines the advantages of process mining and machine learning models by introducing a hybrid method to solve the anomaly detection in data flows, which provides information about each action embedded in a control flow model. By modeling and analyzing the business process of the e-commerce system, this method can dynamically detect changes in user behaviors, transaction processes, and noncompliance situations, and comprehensively analyze and identify fraudulent transactions from multiple perspectives. Important contributions of this paper are listed as follows:

1. A conformance checking method based on process mining is applied in the field of e-commerce transactions to capture the abnormalities.
2. A user behavior detection method is proposed to perform comprehensive anomaly detection based on Petri nets.
3. An SVM model is developed by embedding a multi perspective process mining into machine learning methods to automatically classify fraudulent behaviors.

The proposed system refers to a newly suggested method, process, or solution that is designed to overcome the limitations of the existing system. It outlines how the new system will function, highlighting the improvements it brings in terms of efficiency, accuracy, user experience, or automation. Typically, the proposed system is developed after analyzing the drawbacks of the current setup and aims to provide a more effective and reliable alternative. It may include the use of modern technologies, improved workflows, or enhanced features to address the identified issues and fulfill user requirements more efficiently.

The proposed system refers to a newly designed solution that aims to replace or enhance the existing system by addressing its limitations and improving overall functionality. It is developed after thorough analysis of the current system, identifying issues such as inefficiency, inaccuracy, lack of automation, or poor user experience. The proposed system offers a more advanced, efficient, and user-friendly approach, often incorporating the latest technologies and streamlined processes to deliver better performance. It defines how tasks will be handled, how data will flow, and what new features or tools will be introduced to improve operations.

2.3.1 PROPOSED APPROACH

The proposed Multi-Perspective Fraud Detection Method (MPFDM) enhances fraud detection in multi-participant e-commerce transactions by integrating machine learning, process mining, anomaly detection, and graph-based analysis. Unlike traditional systems that rely only on historical data, this approach continuously monitors real-time user behaviors, transaction sequences, and inter-user relationships to detect fraudulent activities more effectively. The method consists of three key stages: behavioral process modeling, where process mining techniques like Alpha Miner and Heuristic Miner analyze normal transaction flows and identify deviations; feature extraction and anomaly detection, where important transaction attributes such as frequency, geolocation mismatches, and timing patterns are analyzed using unsupervised learning models like K-Means Clustering, Isolation Forest, and Autoencoders to detect suspicious activities; and machine learning and graph-based classification, where supervised models such as Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR) classify transactions, while Graph Neural Networks (GNNs) analyze user relationships to detect collusive fraud and coordinated attacks. By integrating these methods, the proposed approach ensures a multi-perspective, adaptive, and scalable fraud detection system that improves real-time fraud prevention, reduces financial losses, and enhances e-commerce security.

2.3.2 ADVANTAGES OF PROPOSED SYSTEM

- To arrive at a clearer result, the plug-in Multi-Perspective Process Explorer and Conformance Checking are used to match and analyze the event log and the DPN. The result is shown in this system, where each action is represented with different colors. For instance, green represents the move both on model and log, purple means move on the model only, and grey represents invisible actions, that is, skipped actions.
- By clicking on a given action, we can obtain the matching information between the model and the event log in the data flow of each action. The data marked in red indicates a mismatch. We extract these suspicious anomalies and use them as the basis for subsequent training using machine learning models.

2.4 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

2.4.1 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.

2.4.2 Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the 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.

2.4.3 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.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

Hardware requirements refer to the minimum and recommended physical computing components—such as processing units, memory, storage, and networking equipment—needed to develop, train, deploy, and operate the multi-perspective fraud detection system efficiently. These requirements are essential to ensure the system can process large volumes of multi-dimensional data in real time, support advanced machine learning or deep learning algorithms, and maintain reliable performance at scale.

▪ PROCESSOR	: Pentium –IV
▪ RAM	: 4GB (min)
▪ HARD DISK	: 20 GB
▪ KEYBOARD	: Standard Windows Keyboard
▪ MOUSE	: Two or Three Button Mouse
▪ MONITOR	: SVGA

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

▪ OPERATING SYSTEM	: Windows 7 Ultimate.
▪ CODE LANGUAGE	: Python
▪ FRONT-END	: Python
▪ BACK-END	: Django-ORM
▪ DESIGNING	: HTML, CSS, JavaScript
▪ DATABASE	: MySQL
▪ WEB SERVER	: WAMP Server

2.6 PROGRAMMING LANGUAGES USED

2.6.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python can also be used for frontend tasks using frameworks like Flask, Django, and Dash. One of the main advantages of using Python for frontend development is its ease of use and readability. Python code is typically more concise and easier to understand compared to languages like JavaScript, which can lead to faster development and easier maintenance of frontend code. Additionally, Python's extensive standard library and large ecosystem of third-party packages provide a wide range of tools and libraries for frontend development, including data visualization, web scraping, and interactive user interfaces. Python's versatility is another key advantage for frontend development. It can be used for a wide range of tasks, from creating simple web applications to building complex data-driven applications.

Python's popularity in data science and machine learning also makes it a valuable tool for building interactive data visualizations and dashboards in web applications. Python is a versatile and widely-used programming language that is well-suited for data analysis, machine learning, and artificial intelligence applications, making it an ideal choice for our stock price prediction project. Python's simplicity and readability make it easy to write and maintain code, while its extensive libraries and frameworks provide powerful tools for data manipulation, visualization, and modeling.

Python supports multiple programming paradigms, including procedural, object oriented, and functional programming, allowing developers to choose the approach that best suits their needs. Its dynamic typing and automatic memory management contribute to rapid development and prototyping, while its interpreted nature enables cross-platform compatibility and easy integration with other languages and systems.

Python's versatility is evident in its adoption by tech giants like Google, Facebook, and Instagram, as well as leading scientific institutions like NASA and CERN. Its role in powering cutting-edge technologies like machine learning, artificial intelligence, and big data analytics has propelled its popularity and influence across industries. Python's simplicity and expressiveness make it an ideal choice for rapid prototyping and experimentation, enabling developers to iterate quickly and adapt to changing requirements. Additionally, Python's cross-platform compatibility ensures that code written in Python can run seamlessly on various operating systems, including Windows, macOS, and Linux.

The Python community, known for its inclusivity, collaboration, and open-source ethos, plays a crucial role in the language's success. Pythonistas, as they are affectionately called, actively contribute to the language's development, documentation, and ecosystem. Python's governance model, overseen by the Python Software Foundation (PSF), ensures transparency, accountability, and community-driven decision-making. Regular conferences, meetups, and workshops worldwide foster knowledge-sharing, networking, and mentorship opportunities, enriching the Python community and empowering individuals to grow and succeed in their Python journey.

In our project, Python is used for both frontend development. For the frontend, we can use libraries like Flask or Django to create a web application interface for users to interact with the stock price prediction model. These frameworks provide features for routing, form handling, and template rendering, making it easy to create a user-friendly interface. For the coding, Python is used to implement the SVM model and handle data processing tasks. Libraries like NumPy and pandas are used for data manipulation and preprocessing, while scikit-learn and TensorFlow are used for building and training the model.

Overall, Python's flexibility, ease of use, and powerful libraries make it an excellent choice for developing the stock price prediction project. Its ability to handle data analysis, machine learning, and web development tasks makes it a valuable tool for building a comprehensive and effective prediction model.

Creating software prototypes: When compared to compiled languages like C++ and Java, Python is sluggish. If resources are scarce and performance is needed, it can not be the best solution. Python, on the other hand, is an excellent language for prototyping. Consider the following scenario: You can start by making a demo for your game using Pygame (a game creation library). If you like the demo, you can make the game using a language like C++.

Python Programming Characteristics

- It has a larger number of data types and a simpler syntax than any other programming language.
- It's a scripting language that works on every platform and has direct access to operating system APIs.
- It has more run-time stability than other programming languages.
- It contains Perl and Awk's simple text manipulation features.
- In Python, a module can contain one or more classes and free functions.
- Python libraries are cross-platform, meaning they work on Linux, Macintosh, and Windows.
- Python can be converted to bytecode for use in massive applications.
- Python embraces both functional and formal programming, as well as object oriented programming (OOP).
- It has an immersive feature that helps you to communicate with it.
- Software fragments are tested and debugged.
- Since there is no compilation phase in Python, writing, debugging, and checking are all possible.

2.6.2 SQL

SQL (Structured Query Language) is a standard language for interacting with relational databases like MySQL, and it is used in the backend of the stock price prediction project to manage and manipulate data. SQL provides a powerful set of commands for querying, updating, and managing databases, making it an essential tool for working with large datasets and complex data structures.

SQL's compatibility with other technologies and programming languages, such as Python, allows for seamless integration with the frontend and backend components of your project. This enables efficient data transfer and communication between the application and the database, facilitating the development of a robust and reliable stock price prediction system.

SQL operates on the principles of set theory and relational algebra, treating data as sets of rows and columns and providing operators to perform operations like selection, projection, join, and aggregation. This set-based approach allows for efficient manipulation of large datasets and enables users to express complex queries concisely and logically. SQL's standardized syntax and comprehensive feature set are supported by most modern relational database management systems (RDBMS), including MySQL, PostgreSQL, Oracle, Microsoft SQL Server, and SQLite, ensuring portability and interoperability across different platforms and environments.

In the project, SQL is used to create and manage database tables for storing historical trading data, sentiment indicators, and other relevant information. SQL queries can then be used to retrieve data from these tables, filter and aggregate data, and perform calculations needed for training and testing the model.

SQL supports a wide range of features and capabilities, including joins to combine data from multiple tables, subqueries to perform nested queries, aggregate functions to calculate summary statistics, and window functions to perform calculations over partitions of data. It also provides support for data manipulation tasks like sorting, filtering, grouping, and paging, as well as advanced features like stored procedures, triggers, and user-defined functions for procedural logic and automation.

One of the key advantages of using SQL is its simplicity and readability. SQL queries are easy to write and understand, even for users with limited programming experience. This makes it easier to manage and maintain the database schema and perform complex data operations. SQL's support for transactions and data integrity ensures that the data stored in the database is consistent and reliable. This is crucial for financial applications like stock price prediction, where accurate and reliable data is essential for making informed investment decisions.Advanced functionalities like joins, aggregate functions, and transactions enhance data retrieval and manipulation processes while ensuring data integrity and consistency through features like constraints and transactions.

2.6.3 HTML

HTML stands for HyperText Markup Language. It is the standard markup language used to create web pages. HTML is a combination of Hypertext and Markup language. 11 Hypertext defines the link between web pages. A markup language is used to define the text document within the tag to define the structure of web pages.

Hypertext defines the link between web pages. A markup language is used to define the text document within the tag to define the structure of web pages. HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1995.

This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.It consists of a series of elements that define the structure and content of a webpage, including headings, paragraphs, images, links, and multimedia elements.

HTML elements, developers can use custom attributes and data attributes to add additional functionality and interactivity to their web pages. HTML determines how information is organized and presented on a website, laying the groundwork for styling with CSS and interactivity with JavaScript. When deciding on HTML, factors like accessibility, semantic markup, browser compatibility, and SEO considerations should be taken into account to ensure a well-structured and user-friendly website. Additionally, staying up-to-date with HTML standards and best practices is crucial for creating modern, responsive, and accessible web experiences.

HTML documents are hierarchical in nature, with elements nested within other elements to create a structured document tree.HTML5, the latest version of HTML, introduced several new features and APIs for creating rich, interactive web applications, including support for native multimedia playback, offline web applications, geolocation, and drag-and-drop functionality.

HTML documents are structured with tags like <head>,<title> and <body> while semantic elements like ,<heaaders>,<footer>,<nav> and <article> enhance accessibility and search engine optimization. and enhance accessibility and search engine optimization.

HTML stands for HyperText Markup Language and it is used to create webpages. It uses HTML tags and attributes to describe the structure and formatting of a web page.HTML consists of various elements, that are responsible for telling search engines how to display page content. For example, headings, lists, images, links, and more.

Features of HTML

- It is easy to learn and easy to use.
- It is platform-independent.
- Images, videos, and audio can be added to a web page.
- Hypertext can be added to the text.
- It is a markup language.

2.6.4 CSS

CSS (Cascading Style Sheets) is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to HTML documents. It describes how a webpage should look. Using CSS enables consistent styling across a website and facilitates easier maintenance and updates. It's essential to carefully consider factors like design requirements, browser compatibility, performance optimization, and maintainability when working with CSS. Additionally, adhering to best practices and staying updated with evolving standards can ensure efficient and effective use of CSS in web development projects.It prescribes colors, fonts, spacing, etc. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

HTML uses tags and CSS uses rulesets. CSS styles are applied to the HTML element using selectors. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

CSS selectors can target HTML elements based on their type, class, ID, attributes, and relationship to other elements in the document tree. This allows developers to apply styles selectively to specific elements or groups of elements, providing fine-grained control over the appearance of a web page. CSS properties define visual aspects such as width, height, margin, padding, border, background, font, color, and more. These properties can be set using various units of measurement, including pixels, percentages, ems, rems, and viewport units, enabling flexible and responsive design.

