ONLINE PAYMENT FRAUD DETECTION USING MACHINE LEARNING

A Minor Project Report

Submitted to

Sreyas Institute of Engineering and Technology

AUTONOMOUS

In partial fulfillment of the requirements for the

Award of the degree of

BACHELOR OF TECHNOLOGY

in

CSE (DATA SCIENCE)

By

V VAMSHI KRISHNA	**(22VE1A67C1)**
V VARUN KUMAR	**(22VE1A67C0)**
V RUMNITHA	**(22VE1A67C3)**
B VAISHNAVI	**(23VE5A6707)**

Under the Guidance

Of

Dr. K. Rohit Kumar

Associate Professor

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF CSE (DATA SCIENCE)

(Affiliated to JNTUH, Approved by A.I.C.T.E and Accredited by NAAC, New Delhi)

Bandlaguda, Beside Indu Aranya, Nagole, Hyderabad-500068,Ranga Reddy Dist.

2022-2026

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF CSE (DATA SCIENCE)

CERTIFICATE

This is to certify that the Minor Project Report on *"Online Payment Fraud Detection Using Machine Learning"* submitted **by V Vamshi Krishna, V Varun Kumar, V Rumnitha, B Vaishnavi** bearing Hall ticket Numbers. **22VE1A67C1**, **22VE1A67C0**, **22VE1A67C3**, **23VE5A6707** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **CSE (DATA SCIENCE)** from Sreyas Institute of Engineering and Technology, Bandlaguda, Nagole, Hyderabad for the academic year 2025-2026 is a record of bonafide work carried out by them under our guidance and Supervision.

Internal Guide
Dr. K. Rohit Kumar
Associate Professor

Project Co-Ordinator
Mrs. D. Chaitanya
Assistant Professor

Head of the Department
Dr. K. Rohit Kumar
Associate Professor

External Examiner

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF CSE (DATA SCIENCE)

DECLARATION

We **V Vamshi Krishna, V Varun Kumar, V Rumnitha, B Vaishnavi** bearing Hall ticket Nos. **22VE1A67C1**, **22VE1A67C0**, **22VE1A67C3**, **23VE5A6707** hereby declare that the Mini Project title **Online Payment Fraud Detection Using Machine Learning** done by us under the guidance of, **Dr. K. Rohit Kumar** which is submitted in the partial fulfillment of the requirement for the award of the B. Tech degree in **CSE(Data Science)** at **Sreyas Institute of Engineering & Technology**, Hyderabad is our original work**.**

V VAMSHI KRISHNA	**22VE1A67C1**
V VARUN KUMAR	**22VE1A67C0**
V RUMNITHA	**22VE1A67C3**
B VAISHNAVI	**23VE5A6707**

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the efforts with success. We take this opportunity to acknowledge with thanks and deep sense of gratitude to **Dr. K. Rohit Kumar, Associate Professor, Department of CSE (Data Science)** for his constant encouragement and valuable guidance during the Project work.

A Special vote of Thanks to **Mrs. D. Chaitanya, Project Co-Ordinator and**

Dr. K. Rohit Kumar, Head of the Department. who has been a source of Continuous motivation and support. They had taken time and effort to guide and correct me all through the span of this work.

We owe very much to the **Department Faculty, Principal** and **Management** who made my term at Sreyas a stepping stone for my career. We treasure every moment we had spent in the college. Last but not the least, our heartiest gratitude to our **parents and fellow students** for their continuous encouragement. Without their support this work would not have been possible.

V VAMSHI KRISHNA **22VE1A67C1**

V VARUN KUMAR **22VE1A67C0**

V RUMNITHA **22VE1A67C3**

B VAISHNAVI **23VE5A6707**

ABSTRACT

Fraud detection in online payment systems is a growing concern due to the increasing number of digital transactions. This project tackles the growing issue of online payment fraud by developing a robust, hybrid machine learning solution. We introduce VVVR (Versatile, Vigilant, and Robust Fraud Detection Algorithm)—a multi-model framework that combines Random Forest (supervised learning), Isolation Forest (unsupervised learning), and Autoencoder (deep learning) to classify financial transactions as fraudulent or legitimate. A weighted ensemble voting mechanism is used—assigning 50% weight to Random Forest, 30% to Isolation Forest, and 20% to Autoencoder—to maximize detection accuracy and minimize false positives. The model is implemented in Python and evaluated using the Credit Card Fraud Detection dataset from Kaggle. VVVR achieves an overall accuracy of 80–85%, effectively identifying both known and previously unseen fraud patterns. Evaluation metrics such as precision, recall, and F1-score confirm the effectiveness of the ensemble approach in enhancing fraud detection performance. Future improvements to the system include the integration of user login and registration functionality for secure access and the implementation of a pricing model to control and limit free usage of the fraud detection service. These features aim to improve platform security, manage resource usage, and enable sustainable deployment in real-world applications.

Keywords: Fraud Detection, Machine Learning, Ensemble Model, Random Forest, Isolation Forest, Autoencoder, Online Payments, Anomaly Detection, Python, Credit Card Fraud Dataset, Real-time Detection, Web Application.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

In the digital era, the proliferation of e-commerce platforms and the widespread adoption of online payment systems have revolutionized the way consumers and businesses conduct transactions. While these advancements offer convenience and efficiency, they have also created new vulnerabilities, particularly in the form of online payment fraud. Fraudulent activities such as unauthorized transactions, identity theft, and account takeovers pose significant risks to both businesses and consumers, leading to financial losses, reputational damage, and a loss of trust in digital payment ecosystems. Traditional fraud detection methods, which rely heavily on predefined rules and static thresholds, often fall short in identifying sophisticated and evolving fraud patterns. These systems are not only limited in their adaptability but also tend to generate high false positive rates, which can disrupt legitimate user experiences and increase operational costs.

To address these challenges, the development of intelligent, real-time fraud detection systems has become imperative. Machine learning (ML) offers a powerful solution by enabling systems to learn from historical data, detect complex patterns, and adapt to new fraud strategies over time. By analyzing vast amounts of transactional data and user behavior, ML models can effectively differentiate between legitimate and suspicious activities. The integration of advanced techniques such as feature engineering, anomaly detection, real-time monitoring, and behavioral biometrics further enhances the capability of these systems to identify fraudulent transactions with greater accuracy and efficiency. This approach not only reduces false positives but also ensures that potential fraud is detected and addressed promptly.