CSS has evolved significantly since its inception, with successive versions introducing new features, selectors, and properties to meet the evolving needs of web developers. CSS3, the latest version of CSS, introduced advanced features such as media queries for responsive design, flexbox and grid layouts for flexible page structures, transitions and animations for adding dynamic effects, and custom properties for reusable style variables.

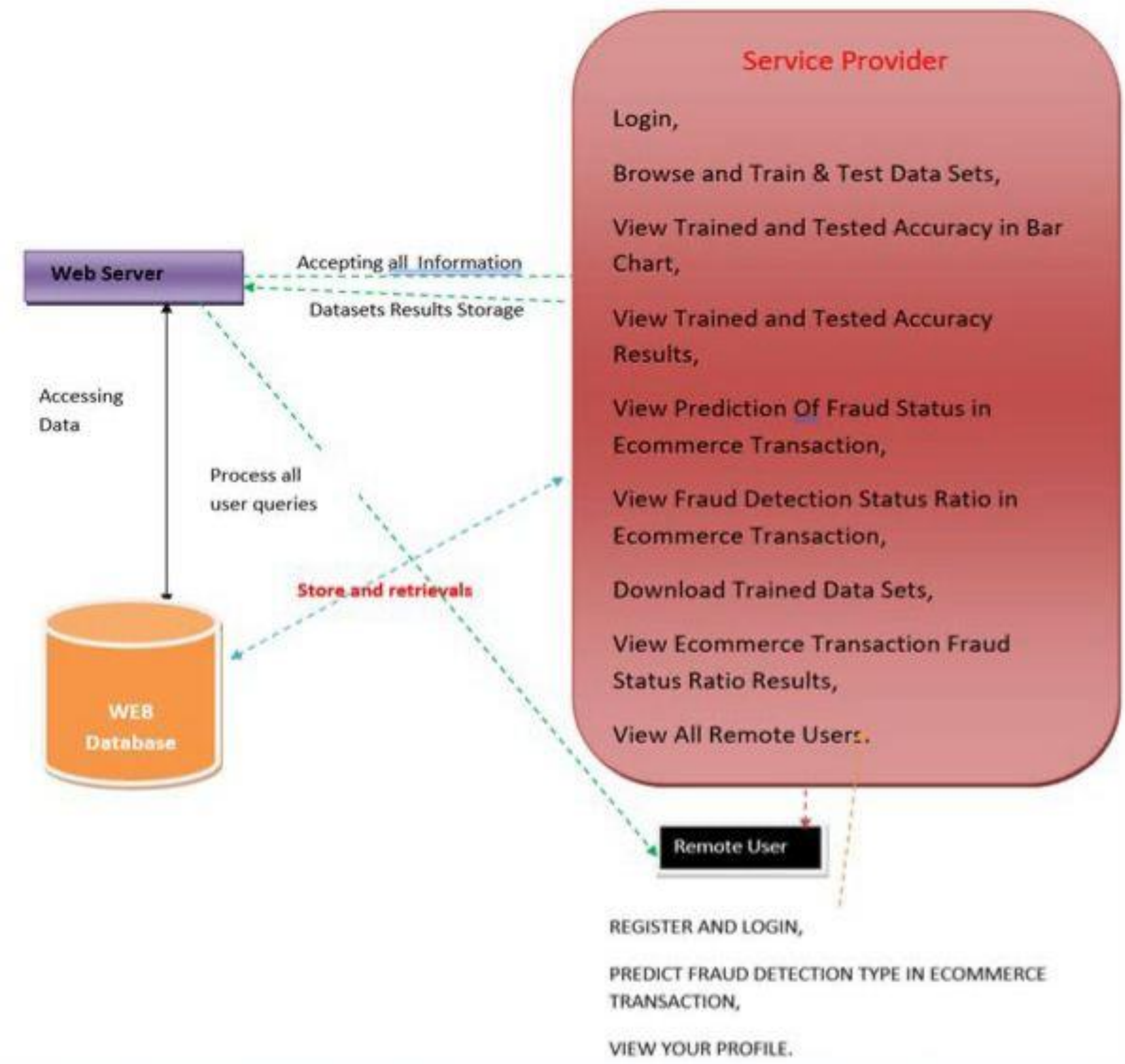
CSS is mainly used for styling up sheets with different colors to make a web page attractive for everyone with the help of html and javascript the webpage looks more attractive. CSS3 introduces advanced features like transitions, animations, and responsive design techniques, enabling developers to create immersive and adaptable web experiences across different devices and screen sizes.

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.


Service Provider
Login,
Browse and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Fraud Status in Ecommerce Transaction,
View Fraud Detection Status Ratio in Ecommerce Transaction,
Download Trained Data Sets,
View Ecommerce Transaction Fraud Status Ratio Results,
View All Remote Users.
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User
REGISTER AND LOGIN,
PREDICT FRAUD DETECTION TYPE IN ECOMMERCE TRANSACTION,
VIEW YOUR PROFILE.

3.1: Project Architecture of Multi Perspective Fraud Detection Method for Multi Participant E-commerce Transaction

DESCRIPTION

The architecture of an eCommerce fraud detection system, illustrating the interaction between the Web Server, Web Database, Service Provider, and Remote User. The Web Server acts as a bridge between users and the Web Database, processing all queries, storing dataset results, and managing data access. The Web. The Service Provider has administrative control, allowing them to log in, browse, train, and test datasets while analyzing fraud detection accuracy through bar charts and results. They can also predict fraud status, view fraud detection ratios, download trained datasets, and monitor remote users. On the other hand, the Remote User can register, log in, predict fraud detection in eCommerce transactions, and view their profile. The data flow within the system ensures smooth interaction, with the Web Server facilitating communication between users and the Web Database, ensuring fraud detection and security enhancement. This system effectively leverages machine learning models to analyze transactions, detect fraud, and provide insights for better eCommerce security.

Web Server

- Acts as the intermediary between the users and the database.
- Accepts all information, processes user queries, and manages dataset results storage.

Web Database

- Stores and retrieves data related to transactions, trained models, fraud predictions, and user information.
- Supports data access requests from the web server.

Service Provider

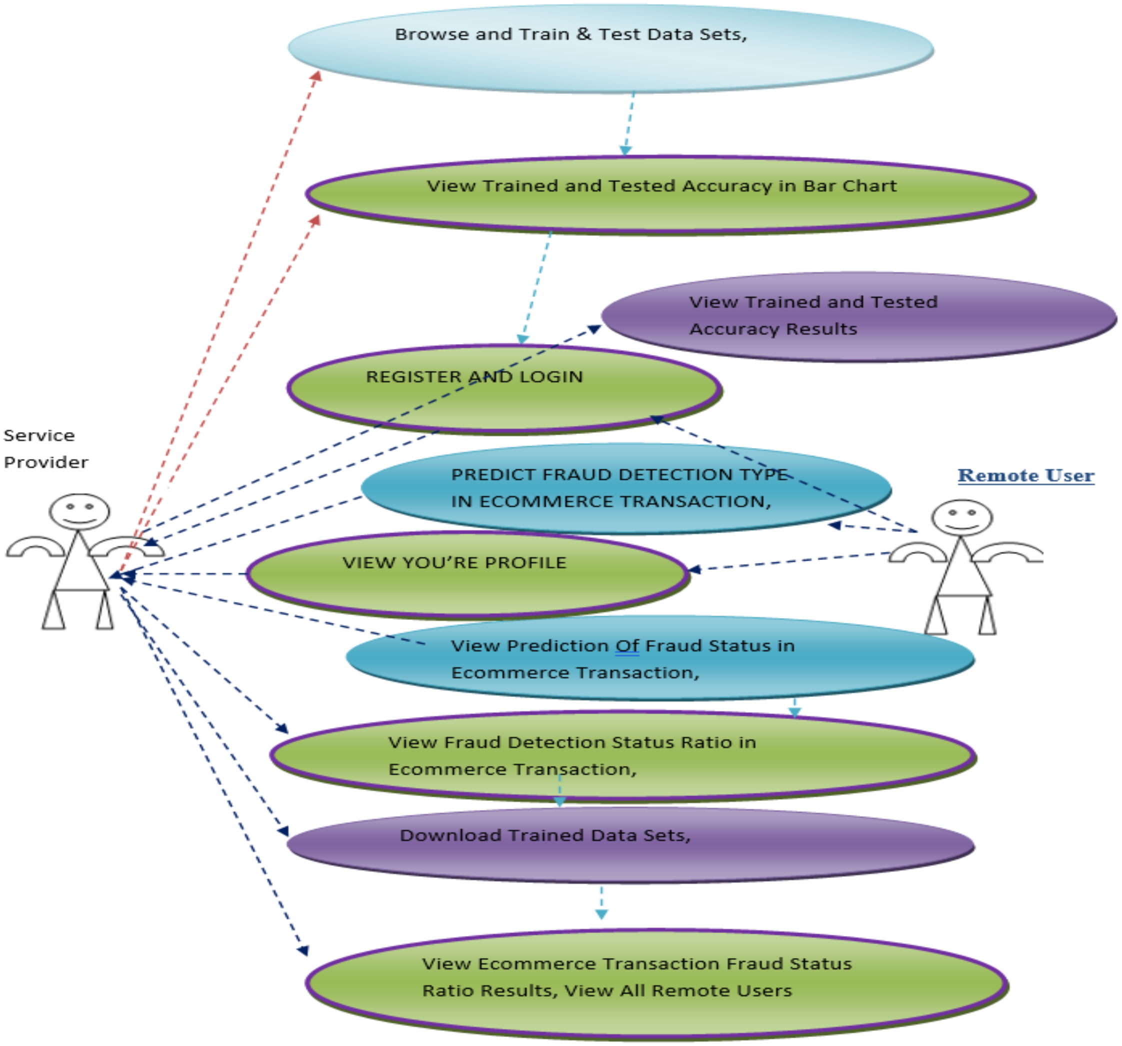
- Has administrative privileges to log in, browse, train, and test datasets.
- Can view trained model accuracy in bar charts and review fraud detection results.
- Can predict fraud status in eCommerce transactions and download trained datasets.

Remote User

- Can register and log in to the system.
- Has access to fraud prediction services for eCommerce transactions.

3.2 USE CASE DIAGRAM

In the use case diagram, we have basically two actors who is the user in the trained model. A use case diagram is a graphical depiction of a user's possible interactions with a system. It shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.



Browse and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
REGISTER AND LOGIN
Service Provider
PREDICT FRAUD DETECTION TYPE IN ECOMMERCE TRANSACTION,
Remote User
VIEW YOU'RE PROFILE
View Prediction Of Fraud Status in Ecommerce Transaction,
View Fraud Detection Status Ratio in Ecommerce Transaction,
Download Trained Data Sets,
View Ecommerce Transaction Fraud Status Ratio Results, View All Remote Users

3.2: Use Case Diagram for Multi-Perspective Fraud Detection.

DESCRIPTION

A use case diagram is a visual representation in Unified Modeling Language (UML) that illustrates the interactions between users (actors) and a system to achieve specific goals (use cases). It captures the functional requirements of a system, showing what the system does from the user's perspective. In a use case diagram, actors are represented as stick figures, and use cases are depicted as ovals. Lines connect actors to the use cases they interact with, showing the relationships and flow of functionality. The diagram helps stakeholders understand the system's expected behavior, supports requirement gathering, and aids in system design and testing. It is commonly used in software engineering during the early stages of development to ensure all user needs are addressed.

Service Provider

1. The Service Provider is responsible for managing and analyzing fraud detection data.

2. They can browse, train, and test datasets for fraud detection purposes.

3. They can view the trained and tested accuracy in both bar chart form and detailed result form.

4. They can also view fraud detection status ratios, predict fraud status in e-commerce transactions, and download trained datasets.

5. Additionally, the Service Provider can monitor the Ecommerce Transaction Fraud Status Ratio Results and view all remote users.

Remote User

1. Register and Login: Remote Users can create an account and log in to the system.

2. View Your Profile: They can view and manage their personal profile.

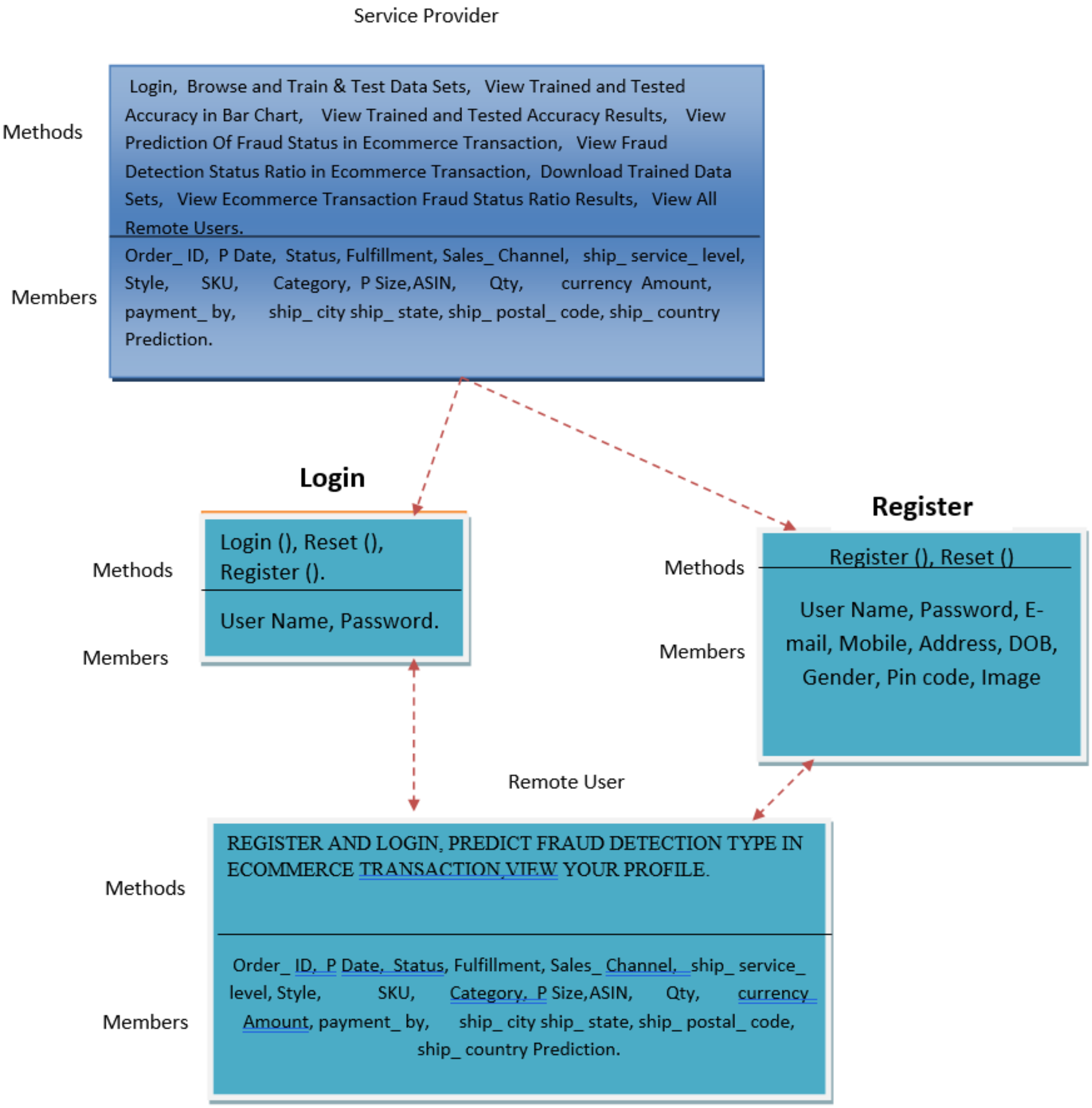
3. They can use the fraud detection system to predict fraud detection types in eCommerce transactions.

4. They also have access to view their profile.

5. View All Remote Users: Remote Users can see a list of all other users in the system.

3.3 CLASS DIAGRAM

Class diagram 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 objects.



Service Provider
Methods
Login, Browse and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, View Prediction Of Fraud Status in Ecommerce Transaction, View Fraud Detection Status Ratio in Ecommerce Transaction, Download Trained Data Sets, View Ecommerce Transaction Fraud Status Ratio Results, View All Remote Users.
Members
Order_ ID, P Date, Status, Fulfillment, Sales_ Channel, ship_ service_ level, Style, SKU, Category, P Size,ASIN, Qty, currency Amount, payment_ by, ship_ city ship_ state, ship_ postal_ code, ship_ country Prediction.
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
Remote User
Methods
REGISTER AND LOGIN, PREDICT FRAUD DETECTION TYPE IN ECOMMERCE TRANSACTION,VIEW YOUR PROFILE.
Members
Order_ ID, P Date, Status, Fulfillment, Sales_ Channel, ship_ service_ level, Style, SKU, Category, P Size,ASIN, Qty, currency Amount, payment_ by, ship_ city ship_ state, ship_ postal_ code, ship_ country Prediction.

3.3: Class Diagram for Multi-Perscpective Fraud Detection Method

DESCRIPTION

The class diagram illustrates the structure of a system by showing its classes, attributes, and the relationships between them. Here’s a detailed description:

The relationships and interactions between the Service Provider, Remote User, and authentication modules (Login and Register) within an eCommerce fraud detection system. The Service Provider is responsible for managing system functionalities, including browsing, training, and testing datasets, viewing fraud prediction results, and monitoring remote users. It maintains a database containing transaction details such as Order ID, product details, shipping information, and payment methods.

For a user to access the system, they must first go through the Register module, where they provide credentials such as username, password, email, mobile number, address, date of birth, gender, and an image. Registered users can then log in via the Login module, where they authenticate using a username and password. Both modules also offer a Reset function for password recovery.

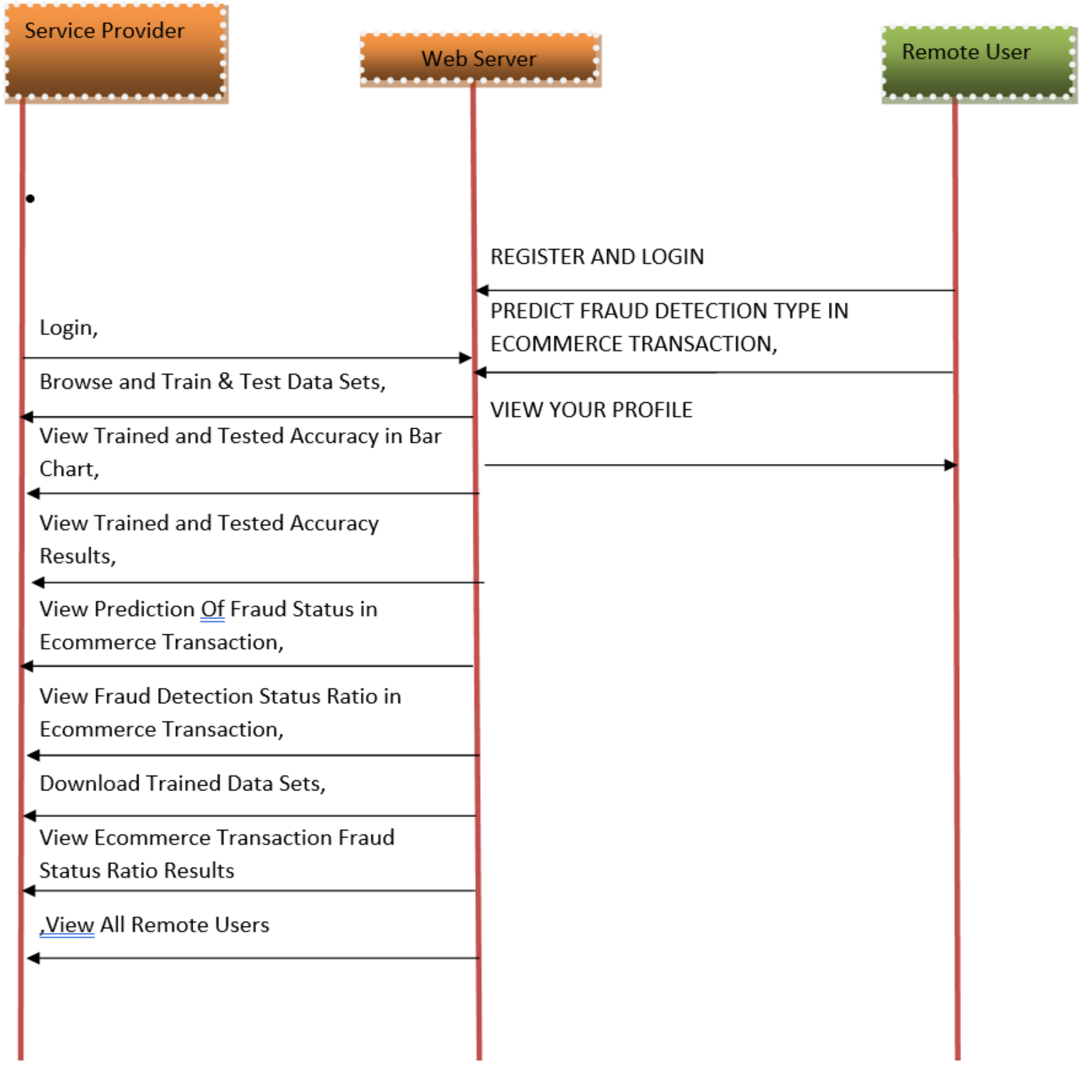
Once logged in, the Remote User can access system features such as predicting fraud detection in eCommerce transactions, viewing fraud status results, and checking their profile. The Service Provider continuously interacts with user data, allowing fraud detection predictions based on transaction details. Users submit transactions, and the system analyzes attributes such as order status, fulfillment details, sales channels, and currency amount to determine fraud likelihood.

The relationships between components ensure a structured process where the Service Provider acts as the central controller, while Remote Users interact through authentication and fraud detection modules.

A class diagram is a type of static structure diagram used in object-oriented modeling to visually represent the classes within a system and the relationships between them. It provides a high-level overview of the system's structure by showing the system’s classes, their attributes, methods (also known as operations), and how they interact with one another. Each class is typically displayed as a box divided into three sections: the class name at the top, attributes in the middle, and methods at the bottom.

3.4 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.


Service Provider
Web Server
Remote User
REGISTER AND LOGIN
PREDICT FRAUD DETECTION TYPE IN ECOMMERCE TRANSACTION,
VIEW YOUR PROFILE
Login,
Browse and Train & Test Data Sets,
View Trained and Tested Accuracy in Bar Chart,
View Trained and Tested Accuracy Results,
View Prediction Of Fraud Status in Ecommerce Transaction,
View Fraud Detection Status Ratio in Ecommerce Transaction,
Download Trained Data Sets,
View Ecommerce Transaction Fraud Status Ratio Results
,View All Remote Users

3.4: Sequence Diagram for Multi-Perscpective Fraud Detection Method

DESCRIPTION

The sequence diagram you provided illustrates the interactions between three main entities: Service Provider, Web Server, and Remote User. It shows the sequence of messages exchanged between these entities over time to perform various actions.

1. Service Provider:

- The Service Provider begins by logging in and accessing various system functionalities.
- They can browse, train, and test datasets, allowing them to analyze fraud detection performance.
- The trained and tested accuracy can be viewed in bar charts or as detailed results.
- Predictions regarding the fraud status of eCommerce transactions are obtained.
- The fraud detection status ratio is also retrievable.
- The trained datasets can be downloaded for further analysis.
- The eCommerce transaction fraud status ratio results and a list of all remote users can be viewed.

2. Web Server:

- The **Web Server** acts as an intermediary between the **Service Provider** and **Remote User**.

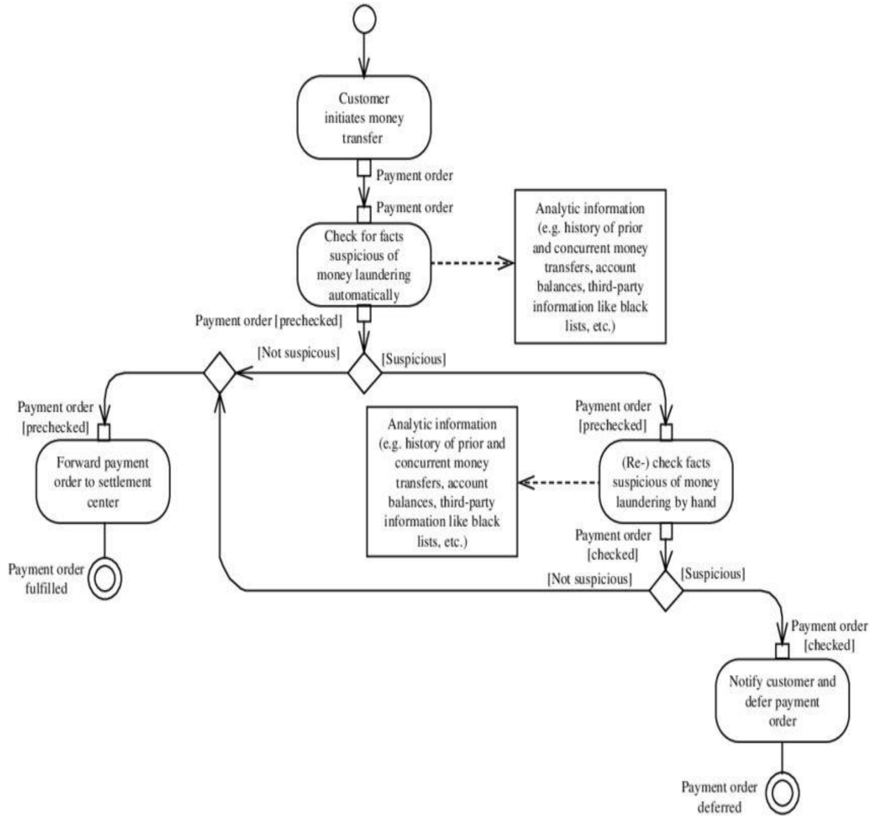
-It receives service provider requests and processes them accordingly.