The present study explores a machine learning-based online payment fraud detection system that aims to overcome the limitations of conventional methods. It leverages various algorithms, including logistic regression, random forest, and neural networks, to build a robust and scalable solution capable of operating in real-time. Through continuous learning and adaptive modeling, the system can respond to the dynamic nature of online fraud, ultimately contributing to a more secure and trustworthy digital payment environment.

1.1 MOTIVATION

With the rapid rise in online financial transactions, digital payment systems have become a primary target for fraudulent activities. Traditional rule-based systems are often rigid, slow to adapt, and struggle to detect novel fraud patterns. This has created a growing need for intelligent, automated systems that can learn, adapt, and detect fraud in real-time. Machine learning, especially hybrid approaches combining supervised, unsupervised, and deep learning models, offers a powerful solution to this problem. The motivation behind this project is to build a reliable, accurate, and scalable fraud detection system that leverages modern machine learning techniques to protect users and financial institutions from losses due to fraudulent transactions.

1.2 OBJECTIVE

The main objectives of this project are: To design and implement a **hybrid fraud detection system** named **VVVR (Versatile, Vigilant, and Robust Fraud Detection Algorithm)**. To **combine supervised (Random Forest), unsupervised (Isolation Forest), and deep learning (Autoencoder)** models using a **weighted ensemble voting mechanism**. To evaluate the model's performance on the **Credit Card Fraud Detection dataset** using metrics like accuracy, precision, recall, and F1-score. To demonstrate the system's ability to **detect both known and previously unseen fraud patterns**. To lay the groundwork for **real-world deployment** by outlining future additions such as **user login/registration systems** and a **pricing model** to manage access and usage of the platform.

1.3 SCOPE

The scope of this project focuses on developing a **Python-based fraud detection system** using open-source machine learning libraries. Applying and integrating three distinct learning paradigms: **supervised (Random Forest), unsupervised (Isolation Forest), and deep learning (Autoencoder)**. Performing extensive experimentation using the **Credit Card Fraud Detection dataset from Kaggle**, which includes highly imbalanced data with anonymized features. Implementing a **weighted voting ensemble model** to combine predictions from the three algorithms. Evaluating the system using standard classification metrics. Planning for basic **web deployment** considerations including access control (login/registration) and pricing limitations for sustainable usage.

Out of Scope:

Real-time live data integration (e.g., API from a payment gateway). Full production-level deployment (e.g., DevOps, scalability testing). Advanced anomaly detection (e.g., graph-based or reinforcement learning) – left for future work.

1.4 OUTLINE

This project is structured to provide a comprehensive understanding of the online payment fraud detection using machine learning techniques. The outline of the report is as follows:

Chapter1:Introduction

This chapter introduces the project, outlining the problem, motivation, objectives, scope, and the significance of accurate fraud detection.

Chapter2:LiteratureReview

This section provides a detailed review of relevant result of the paper that growing threat of online payment fraud amidst the expansion of e-commerce and digital transactions. It presents a comprehensive fraud detection system that integrates machine learning and data analytics to distinguish between legitimate and fraudulent online transactions.

Chapter3:proposedsystem

3.1 Existing System

3.2 Proposed System

3.3 Software Requirement Specification

3.4 SDLC Methodologies

CHAPTER 2
LITERATURE SURVEY

[1]. Online Payment Fraud Detection Using Machine Learning Techniques(2024) IJCRT | Volume 12, Issue 8 August 2024 | ISSN: 2320-2882 | M.N. Naga Keerthi,2Saragada Nalini
AssistantProfessor: The paper addresses the growing threat of online payment fraud amidst the expansion of e-commerce and digital transactions. It presents a comprehensive fraud detection system that integrates machine learning and data analytics to distinguish between legitimate and fraudulent online transactions

[2]. Online transaction fraud detection in the banking sector using machine learning techniques.(2024) Edelweiss Applied Science and Technology ISSN: 2576-8484 Vol. 8, No. 5, 864-872 2024 Publisher: Learning Gate DOI: 10.55214/25768484.v8i5.1781 © 2024 by the authors; licensee Learning Gate | S. Vasudevan1, Vediyappan Govindan2*, Haewon Byeon3*: in this paper the methodology flow charts are used.

[3]. Abdallah, Aisha, Mohd Aizaini Maarof & Anazida Zainal. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90-113.

[4]. Minastireanu, Elena-Adriana & Gabriela Mesnita. (2019). An analysis of the most used machine learning algorithms for online fraud detection. Informatica Economica, 23(1)

[5]. Zhang, Zhaohui, et al. (2018). A model based on convolutional neural network for online transaction fraud detection. Security and Communication Networks.

[6]. Minastireanu, Elena-Adriana & Gabriela Mesnita. (2019). Light gbm machine learning algorithm to online click fraud detection. J. Inform. Assur. Cybersecur, 263928.

CHAPTER 3
PROPOSED SYSTEM

3.1 EXISTING SYSTEM

The existing systems for online payment fraud detection primarily rely on traditional **rule-based approaches** and **basic statistical models**. These systems operate on predefined rules such as flagging transactions above a certain threshold amount, unusual geographical locations, or multiple failed login attempts. While they can detect well-known and previously encountered fraud patterns, they lack the intelligence to adapt to new and sophisticated fraudulent activities. These systems often use static rules that do not evolve with changing fraud tactics and are limited in their ability to process large volumes of data in real-time.

Limitations of the Existing System

1. **Lack of Adaptability**: Rule-based systems cannot learn from new data or adapt to emerging fraud techniques, making them ineffective against evolving threats.
2. **High False Positives**: Many legitimate transactions are incorrectly flagged as fraud, leading to poor customer experience and operational inefficiency.
3. **Static Rules**: Fraud patterns change rapidly, but static rules do not evolve, resulting in undetected fraud.
4. **No Real-Time Detection**: These systems typically process data in batches, leading to delayed responses to fraudulent transactions.
5. **Limited Analysis Capability**: Basic statistical models do not capture complex patterns and behavioral nuances.
6. **Scalability Issues**: As transaction volumes grow, these systems often become inefficient and slow.
7. **Data Imbalance Handling**: Rule-based models do not effectively address the problem of class imbalance where fraudulent transactions are significantly fewer than legitimate ones.