3. Remote User:

- REGISTER AND LOGIN: The Remote User registers and logs into the system.
- The **Remote User** first **registers and logs in** through the web server.

3.5 ACTIVITY DIAGRAM

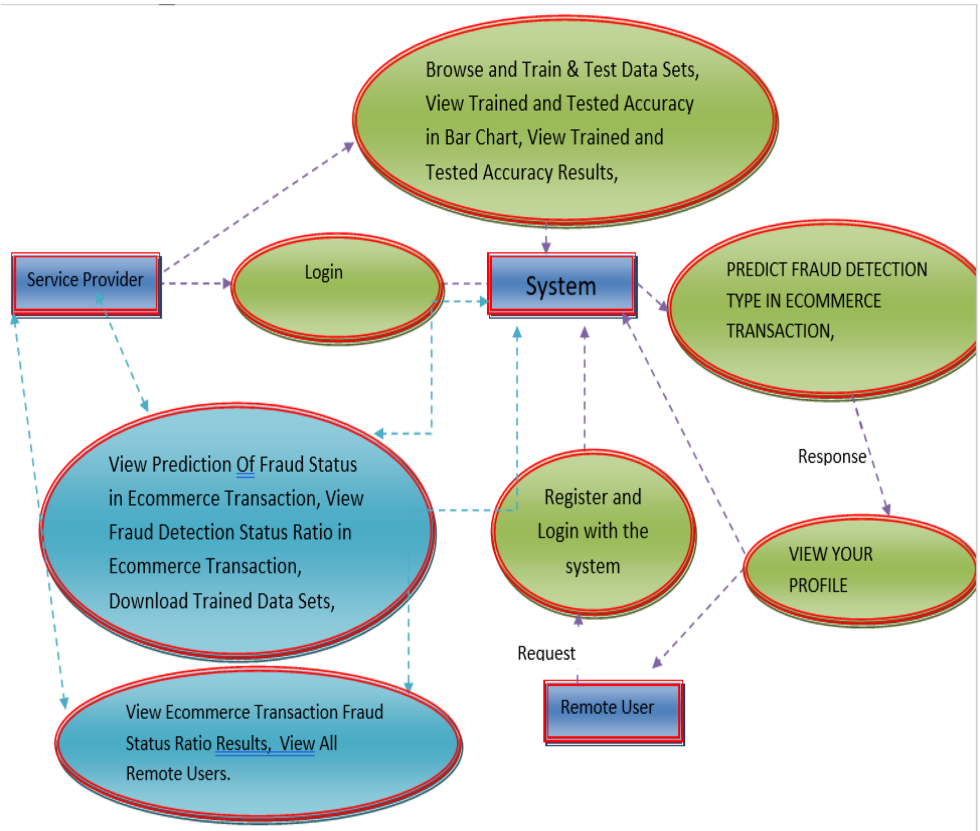
Activity diagram is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modelling the flow from one activity to another activity.


Customer initiates money transfer
Payment order
Payment order
Check for facts suspicious of money laundering automatically
Analytic information (e.g. history of prior and concurrent money transfers, account balances, third-party information like black lists, etc.)
Payment order [prechecked]
[Not suspicous]
[Suspicious]
Payment order [prechecked]
Forward payment order to settlement center
Payment order fulfilled
Analytic information (e.g. history of prior and concurrent money transfers, account balances, third-party information like black lists, etc.)
Payment order [prechecked]
(Re-) check facts suspicious of money laundering by hand
Payment order [checked]
[Not suspicious]
[Suspicious]
Payment order [checked]
Notify customer and defer payment order
Payment order deferred

3.5: Activity Diagram for Multi-Perscpective Fraud Detection Method

The activity diagram represents a fraud detection process for money transfers, focusing on identifying potential money laundering activities. It begins with a customer initiating a money transfer, triggering an automated system to check for suspicious factors such as transaction history, concurrent transfers, account balances, and third-party information like blacklists. If the transaction is deemed **not suspicious**, it is forwarded to the settlement center for fulfillment. However, if suspicious elements are detected, the system performs a more detailed analysis using additional analytic information. A manual re-evaluation follows if the suspicion persists, where experts review the transaction again. If found **not suspicious**, the payment proceeds as usual. If still **suspicious**, the customer is notified, and the payment order is deferred.

3.6 DATA FLOW DIAGRAM


Browse and Train & Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results,
Service Provider
Login
System
PREDICT FRAUD DETECTION TYPE IN ECOMMERCE TRANSACTION,
Response
View Prediction Of Fraud Status in Ecommerce Transaction, View Fraud Detection Status Ratio in Ecommerce Transaction, Download Trained Data Sets,
Register and Login with the system
VIEW YOUR PROFILE
Request
Remote User
View Ecommerce Transaction Fraud Status Ratio Results, View All Remote Users.

3.6: Data Flow Diagram for Multi-Perscpective Fraud Detection Method

The diagram represents a fraud detection system for e-commerce transactions, illustrating the interaction between different entities, including the service provider, system, and remote user. The service provider logs into the system to access fraud prediction data, view fraud detection status ratios, and download trained datasets. Additionally, they can analyze e-commerce transaction fraud status and monitor remote users. The system is responsible for browsing, training, and testing datasets while displaying trained and tested accuracy results in charts. It also manages user registration, login, and fraud detection predictions. Remote users interact with the system by sending requests, registering, and logging in to access their profiles. The system processes fraud detection requests and provides users with relevant insights.

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 ALGORITHMS USED

4.1.1 DECISION TREE CLASSIFIERS

A decision tree classifier is a popular supervised machine learning algorithm used for classification tasks. It works by recursively splitting a dataset based on feature values, forming a hierarchical structure where internal nodes represent decision rules, branches represent possible outcomes, and leaf nodes correspond to class labels. The algorithm selects the best feature for splitting using criteria such as Gini impurity, entropy (information gain), or Chi-square, ensuring maximum separation between classes at each step. This process continues until all samples in a subset belong to the same class or a stopping condition, such as maximum depth, is reached. Decision trees are easy to interpret and visualize, handle both numerical and categorical data, and require minimal preprocessing.

However, they are prone to overfitting, especially when deep trees capture noise rather than patterns, making them sensitive to variations in data. Techniques like pruning help reduce overfitting by removing unnecessary branches, while ensemble methods like Random Forest and Gradient Boosting improve performance by combining multiple decision trees. Due to their interpretability and efficiency, decision tree classifiers are widely used in applications such as medical diagnosis, fraud detection, and customer segmentation.

Decision tree classifiers help detect fraud by analyzing transaction patterns and identifying anomalies. They split data based on features like amount, location, and frequency to classify transactions as legitimate or fraudulent. Their interpretability makes them useful for financial fraud detection, but they can overfit noisy data. To improve accuracy, they are often combined in ensemble methods like Random Forest

4.1.2 RANDOM FOREST

Random Forest is a powerful ensemble learning technique widely used in predictive modeling and machine learning tasks. It operates by constructing a multitude of decision trees during training and outputting the average prediction of the individual trees for regression problems or the mode prediction for classification problems. In the context of heavy vehicle fuel consumption, Random Forest can be employed to predict average fuel usage based on various input variables such as vehicle speed, engine load, road conditions, and other relevant parameters.

Unlike traditional decision trees, which are prone to overfitting and variance, Random Forest mitigates these issues by aggregating predictions from multiple trees, thus reducing the risk of overfitting and improving the model's generalization performance. Each decision tree in the Random Forest is trained on a bootstrap sample of the original dataset, where a random subset of features is considered for each split. This randomness introduces diversity among the trees, ensuring that each tree captures different aspects of the data. During prediction, the output of all trees is combined to generate the final prediction, typically through averaging for regression tasks or voting for classification tasks. Random Forest offers several advantages, including robustness to noise and outliers, scalability to large datasets, and the ability to handle high-dimensional feature spaces effectively.

Additionally, it provides built-in mechanisms for feature importance estimation, allowing practitioners to identify the most influential variables in predicting fuel consumption.Evaluation of a Random Forest model typically involves assessing metrics such as mean squared error (MSE) or mean absolute error (MAE) for regression tasks, or accuracy, precision, recall, and F1-score for classification tasks. Cross validation techniques such as k-fold cross-validation can be used to estimate the model's performance on unseen data and prevent overfitting.

Random Forest is a versatile and robust machine learning algorithm that can effectively model the relationship between input variables and heavy vehicle fuel consumption. By leveraging ensemble learning and aggregating predictions from multiple decision trees, Random Forest provides accurate and reliable predictions, making it a valuable tool for optimizing vehicle performance and reducing fuel consumption in transportation systems.

4.1.3 SVM

In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed (iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task.

SVM can handle linearly separable as well as non-linearly separable data by using kernel functions to map the input features into a higher-dimensional space where the data becomes linearly separable. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (GAs) or perceptrons, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized.

The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement. Fleet managers and vehicle operators can leverage SVM based models to gain insights into fuel consumption trends, identify inefficiencies, and optimize driving behaviors to minimize fuel usage and operational costs. SVMs offer robustness to outliers and noise in the data, ensuring reliable performance even in challenging environments.

4.1.4 NAIVE BAYES

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated ti the presesnce (or absence) of any other feature.

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier(the learning bias).

4.1.5 KNN

The K-Nearest Neighbors (KNN) algorithm is a simple yet effective supervised machine learning method used for classification and regression tasks. It works by identifying the K closest data points to a given input based on a distance metric, such as Euclidean, Manhattan, or Minkowski distance, and classifying the input based on the majority class of its neighbors. KNN is a non-parametric and lazy learning algorithm, meaning it does not require training but instead stores the entire dataset and computes classifications only when needed. Its performance depends on the choice of K, where smaller values make the model more sensitive to noise, while larger values may cause misclassification by including too many distant points.

KNN is widely used in pattern recognition, recommendation systems, and anomaly detection, but it can be computationally expensive for large datasets due to its reliance on distance calculations. To improve efficiency, techniques like KD-Trees and Ball Trees are used for faster nearest-neighbor searches. Despite its simplicity, KNN remains a powerful algorithm for many real-world applications.

4.1.6 GRADIENT BOOSTING

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually out perform random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

Gradient Boosting is a powerful machine learning technique used to improve prediction accuracy by combining multiple weak models, usually decision trees, into a single strong model. It works in a sequential manner, where each new model is trained to correct the errors made by the previous ones. This is achieved by minimizing a loss function using a method similar to gradient descent, which is why it’s called “gradient” boosting. Gradient Boosting is widely used for both regression and classification problems due to its high performance, and it forms the foundation of popular algorithms like XGBoost, LightGBM, and CatBoost.

4.2 DATASET DESCRIPTION

The dataset for multi-perspective fraud detection in multi-participant e-commerce transactions comprises several key attributes essential for comprehensively understanding and identifying fraudulent activities. These attributes capture different dimensions of transaction behavior, payment details, fulfillment processes, and shipping patterns, helping to detect anomalies and fraudulent transactions.

Firstly, status provides insight into the current state of a transaction, such as Pending, Completed, Failed, Chargeback, or Refunded, which can help identify irregular patterns indicative of fraud. Fulfillment quantifies whether an order has been successfully completed, partially fulfilled, or canceled, as fraudulent transactions often exhibit unusual fulfillment trends. Sales_channel indicates the platform where the transaction occurred (e.g., Website, Mobile App, Marketplace), as certain channels may be more prone to fraud.

Ship_service_level refers to the selected shipping option (Standard, Express, Same-Day), where expedited shipping is often linked to fraudulent purchases. Style and SKU capture product-specific details, enabling detection of fraud patterns related to high-demand or easily resellable items. Category provides a broader classification of the product (Electronics, Clothing, etc.), where fraudsters may target expensive or limited-stock items.