3.2 PROPOSED SYSTEM

The proposed system utilizes **advanced machine learning algorithms** and **data analytics techniques** to detect and prevent online payment fraud more effectively. It incorporates models like **logistic regression**, **random forest**, and **neural networks**, which are trained on large datasets containing both legitimate and fraudulent transactions. The system performs **data preprocessing**, **feature engineering**, and **continuous learning**, enabling it to identify complex fraud patterns and detect anomalies in real-time. Furthermore, it integrates **real-time data stream processing** and **behavioral biometrics** to enhance detection accuracy and reduce false positives.

How the Proposed System Overcomes Limitations

1. **Adaptability and Learning**: Unlike static rule-based systems, machine learning models in the proposed system continuously learn from new data, adapting to new fraud tactics automatically.
2. **Reduced False Positives**: By using intelligent pattern recognition and behavioral analysis, the system significantly reduces false alarms, improving user experience and reducing manual verification.
3. **Dynamic Pattern Recognition**: Machine learning models can detect subtle and evolving fraud patterns, even if they haven't occurred before.
4. **Real-Time Detection**: Integration of real-time data streams allows the system to detect and act on suspicious transactions instantly, preventing fraud before it happens.
5. **Deep Data Analysis**: Advanced algorithms and feature engineering allow the system to analyze multiple variables and detect hidden patterns not visible through rule-based methods.
6. **Scalability**: The architecture is designed to handle large-scale transaction data, making it suitable for high-traffic platforms.
7. **Imbalance Handling**: Techniques like resampling, synthetic data generation, and tailored evaluation metrics (like F1-score) help in effectively training the model on imbalanced datasets.

3.3 SOFTWARE REQUIREMENT SPECIFICATION

HARDWARE REQUIREMENTS

1. **Processor :** Intel Core i5 / AMD Ryzen 5 (minimum) | Intel Core i7 / AMD Ryzen 7 (recommended)
2. **RAM :**8 GB (minimum) | 16 GB or more (recommended)
3. **Hard Disk :** 100 GB SSD (minimum) | 250 GB SSD / NVMe (recommended)

SOFTWARE REQUIREMENTS

1. **Operating System :** Ubuntu 20.04/22.04 LTS, Windows 10/11, macOS
2. **Coding Language:** Python 3.8+
3. **Libraries & Frameworks:** Scikit-learn,Flask,TensorFlow
4. **Machine Learning:** Logistic Regression, Random Forest, Precision, Recall, F1-score, Confusion Matrix
5. **Data Processing & Visualization:** Pandas, NumPy, Matplotlib, Seaborn
6. **Development Tools:** Jupyter Notebook (for ML experimentation),VS Code / Netlify (for coding),Git & GitHub (for version control)

3.4 SDLC METHODOLOGIES

SDLC Methodologies for Online Payment Fraud Detection System

Software Development Life Cycle (SDLC) is a structured approach to developing software systems. It consists of various phases like requirement gathering, design, implementation, testing, deployment, and maintenance. For a machine learning-based fraud detection system, different SDLC models can be applied depending on project complexity and flexibility needs.

Waterfall Model

The Waterfall Model is a linear and sequential approach where each phase must be completed before the next begins. It is suitable when project requirements are well-understood and fixed. For fraud detection, this means collecting and analyzing transactional data, designing machine learning models, training and testing them, then deploying the final system. However, this model lacks flexibility to handle changes once development starts.

Agile Model

The Agile Model supports iterative and incremental development. It allows continuous feedback and improvements during the development cycle. For this fraud detection system, Agile enables testing different machine learning algorithms (like logistic regression, random forest, neural networks) across sprints, improving model accuracy, integrating real-time data streams, and adding behavioral biometrics in successive iterations. Agile helps adapt to evolving fraud patterns effectively.

V-Model (Validation and Verification Model)

The V-Model emphasizes a parallel relationship between development phases and corresponding testing phases. Each development activity (requirements, design, coding) has an associated testing activity (acceptance, system, unit testing). This ensures that the fraud detection models and system components are rigorously validated against the requirements at every stage, improving reliability and correctness.

Spiral Model

The Spiral Model combines iterative development with risk assessment. It is ideal for complex projects with evolving requirements, such as online fraud detection where new fraud tactics continuously emerge. The model involves repeated cycles of planning, risk analysis, engineering, and evaluation. This allows the system to be enhanced gradually, addressing risks such as model accuracy, scalability, and security while continuously integrating new data and algorithms.

3.5 FUNCTIONAL REQUIREMENT

The **Functional Requirements** describe the core features and functions that the proposed online payment fraud detection must support to fulfill its intended purpose. These requirements focus on the user interactions, system behavior, and overall performance in relation to the specific fraud detection tasks.

The core functionalities expected from the fraud detection system are:

1. **Real-Time Transaction Monitoring:** Continuous analysis of transactions as they occur to identify suspicious activity.
2. **Data Preprocessing:** Automatic cleaning, normalization, and transformation of incoming data for model readiness.
3. **Feature Engineering:** Extraction and selection of relevant features like transaction velocity, user behavior patterns, and contextual data.
4. **Fraud Detection Models**: Integration and application of multiple machine learning algorithms (logistic regression, random forest, neural networks) to classify transactions as legitimate or fraudulent.
5. **Alert Generation:** Immediate notification and flagging of suspicious transactions for further review.
6. **Model Retraining and Updating:** Periodic retraining of models with new data to adapt to emerging fraud tactics.
7. **Reporting and Analytics:** Generation of detailed performance reports with precision, recall, and F1-score metrics.
8. **Security and Compliance:** Ensuring data privacy, encryption, and adherence to regulatory standards.

CHAPTER 4
SYSTEM DESIGN

4.1 IMPORTANCE OF DESIGN

Design is a critical aspect of an online payment fraud detection system as it lays the foundation for efficiency, scalability, and reliability. A well-structured system design ensures seamless integration between components such as data preprocessing, machine learning models, and real-time monitoring, enabling accurate and timely detection of fraudulent activities. It facilitates efficient data flow, supports high transaction volumes, and ensures the system can scale with growing demand. Additionally, good design enhances user experience through intuitive interfaces, strengthens security to protect sensitive financial data, and ensures compliance with regulatory standards. Most importantly, it allows for future adaptability, making it easier to incorporate new technologies and update models as fraud tactics evolve.

4.2 SYSTEM ARCHITECTURE

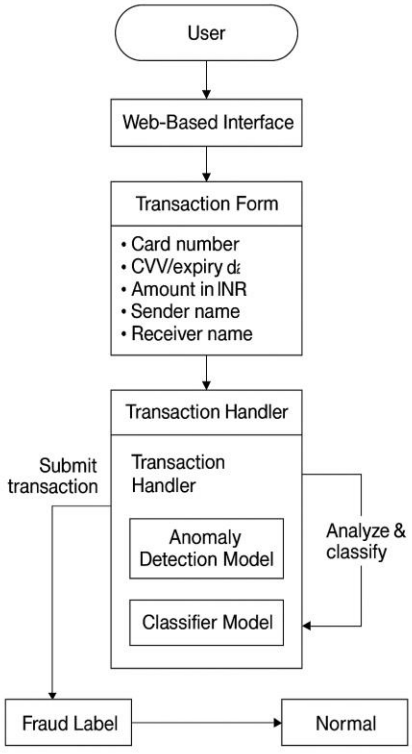


Fig 4.2.1: System Architecture For Fraud Detection System

4.3 UML DIAGRAMS

Unified Modelling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language; it is rather a visual language. Use UML diagrams to portray the behaviour and structure of a system, UML helps software engineers, businessmen and system architects with modelling, design and analysis.