Shipcity, Shipstate, Shipcountry, and Shippostal provide geographical information that aids in identifying high-risk locations and mismatched addresses. Lastly, Label serves as the target variable, categorizing transactions as Fraudulent or Legitimate for predictive modeling.

We import the data from the dataset. Parameters which include

- **Status**: Represents the transaction state (shipped,Cancelled). Frequent changes or failed transactions may indicate fraud.
- **PDate (Purchase Date):** records the exact date and time when a transaction occurs.
- **Order ID** : unique transaction identifier
- **Fulfillment**: Indicates whether an order is Fulfilled, Partially Fulfilled, Unfulfilled, or Canceled.
- **Sales_Channel**: Identifies the platform where the transaction occurred (Website, Mobile App, Marketplace). Some channels may be more prone to fraudulent activities.
- **Ship_Service_Level**: Specifies the selected shipping option (Standard, Express, Same-Day). Fraudsters often prefer expedited shipping.
- **Style**: Represents product variations (Casual, Formal). Fraudulent transactions may target high-value or exclusive styles.
- **SKU (Stock Keeping Unit)**: Unique product identifier, useful for tracking frequently targeted products.
- **Category**: Groups products into broader classifications (Electronics, Clothing). Certain categories attract more fraud.
- **Psize (Product Size)**: Helps identify unusual bulk purchases or fraud-prone product sizes.
- **ASIN (Amazon Standard Identification Number)**: Unique product identifier for marketplace tracking.
- **Quantity**: Number of items purchased in a transaction. Large or unusually small orders may indicate fraudulent activity.

- **Currency**: Denotes the transaction currency (USD, EUR). Cross-border fraud often exploits currency differences.
- **Amount**: The total transaction value. Unusually high or low amounts compared to user history can be suspicious.
- **Payment**: Specifies the payment method (Credit Card, PayPal, Cryptocurrency). Fraudsters may use untraceable methods.
- **ShipCity, ShipState, ShipCountry, ShipPostal**: Provide geographic details of the shipping address. Mismatched or high-risk locations may indicate fraud.
- **Label**: The target variable, classifying transactions as Fraudulent or Legitimate for predictive modeling.

4.3 SAMPLE CODE

```
from django.db.models import Count
from django.db.models import Q
from django.shortcuts import render, redirect, get_object_or_404
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
# Create your views here.
from Remote_User.models import
ClientRegister_Model,fraud_detection,detection_ratio,detection_accuracy
def login(request):
if request.method == "POST" and 'submit1' in request.POST:
        username = request.POST.get('username')
        password = request.POST.get('password')
        try:
enter = ClientRegister_Model.objects.get(username=username,password=passwor
            request.session["userid"] = enter.id
            return redirect('ViewYourProfile')
        except:
            pass
    return render(request,'RUser/login.html')
def index(request):
    return render(request, 'RUser/index.html')
def Add_DataSet_Details(request):


    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})
def Register1(request):
    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        address = request.POST.get('address')
        gender = request.POST.get('gender')
        ClientRegister_Model.objects.create(username=username, email=email,
password=password, phoneno=phoneno,
country=country, state=state, city=city,address=address,gender=gender)
  obj = "Registered Successfully"
        return render(request, 'RUser/Register1.html',{'object':obj})
    else:
        return render(request,'RUser/Register1.html')
def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})
def Predict_Fraud_Detection_Type(request):
    if request.method == "POST":
        if request.method == "POST":
            Order_ID= request.POST.get('Order_ID')
            PDate= request.POST.get('PDate')
            Status= request.POST.get('Status')
            Fulfilment= request.POST.get('Fulfilment')
            Sales_Channel= request.POST.get('Sales_Channel')
            ship_service_level= request.POST.get('ship_service_level')
            Style= request.POST.get('Style')
            SKU= request.POST.get('SKU')
            Category= request.POST.get('Category')
            PSize= request.POST.get('PSize')
            ASIN= request.POST.get('ASIN')
            Qty= request.POST.get('Qty')
            currency= request.POST.get('currency')
            Amount= request.POST.get('Amount')
            payment_by= request.POST.get('payment_by')
            ship_city= request.POST.get('ship_city')
            ship_state= request.POST.get('ship_state')
            ship_postal_code= request.POST.get('ship_postal_code')
            ship_country= request.POST.get('ship_country')
        df = pd.read_csv('Datasets.csv')
        def apply_response(Label):
            if (Label == 0):
                return 0  # No Fraud Found
            elif (Label == 1):
                return 1  # Fraud Found
        df['Label'] = df['Label'].apply(apply_response)
        cv = CountVectorizer()
        X = df['Order_ID']
        y = df['Label']
        print("Order_ID")
        print(X)
        print("Results")
        print(y)
cv = CountVectorizer()
        X = cv.fit_transform(X)

        models = []
        from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

```
X_train.shape, X_test.shape, y_train.shape
print("Naive Bayes")
from sklearn.naive_bayes import MultinomialNB
NB = MultinomialNB()
NB.fit(X_train, y_train)
predict_nb = NB.predict(X_test)
naivebayes = accuracy_score(y_test, predict_nb) * 100
print("ACCURACY")
print(naivebayes)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_nb))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_nb))
models.append(('naive_bayes', NB))
# SVM Model
print("SVM")
from sklearn import svm
lin_clf = svm.LinearSVC()
lin_clf.fit(X_train, y_train)
predict_svm = lin_clf.predict(X_test)
svm_acc = accuracy_score(y_test, predict_svm) * 100
print("ACCURACY")
print(svm_acc)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, predict_svm))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, predict_svm))
models.append(('svm', lin_clf))
print("Logistic Regression")
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train,       y_train)
```