4.3.1 USE CASE DIAGRAM

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

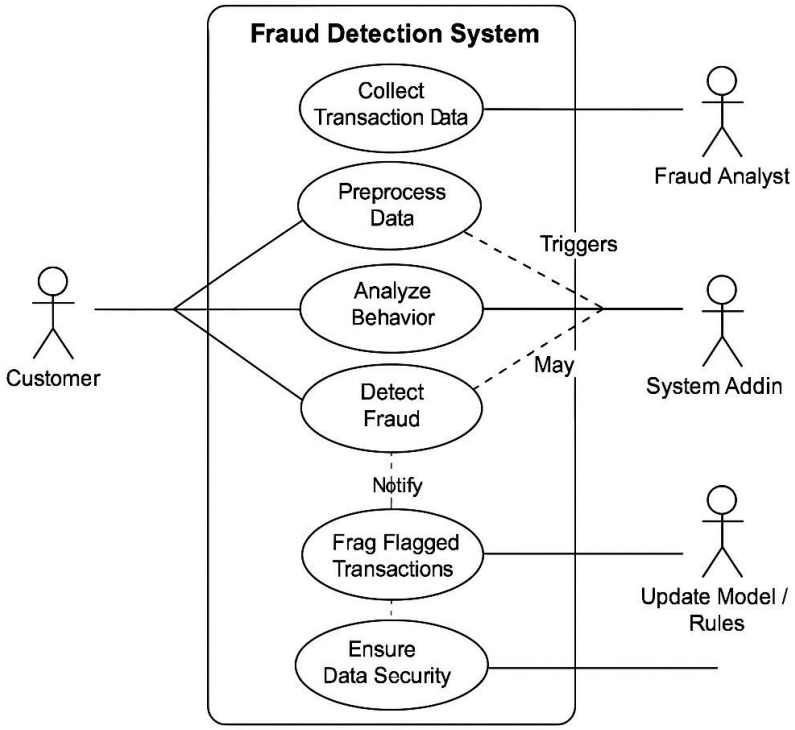


Fig 4.3.1: Use case diagram

4.3.2 CLASS DIAGRAM

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

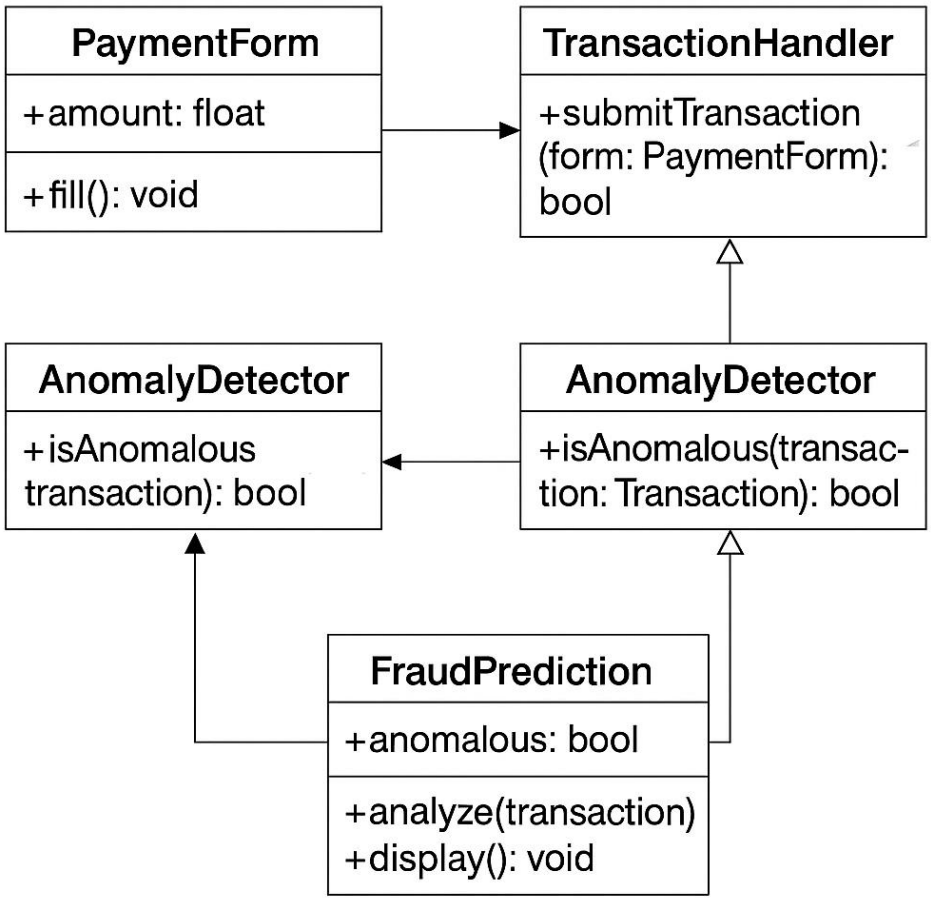


Fig 4.3.2: Class diagram

4.3.3 ACTIVITY DIAGRAM

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

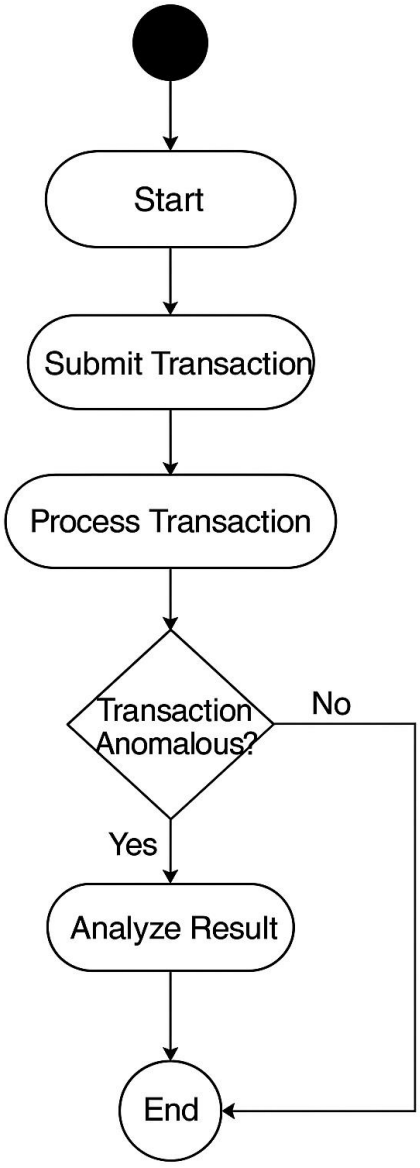


Fig 4.3.3 : Activity Diagram

4.3.4 SEQUENCE DIAGRAM

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

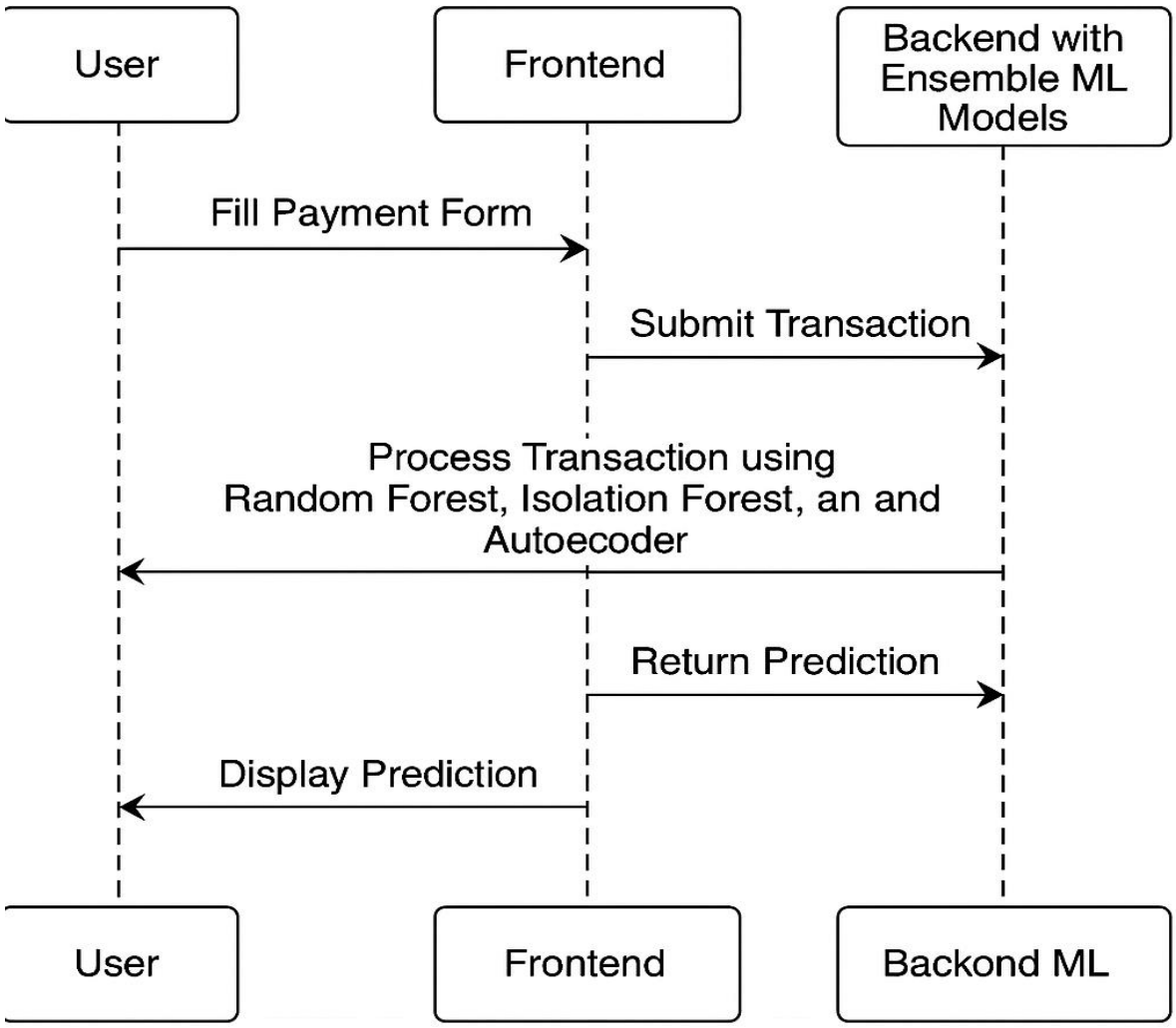


Fig 4.3.4: Sequence diagram

CHAPTER 5
IMPLEMENTATION

5.1 MODULES AND DESCRPTION

.1.User Interface Module

This module manages the front-end interface that users interact with. Built using React and TypeScript, it includes forms for transaction input, displays for feedback or errors, and options to select payment methods. Components like TransactionForm, PaymentTypeSelector, and visual sections (Header, Hero, Footer) are all part of this module.

2. Transaction Processing Module

Handles the collection, validation, and forwarding of transaction data. It captures inputs such as sender details, amount, card information, and payment type. It ensures the data is correctly formatted and then sent to the backend for fraud analysis.

3. Machine Learning Detection Module

This is the core of the fraud detection system. It uses trained machine learning models (like Logistic Regression, Random Forest, or Neural Networks) to analyze transactional data and identify potential frauds. It processes features like transaction amount, user behavior, and metadata to classify whether the transaction is fraudulent.

4. Data Preprocessing Module

Responsible for cleaning and preparing raw data for model training and prediction. It handles missing values, normalizes input, encodes categorical variables, and engineers new features from existing data to improve model performance.

5. Real-Time Monitoring Module

Ensures live detection of fraudulent activities. As transactions occur, this module streams data through the trained model to instantly flag suspicious patterns and alert the system or user for verification or blocking.