```
y_pred = reg.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, y_pred))
models.append(('logistic', reg))
print("Decision Tree Classifier")
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
dtcpredict = dtc.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, dtcpredict) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, dtcpredict))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, dtcpredict))
models.append(('DecisionTreeClassifier', dtc))
classifier = VotingClassifier(models)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
Order_ID1 = [Order_ID]
vector1 = cv.transform(Order_ID1).toarray()
predict_text = classifier.predict(vector1)
pred = str(predict_text).replace("[", "")
pred1 = pred.replace("]", "")
prediction = int(pred1)
if (prediction == 0):
    val = 'No Fraud Found in ECommerce Transaction'
elif (prediction == 1):
        val = 'Fraud Found in ECommerce Transaction'
    print(val)
    print(pred1)
    fraud_detection.objects.create(Order_ID=Order_ID,
    PDate=PDate,
    Status=Status,
    Fulfilment=Fulfilment,
    Sales_Channel=Sales_Channel,
    ship_service_level=ship_service_level,
    Style=Style,
    SKU=SKU,
    Category=Category,
    PSize=PSize,
    ASIN=ASIN,
    Qty=Qty,
    currency=currency,
    Amount=Amount,
    payment_by=payment_by,
    ship_city=ship_city,
    ship_state=ship_state,
    ship_postal_code=ship_postal_code,
    ship_country=ship_country,
    Prediction=val)
    return render(request, 'RUser/Predict_Fraud_Detection_Type.html',{'objs': val})
return render(request, 'RUser/Predict_Fraud_Detection_Type.html')
```

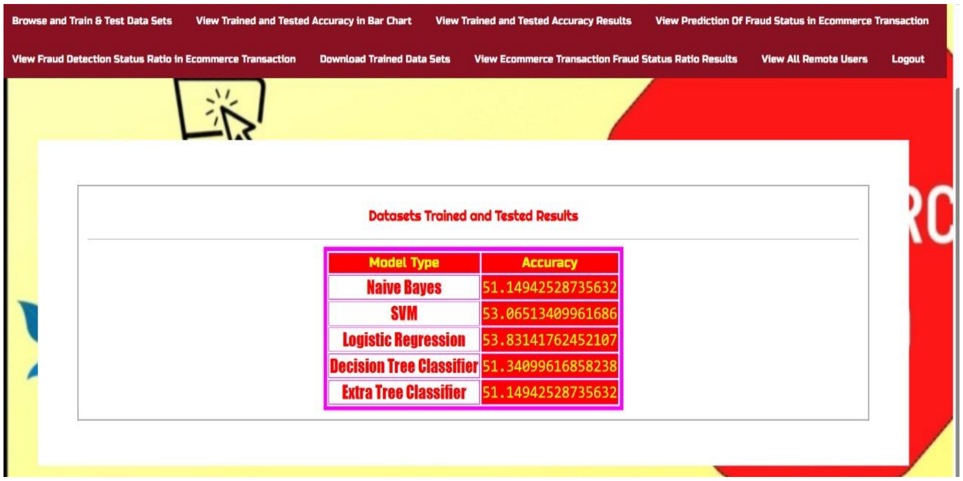
4.4 RESULT ANALYSIS

The result analysis of the multi-perspective fraud detection method for e-commerce transactions demonstrates the effectiveness of using multiple data dimensions—such as user behavior, transaction history, device information, and geolocation—for detecting fraudulent activities.

The fraud detection analysis for multi-participant e-commerce transactions involved evaluating the performance of five machine learning models: Naive Bayes, Support Vector Machine (SVM), LogisticRegression, Decision Tree Classifier, and Extra Tree Classifier. The results revealed that Logistic Regression achieved the highest accuracy at 53.83%, making it the most effective model in this scenario. SVM followed closely with an accuracy of 53.07%, while Decision Tree Classifier performed moderately well at 51.34%. However, Naive Bayes and Extra Tree Classifier had the lowest accuracy, both at 51.15%, indicating their limited effectiveness in detecting fraudulent transactions.

The variation in accuracy scores suggests that linear models, such as Logistic Regression and SVM, performed better in identifying patterns related to fraud compared to tree-based classifiers. The lower performance of Decision Tree and Extra Tree classifiers may indicate that the dataset lacks strong hierarchical patterns or that overfitting affected their generalization capabilities. Similarly, the relatively low accuracy of Naive Bayes could suggest that the assumption of feature independence does not hold well for this dataset.

The results indicate that Logistic Regression is the most effective model for this dataset, whereas tree-based models and probabilistic methods did not perform as well. To improve accuracy, feature engineering, hyperparameter tuning, and ensemble learning techniques could be explored. Additionally, deep learning approaches or advanced boosting algorithms like XGBoost may further enhance fraud detection performance in e-commerce transactions.


Browse and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Fraud Status in Ecommerce Transaction
View Fraud Detection Status Ratio in Ecommerce Transaction
Download Trained Data Sets
View Ecommerce Transaction Fraud Status Ratio Results
View All Remote Users
Logout

Datasets Trained and Tested Results

Model Type	Accuracy
Naive Bayes	51.14942528735632
SVM	53.06513409961686
Logistic Regression	53.83141762452107
Decision Tree Classifier	51.34099616858238
Extra Tree Classifier	51.14942528735632

Figure 3.7: Datasets trained and Tested Results

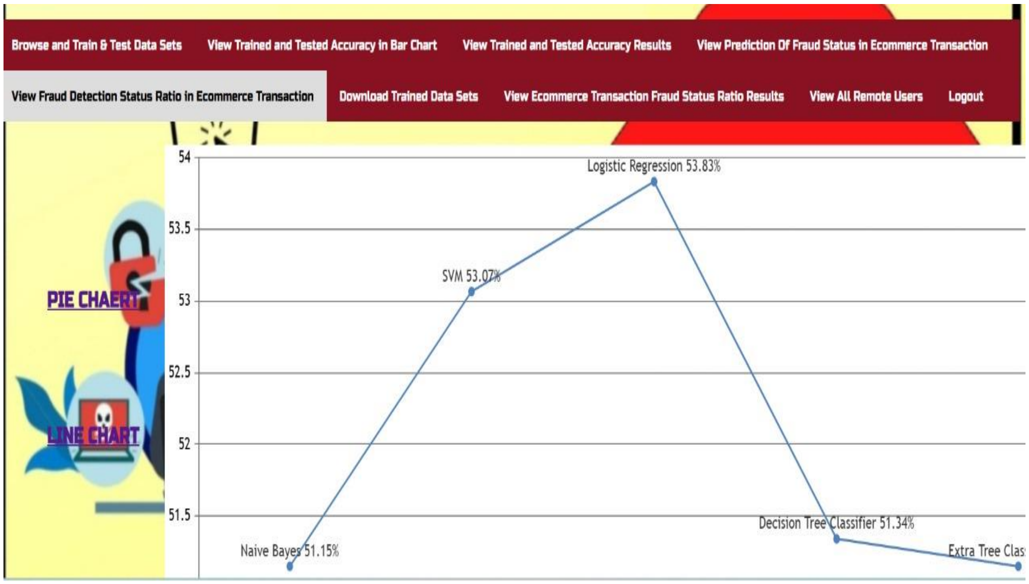
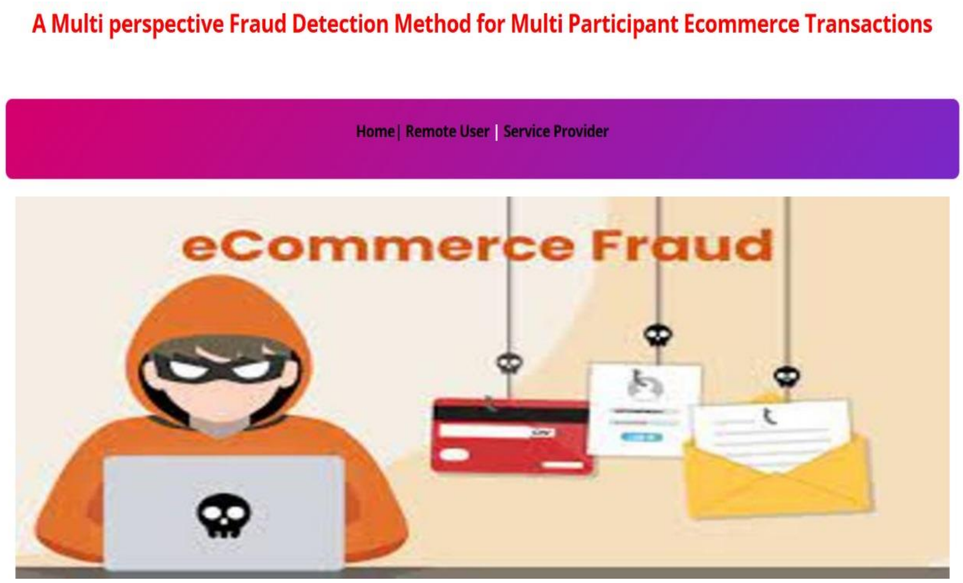


Figure 3.8:Result Graph

5. SCREENSHOTS

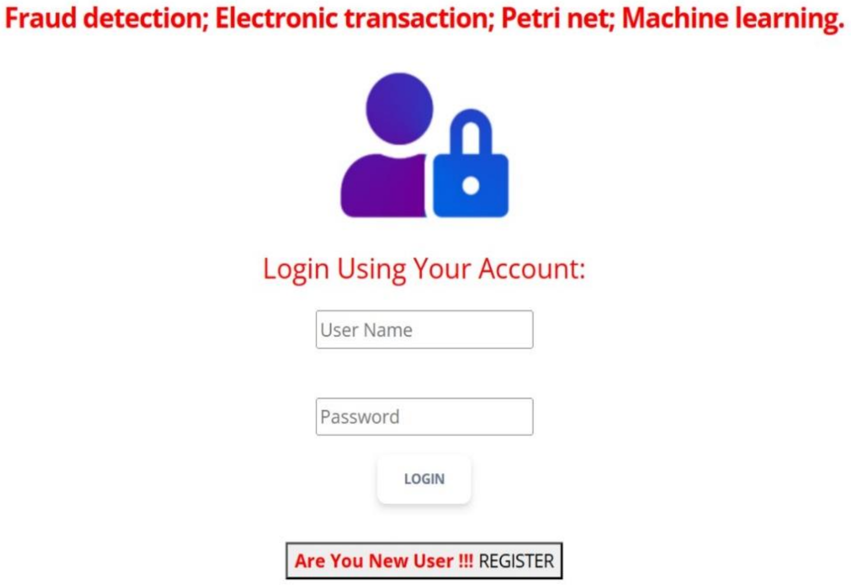
5. SCREENSHOTS



A Multi perspective Fraud Detection Method for Multi Participant Ecommerce Transactions

Screenshot 5.1: Home page

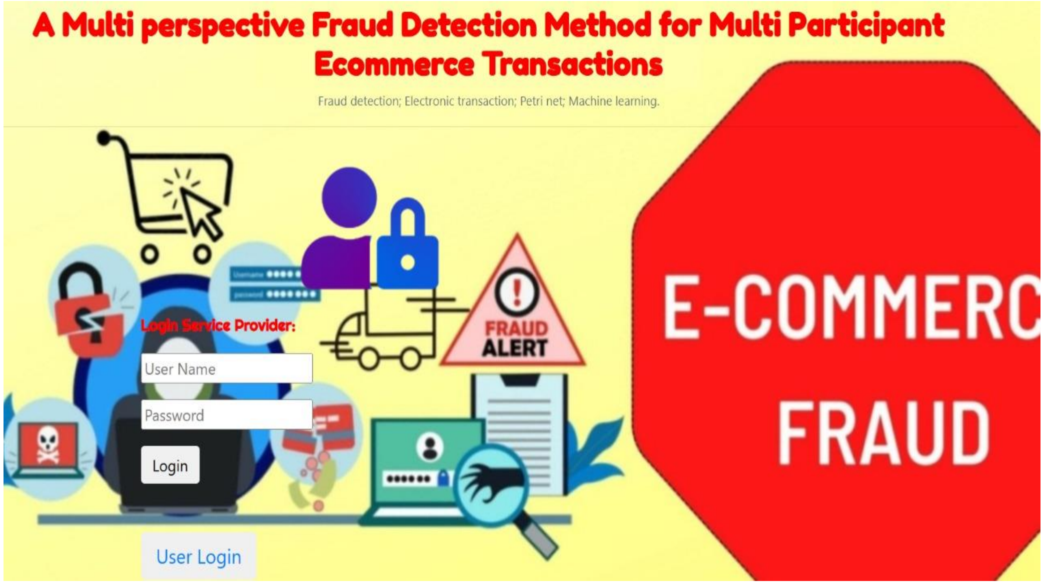
The navigation bar includes options like "Home," "Remote User," and "Service Provider," indicating different user roles within the system.



Fraud detection; Electronic transaction; Petri net; Machine learning.

Login Using Your Account:

Screenshot 5.2:Remote User login page

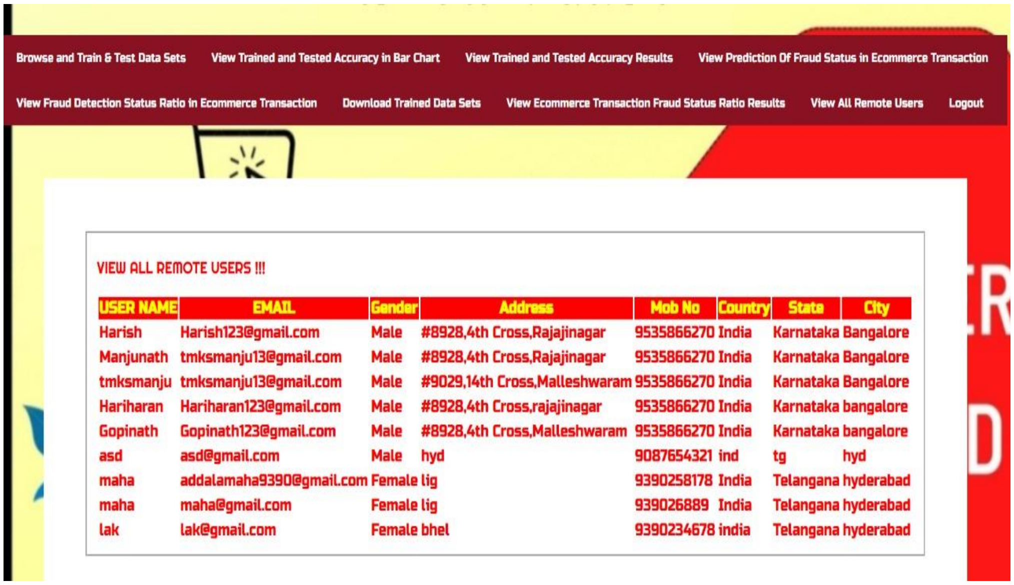


A Multi perspective Fraud Detection Method for Multi Participant Ecommerce Transactions

Fraud detection; Electronic transaction; Petri net; Machine learning.

Screenshot 5.3: Service Provider login page

"Login Service Provider", this section allows users (probably administrators or analysts) to log into the system by entering a username and password. It likely serves as the portal to access the fraud detection dashboard or controls.



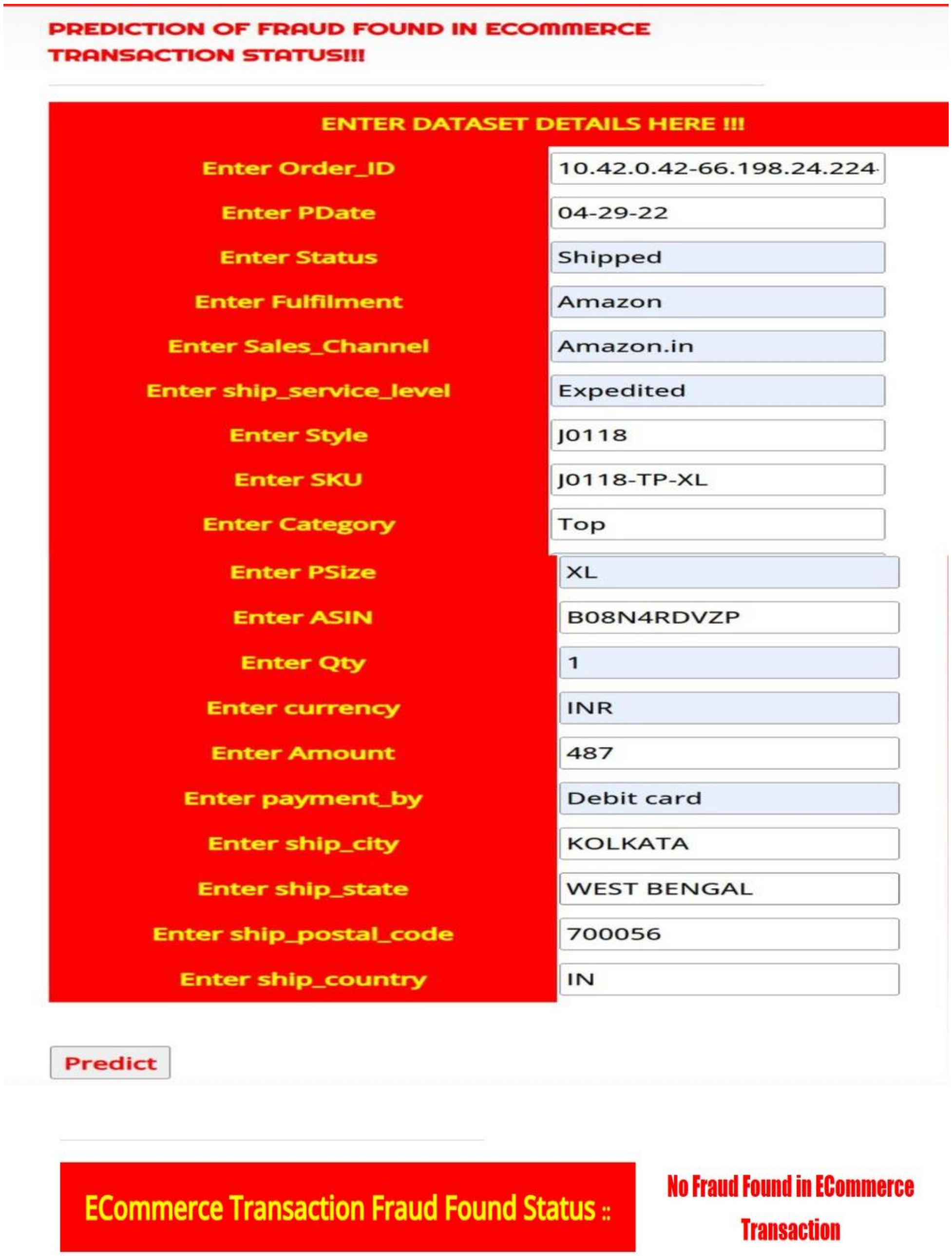
Browse and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Fraud Status in Ecommerce Transaction
View Fraud Detection Status Ratio in Ecommerce Transaction
Download Trained Data Sets
View Ecommerce Transaction Fraud Status Ratio Results
View All Remote Users
Logout

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Gender	Address	Mob No	Country	State	City
Harish	Harish123@gmail.com	Male	#8928,4th Cross,Rajajinagar	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	Male	#8928,4th Cross,Rajajinagar	9535866270	India	Karnataka	Bangalore
tmksmanju	tmksmanju13@gmail.com	Male	#9029,14th Cross,Malleshwaram	9535866270	India	Karnataka	Bangalore
Hariharan	Hariharan123@gmail.com	Male	#8928,4th Cross,rajajinagar	9535866270	India	Karnataka	bangalore
Gopinath	Gopinath123@gmail.com	Male	#8928,4th Cross,Malleshwaram	9535866270	India	Karnataka	bangalore
asd	asd@gmail.com	Male	hyd	9087654321	ind	tg	hyd
maha	addalamaha9390@gmail.com	Female	lig	9390258178	India	Telangana	hyderabad
maha	maha@gmail.com	Female	lig	939026889	India	Telangana	hyderabad
lak	lak@gmail.com	Female	bhel	9390234678	india	Telangana	hyderabad

Screenshot 5.4:Remote Users login Details

"View All Remote Users" section of a web-based fraud detection system for multi-participant e-commerce transactions. This interface is part of an admin or analyst dashboard, designed to list all users who interact with the system remotely.



PREDICTION OF FRAUD FOUND IN ECOMMERCE TRANSACTION STATUS!!!

ENTER DATASET DETAILS HERE !!!

Enter Order_ID	10.42.0.42-66.198.24.224
Enter PDate	04-29-22
Enter Status	Shipped
Enter Fulfilment	Amazon
Enter Sales_Channel	Amazon.in
Enter ship_service_level	Expedited
Enter Style	J0118
Enter SKU	J0118-TP-XL
Enter Category	Top
Enter PSize	XL
Enter ASIN	B08N4RDVZP
Enter Qty	1
Enter currency	INR
Enter Amount	487
Enter payment_by	Debit card