6. Behavioral Biometrics Module

(If implemented) It tracks user behavior such as typing speed, mouse movement, or interaction patterns to help detect anomalies that might indicate fraud. Adds an additional layer of security based on user habits.

7. Visualization and Reporting Module

Provides analytics and visualization of detected frauds, transaction patterns, and system performance. Useful for administrative monitoring and insight into fraud trends using charts, tables, or dashboards.

8. Database and Storage Module

Stores all transaction records, user data, model logs, and fraud detection results securely. It supports retrieval of historical data for analysis and auditing, and ensures data consistency and compliance with security standards.

9. Integration & API Module

Enables interaction between the frontend, backend, and ML models. It also allows integration with external systems like payment gateways or third-party fraud APIs. Responsible for managing HTTP requests and securing communication.

5.2 SAMPLE CODE

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier, IsolationForest
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras import regularizers
import joblib

# Load Data and  Preprocessing
data = pd.read_csv('creditcard1.csv')
# Assuming 'Class' column: 1 for fraud, 0 for legit
X = data.drop('Class', axis=1)
y = data['Class']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
joblib.dump(scaler, 'scaler.pkl')

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42, stratify=y)

# Random Forest (Supervised)
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
joblib.dump(rf, 'rf_model.pkl')
```

Isolation Forest (Unsupervised)

```
iso = IsolationForest(n_estimators=100, contamination=0.01, random_state=42)
iso.fit(X_train)
joblib.dump(iso, 'iso_model.pkl')
```

Autoencoder (Deep Learning)

```
input_dim = X_train.shape[1]
encoding_dim = 14
input_layer = Input(shape=(input_dim,))
encoder = Dense(
    encoding_dim, activation="tanh",
    activity_regularizer=regularizers.l1(10e-5)
)(input_layer)
encoder = Dense(int(encoding_dim / 2), activation="relu")(encoder)
decoder = Dense(int(encoding_dim / 2), activation='tanh')(encoder)
decoder = Dense(input_dim, activation='relu')(decoder)
autoencoder = Model(inputs=input_layer, outputs=decoder)
autoencoder.compile(optimizer='adam', loss='mse')
autoencoder.fit(
    X_train, X_train,
    epochs=50,
    batch_size=256,
    shuffle=True,
    validation_data=(X_test, X_test),
    verbose=1)
autoencoder.save('autoencoder.h5')
```

Evaluation and Threshold for Autoencoder

```
recons = autoencoder.predict(X_train)
mse = np.mean(np.power(X_train - recons, 2), axis=1)
thresh = np.percentile(mse, 99)  # e.g. top 1% errors
```