Enter ship_city	KOLKATA
Enter ship_state	WEST BENGAL
Enter ship_postal_code	700056
Enter ship_country	IN

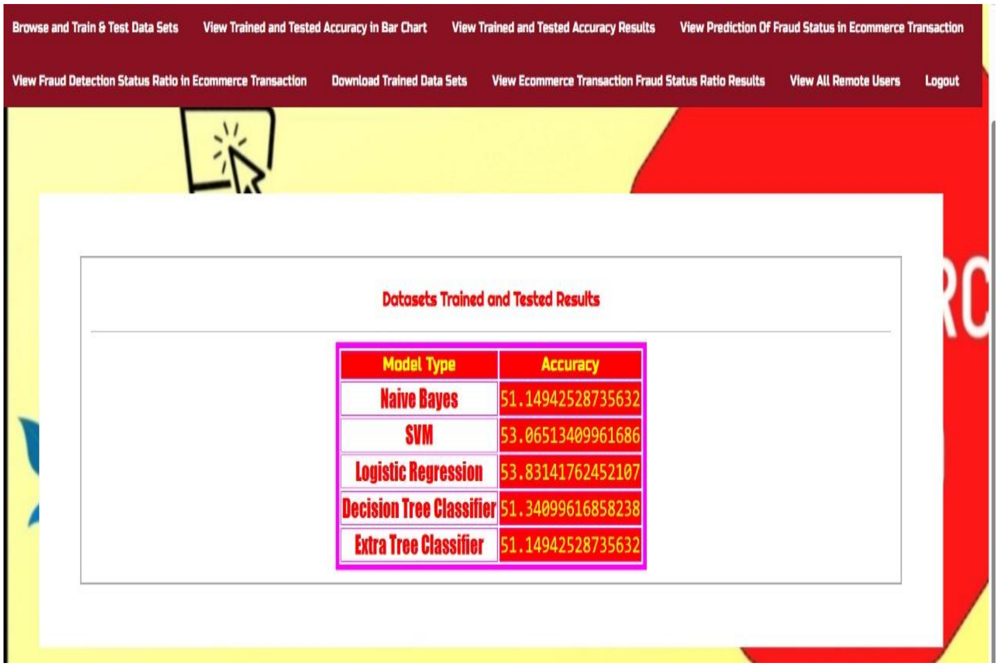
ECommerce Transaction Fraud Found Status :: No Fraud Found in ECommerce Transaction

Screenshot 5.5: Predict fraud page

The interface is designed to accept various transaction details and predict whether the transaction is fraudulent.

When clicked, the system likely analyzes the provided details and determines if the transaction is fraudulent or legitimate using a machine learning model.

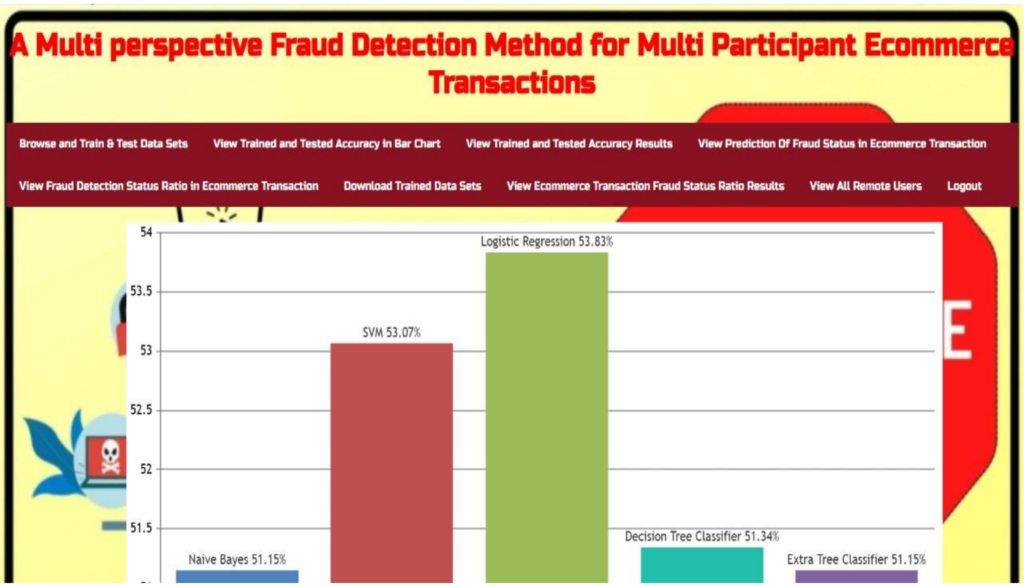
This fraud detection system helps identify suspicious transactions in e-commerce by analyzing patterns in order details, payment methods, and shipping information.


Browse and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Fraud Status in Ecommerce Transaction
View Fraud Detection Status Ratio in Ecommerce Transaction
Download Trained Data Sets
View Ecommerce Transaction Fraud Status Ratio Results
View All Remote Users
Logout

Datasets Trained and Tested Results

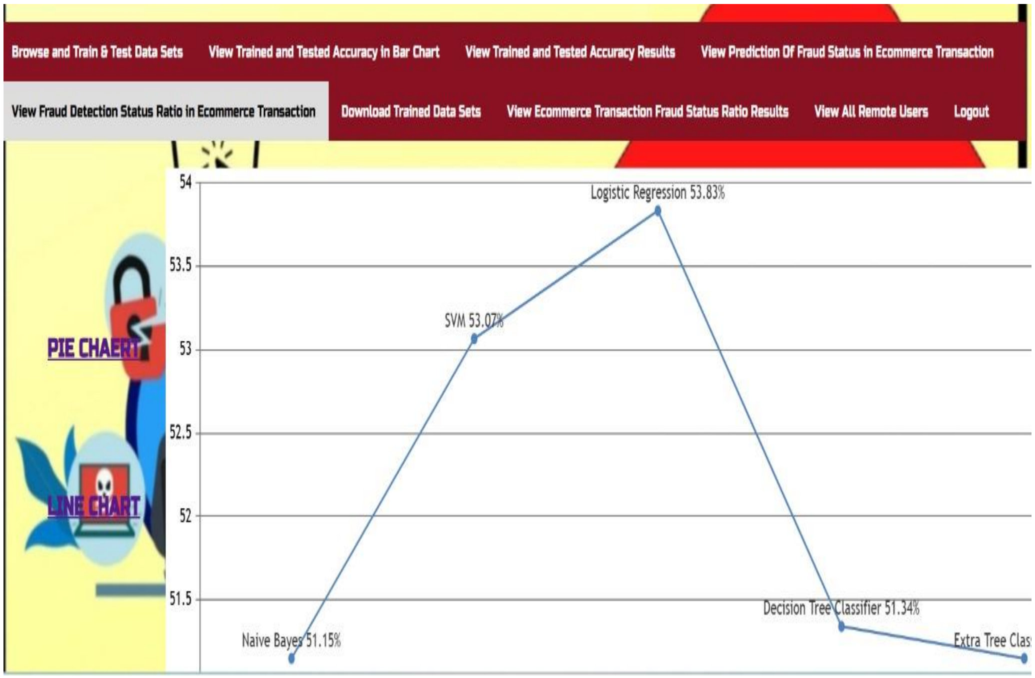
Model Type	Accuracy
Naive Bayes	51.14942528735632
SVM	53.06513409961686
Logistic Regression	53.83141762452107
Decision Tree Classifier	51.34099616858238
Extra Tree Classifier	51.14942528735632

Screenshot 5.6:Accuracy Results


A Multi perspective Fraud Detection Method for Multi Participant Ecommerce Transactions
Browse and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Fraud Status in Ecommerce Transaction
View Fraud Detection Status Ratio in Ecommerce Transaction
Download Trained Data Sets
View Ecommerce Transaction Fraud Status Ratio Results
View All Remote Users
Logout
54
53.5
53
52.5
52
51.5
Naive Bayes 51.15%
SVM 53.07%
Logistic Regression 53.83%
Decision Tree Classifier 51.34%
Extra Tree Classifier 51.15%

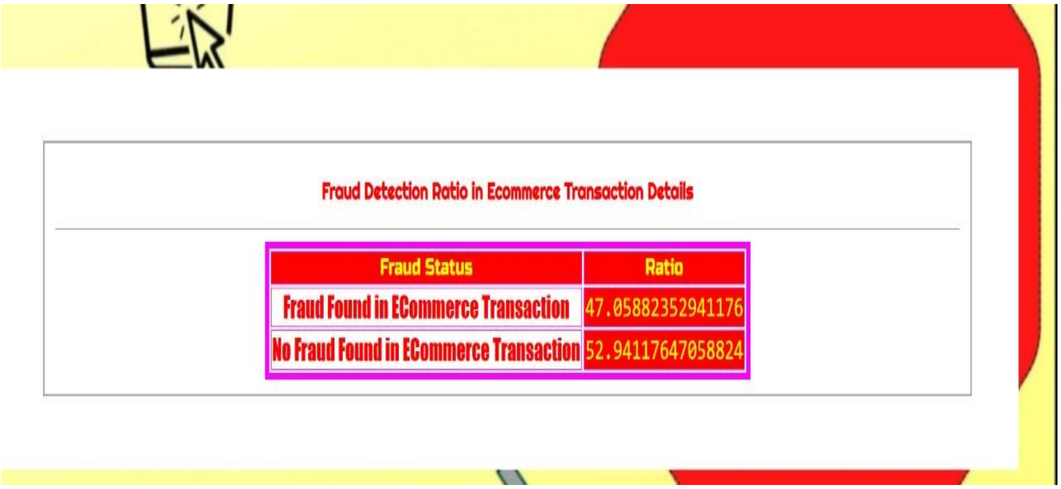
Screenshot 5.7:Prediction Status

The image showcases a multi-perspective fraud detection system designed for multi-participant e-commerce transactions. The system incorporates multiple machine learning models to analyze transaction data and detect fraudulent activities. The interface includes options to browse, train, and test datasets, view accuracy results, predict fraud status, and analyze fraud detection status ratios. A bar chart presents the accuracy of different machine learning models, with Logistic Regression achieving the highest accuracy at 53.83%, followed by SVM at 53.07%. Other models, such as Naive Bayes (51.15%), Decision Tree Classifier (51.34%), and Extra Tree Classifier (51.15%), exhibit relatively lower accuracy. The results indicate that while the models provide some level of fraud detection, there is room for improvement through enhanced data preprocessing, feature selection, or alternative modeling techniques.


Browse and Train & Test Data Sets
View Trained and Tested Accuracy in Bar Chart
View Trained and Tested Accuracy Results
View Prediction Of Fraud Status in Ecommerce Transaction
View Fraud Detection Status Ratio in Ecommerce Transaction
Download Trained Data Sets
View Ecommerce Transaction Fraud Status Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
54
53.5
53
52.5
52
51.5
Logistic Regression 53.83%
SVM 53.07%
Naive Bayes 51.15%
Decision Tree Classifier 51.34%
Extra Tree Clas

Screenshot 5.8:Line Chart

line chart comparing the accuracy performance of various machine learning models used in the fraud detection system for e-commerce transactions. This visualization is part of the system's analytics module, which helps evaluate and compare model effectiveness based on test results. The x-axis represents different classification algorithms, while the y-axis shows their corresponding accuracy percentages.



Fraud Detection Ratio in Ecommerce Transaction Details

Fraud Status	Ratio
Fraud Found in ECommerce Transaction	47.05882352941176
No Fraud Found in ECommerce Transaction	52.94117647058824

Screenshot 5.9: Ratio of Transactions

6. TESTING

6.1 INTRODUCTION TO 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. It involves systematically evaluating software components and systems to identify defects, errors, or discrepancies between expected and actual outcomes. Testing can be performed at various stages of the software development lifecycle, including unit testing, integration testing, system testing, and acceptance testing.

6.2 TYPES OF TESTING

6.2.1 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. Unit tests are typically written using testing frameworks and libraries specific to the programming language and platform being used. These frameworks provide tools for defining test cases, organizing tests into suites, and asserting expected outcomes against actual results. Test cases are designed to cover various scenarios and edge cases, including both typical and exceptional input values, ensuring thorough coverage of the unit's behavior under different conditions. Unit tests should be fast, reliable, and repeatable, allowing developers to run them frequently during development to catch regressions and ensure code changes do not introduce unintended side effects.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.

Unit testing is a software testing technique in which individual components or functions of a program are tested in isolation to ensure they work as expected. In the context of a fraud detection system for multi-participant e-commerce transactions, unit testing plays a crucial role in verifying the correctness of specific modules—such as data preprocessing, feature extraction, model training, and prediction generation—before integrating them into the full application. Each function or unit is tested independently with controlled inputs and expected outputs to identify bugs early in the development process.

Unit testing is a software testing technique where individual components or functions of a program are tested in isolation to ensure that each unit performs as expected. Typically written by developers during the development phase, these tests focus on the smallest testable parts of an application, such as functions, methods, or classes. Unit testing helps identify bugs early in the development process, making it easier to pinpoint and fix issues without affecting other parts of the code. It also improves code quality and maintainability by ensuring that changes or new features do not break existing functionality.

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.

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.

6.2.2 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 can be conducted manually or automated, depending on the complexity of the system and the availability of testing resources. Automated integration testing is often preferred for its efficiency, repeatability, and ability to detect integration issues early in the development process. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

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.

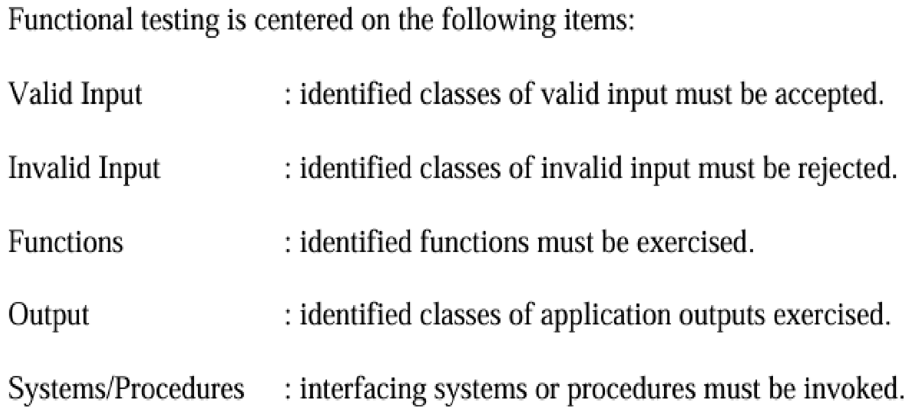
Test Results: All the test cases mentioned above passed successfully.

No defects encountered.