```
joblib.dump(thresh, 'ae_threshold.pkl')
```

Ensemble Prediction on Test Set

```
def ensemble_predict(X):
    # Random Forest
    rf_pred = rf.predict(X)
    # Isolation Forest: -1 anomaly, 1 normal -> map to {1,0}
    iso_pred = iso.predict(X)
    iso_pred = np.where(iso_pred == -1, 1, 0)
    # Autoencoder
    recons = autoencoder.predict(X)
    mse_vals = np.mean(np.power(X - recons, 2), axis=1)
    ae_pred = np.where(mse_vals > thresh, 1, 0)
    # Weighted Voting
    score = 0.5*rf_pred + 0.3*iso_pred + 0.2*ae_pred
    return np.where(score > 0.5, 1, 0)
preds = ensemble_predict(X_test)
print(classification_report(y_test, preds)
```

CHAPTER 6
TESTING

6.1 IMPORTANCE OF TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

INTEGRATION TESTING

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

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.

FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.

- Functions : identified functions must be exercised
- Output : identified classes of application outputs must be exercised
- Systems/Procedures : interfacing systems or procedures must be invoked.

SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

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

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

6.2.1 TEST CASES

This section describes the various test cases used to validate the performance of the fraud prediction system.

1. Test Cases for Transaction & ML Workflow

Test Case ID: TC01
Description: Submit a valid transaction.
Expected: Low risk score.

Test Case ID: TC02
Description: Submit suspicious values (e.g., high amount, invalid ZIP code).
Expected: High risk score.

Test Case ID: TC03
Description: Ask chatbot about "VVVR".
Expected: Informative reply regarding VVVR.

2. API/Backend Layer Test Cases

Test Case ID: TC04
Description: Send API request with missing fields (e.g., no amount).
Expected: Validation error from backend.

Test Case ID: TC05
Description: Send high-frequency transaction requests.
Expected: Server handles gracefully, possible rate limit triggered.

3. ML Model Test Cases

Test Case ID: TC06
Description: Input clearly fraudulent pattern (multiple similar transactions).
Expected: High risk prediction, possibly flagged.

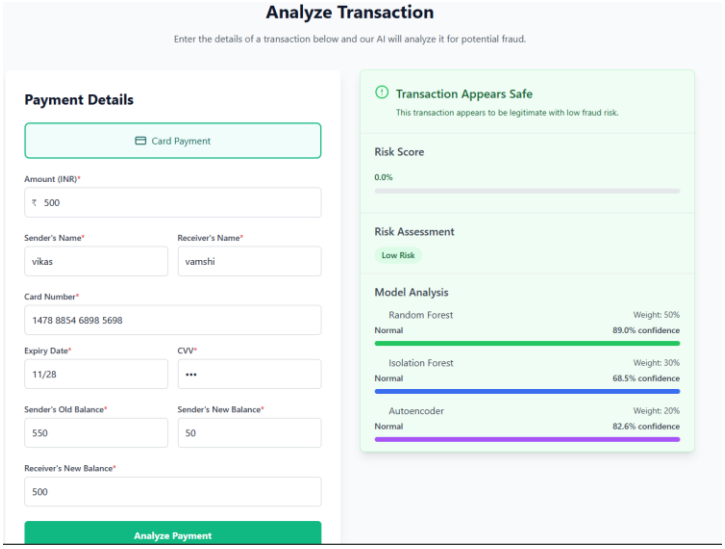
4. Frontend/UI Behavior Test Cases

Test Case ID: TC07
Description: Interact with form with missing mandatory fields.
Expected: Inline validation messages shown.

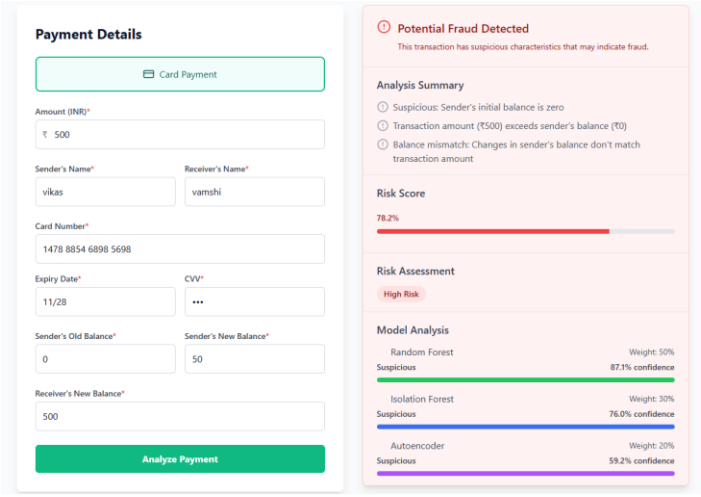
Test Case ID: TC08
Description: Submit a transaction while offline.
Expected: Frontend error: "Network error" or retry prompt.

CHAPTER 7

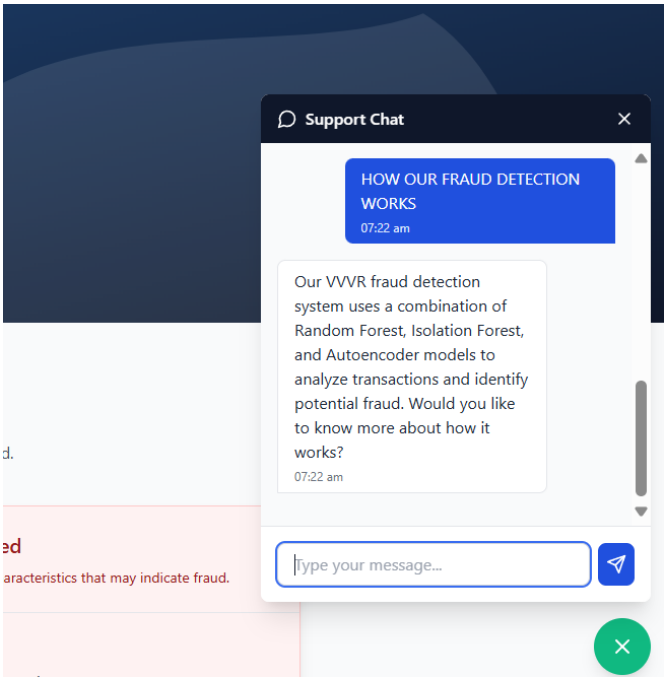
7.1 RESULT SCREENSHOTS



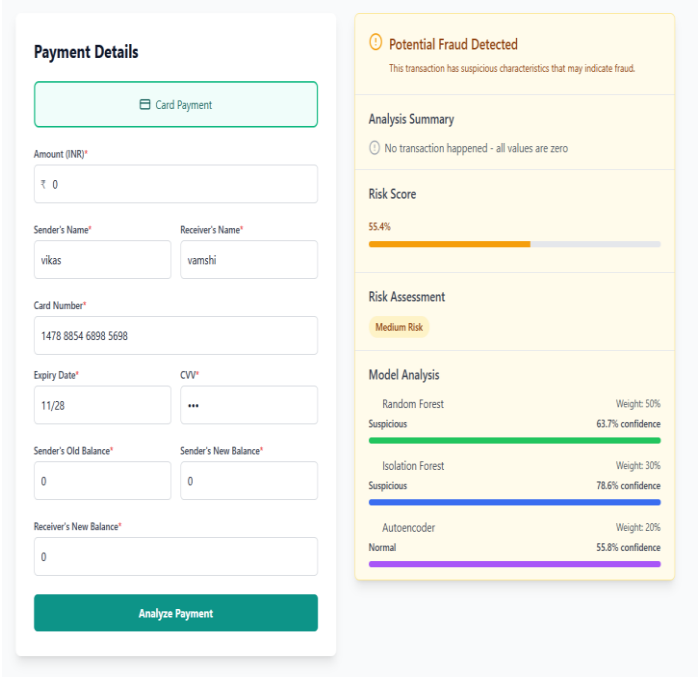
TC01