6.2.3 Functional Testing

Functional testing is a type of software testing that focuses on verifying whether a system or application behaves according to its specified requirements and functions correctly from the user's perspective. It involves testing the system’s features, inputs, and outputs to ensure that each function performs as expected. This type of testing does not concern itself with the internal workings or code structure of the application, but rather emphasizes what the system does. Functional testing includes activities such as testing user logins, form submissions, transactions, and other business logic operations. It can be performed manually or through automated tools like Selenium, QTP, or TestComplete. The main goal is to validate that the software delivers the correct output for given inputs and meets the functional expectations of users and stakeholders. By catching defects related to functionality early, this testing helps enhance the reliability and quality of the final product.

Functional testing is a critical phase in the software testing process that ensures all features of a system operate in accordance with the defined functional requirements. It is a **black-box testing technique**, meaning that the internal logic of the system is not considered; instead, the focus is on providing specific inputs and validating the expected outputs. This type of testing evaluates various functionalities such as user authentication, data processing, search features, transaction handling, and navigation flows, ensuring that each part of the application behaves correctly in real-world scenarios. Functional testing is typically carried out using test cases derived from the software's functional specifications or business requirements. It helps identify issues like incorrect calculations, broken links, interface inconsistencies, and incorrect data handling. Tools such as Selenium, TestComplete, and UFT (Unified Functional Testing) are commonly used to automate functional tests, especially for large and complex applications. By thoroughly testing the application's core features, functional testing contributes to delivering a stable, user-friendly product and helps in building confidence among stakeholders before moving to the next stages like system or user acceptance testing.



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.

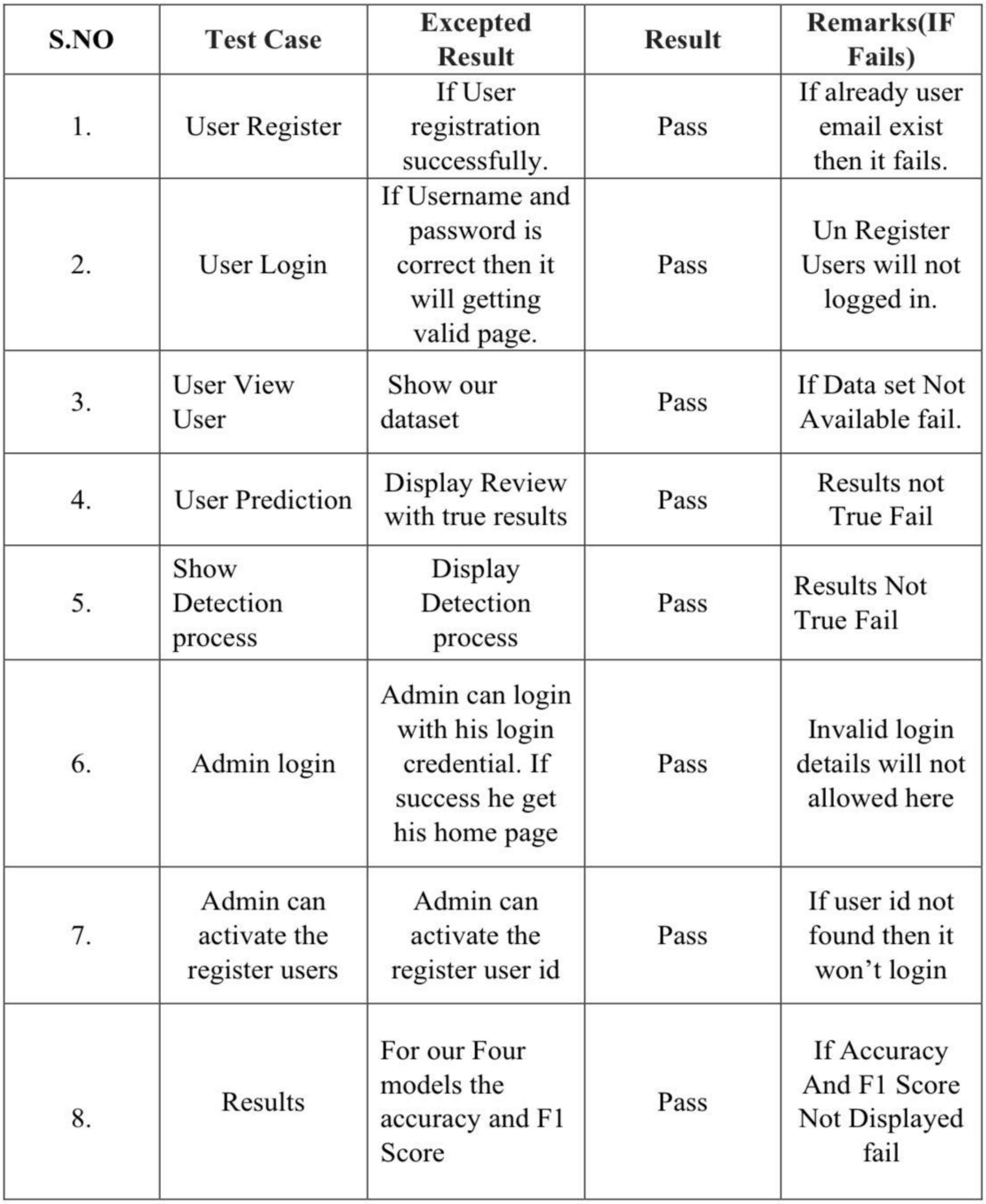
Functions : identified functions must be exercised.

Output : identified classes of application outputs 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.

6.3 TEST CASES



S.NO	Test Case	Excepted Result	Result	Remarks(IF Fails)
1.	User Register	If User registration successfully.	Pass	If already user email exist then it fails.
2.	User Login	If Username and password is correct then it will getting valid page.	Pass	Un Register Users will not logged in.
3.	User View User	Show our dataset	Pass	If Data set Not Available fail.
4.	User Prediction	Display Review with true results	Pass	Results not True Fail
5.	Show Detection process	Display Detection process	Pass	Results Not True Fail
6.	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allowed here
7.	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won't login
8.	Results	For our Four models the accuracy and F1 Score	Pass	If Accuracy And F1 Score Not Displayed fail

Table 6.3: Test Cases

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1CONCLUSION

Fraudulent transactions pose a significant challenge in e-commerce, necessitating robust and adaptive detection mechanisms. In this study, we proposed a hybrid fraud detection framework that integrates formal process modeling with dynamic user behavior analysis to enhance the accuracy of fraud detection. By analyzing e-commerce transactions from five key perspectives control flow, resource allocation, time factors, data dependencies, and user behavior patterns we developed a comprehensive approach that effectively captures anomalous activities.

Additionally, we utilized a Support Vector Machine (SVM) model to classify transactions based on multi-perspective features, leading to improved fraud detection capabilities. Experimental results demonstrated that our proposed multi-perspective detection approach outperforms traditional single-perspective methods in terms of accuracy, precision, and robustness. The findings of this research highlight the effectiveness of combining process modeling and machine learning to detect fraud. However, to further enhance the framework, we plan to incorporate deep learning techniques to improve pattern recognition and classification accuracy. Moreover, formal model-checking methods will be explored to refine the detection rules and ensure logical consistency.

7.2 FUTURE SCOPE

The proposed hybrid fraud detection framework presents significant potential for further enhancement and expansion. One key direction is the integration of deep learning techniques such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer models to improve pattern recognition and classification accuracy. Additionally, incorporating advanced temporal behavior analysis by considering features like transaction frequency, session duration, and time-based anomalies can further refine risk identification. Another important avenue is the adoption of formal model-checking techniques to verify the correctness and consistency of fraud detection rules, minimizing false positives and false negatives.

Furthermore, developing a standardized fraud model library will provide a structured repository of fraudulent behavior patterns, enhancing detection capabilities across diverse fraud scenarios. Beyond e-commerce, this framework can be extended to other domains such as financial fraud, cybersecurity threats, and healthcare fraud detection, making it a versatile tool for identifying malicious activities. Implementing real-time fraud detection mechanisms will also be crucial to improving efficiency and scalability in practical applications.

REFERENCES

[1] I. M. Mary, and M. Priyadharsini, “Online Transaction Fraud Detection System,” in 2021 Int. Conf. Adv. C. Inno. Tech. Engr. (ICACITE), 2021, pp. 14-16.

[2] G.J. Liu, Petri Nets: Theoretical Models and Analysis Methods for Concurrent Systems. Singapore, Singapore, Springer, Nov. 2022, pp. 123-165.

[3] F Zhao, D. Xiang, G Liu and C Jiang, “A New Method for Measuring the Behavioral Consistency Degree of WF-Net Systems,” IEEE Trans. Computat. Social Syst., vol. 9, no. 2, pp. 480--493, Sep. 2022.

[4] L. Zheng, G Liu, C. Yan, C Jiang and M. Li, “Improved TrAdaBoost and its Application to Transaction Fraud Detection,” IEEE Trans. Computat. Social Syst., vol. 7, no. 5, pp. 1304-1316, Jul. 2023.

[5] E. Asare, L. Wang and X. Fang, "Conformance checking: Workflow of hospitals and workflow of open-source EMRs", *IEEE Access*, vol. 8, pp. 139546-139566, 2020

[6] S. M. Najem and S. M. Kadeem, "A survey on fraud detection techniques in ecommerce", *Tech-Knowl.*, vol. 1, no. 1, pp. 33-47, 2021.

GITHUB LINK

Project Code GitHub Link

https://github.com/mahalakshmiaddal/MultiPerspective-Fraud-Detection