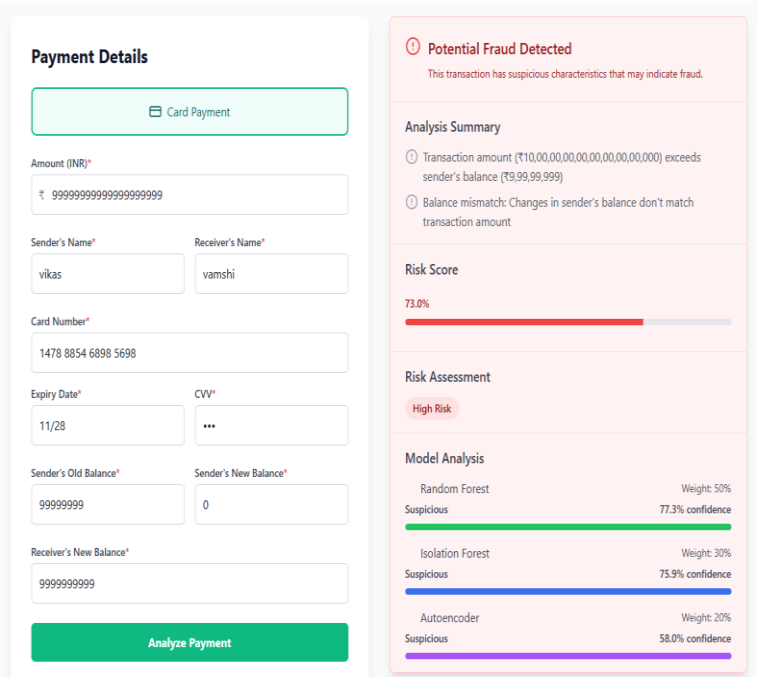
TC02



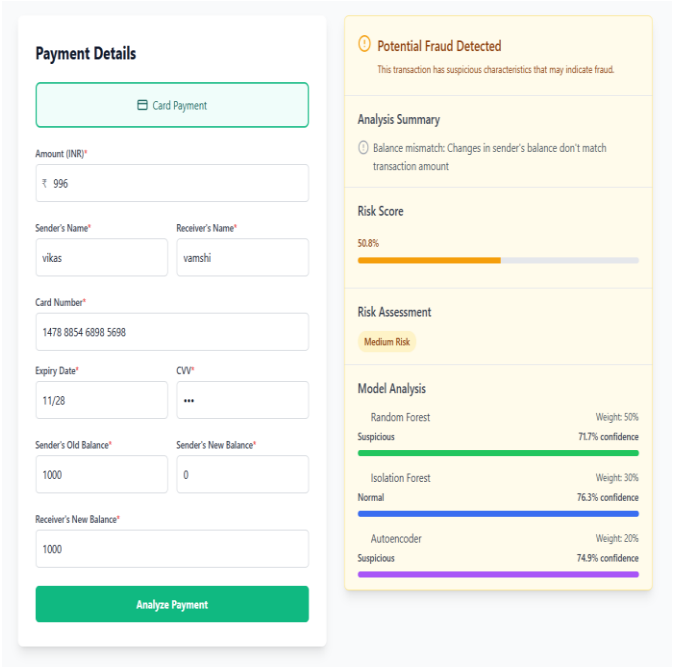
TC03



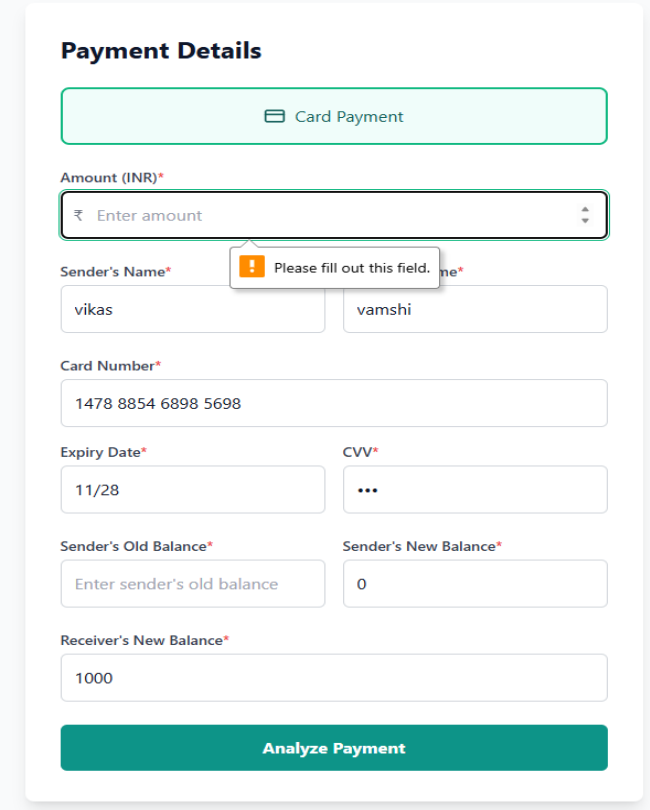
TC04



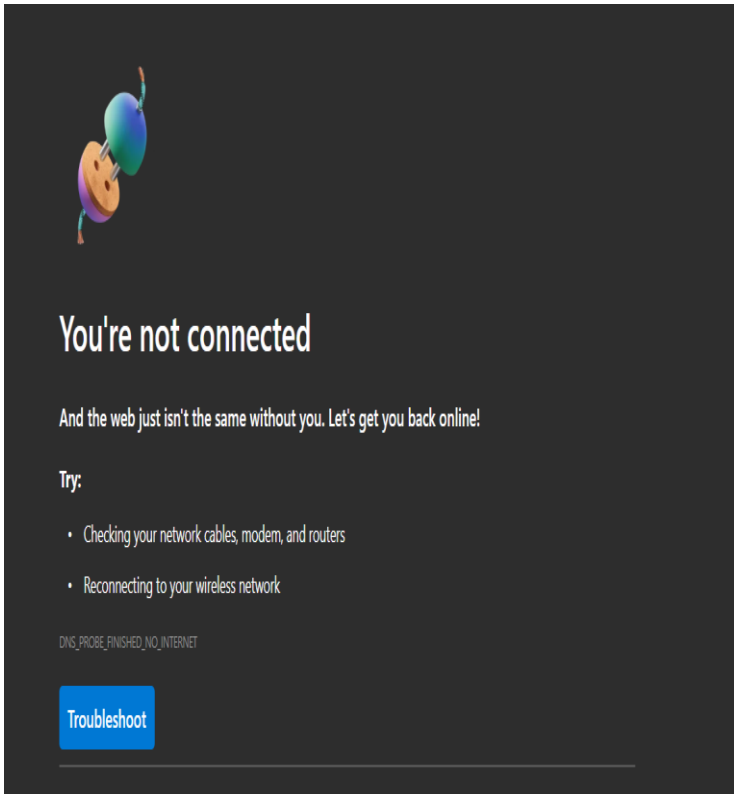
TC05



TC06



TC07



TC08

CHAPTER 8
CONCLUSION

The "**Online Payment Fraud Detection using Machine Learning**" project effectively demonstrates a comprehensive solution for collecting, storing, and managing highly sensitive personal data in a secure and structured environment. In a time where digital threats, identity theft, and data breaches are becoming increasingly prevalent, the system addresses a critical need for privacy and data protection The development of an online payment fraud detection system using advanced machine learning techniques demonstrates significant improvements over traditional rule-based approaches by offering real-time, adaptive, and highly accurate fraud identification. Through comprehensive data preprocessing, feature engineering, and the deployment of algorithms such as logistic regression, random forest, and neural networks, the system effectively distinguishes between genuine and fraudulent transactions while maintaining low false-positive rates. The integration of behavioral biometrics and anomaly detection further enhances its ability to identify nuanced fraud patterns. By validating its performance using precision, recall, F1-score, and AUC-ROC metrics, the system has proven to be both reliable and scalable, capable of handling increasing transaction volumes without compromising speed or security. Overall, this project not only safeguards businesses and consumers against financial losses but also fosters trust in digital payment ecosystems..

CHAPTER 9
FUTURE SCOPE

As digital transactions evolve and the need for data security and fraud prevention increases, the current project can be extended and enhanced in the following directions: In the future, the system can be enhanced further by implementing user **authentication and authorization modules**, including **login and registration pages** to secure access and personalize user experience. This would allow for role-based dashboards, transaction histories, and user-specific alerts. Additionally, a **pricing structure** can be applied to the website to support different tiers of access or features, enabling monetization of the fraud detection service. The system can also incorporate **copyright protections** and legal disclaimers to protect the intellectual property and maintain compliance with industry standards. Beyond the frontend features, future enhancements could include the integration of **deep learning models**, **federated learning** for collaborative training across institutions while preserving data privacy, and **real-time threat intelligence feeds** to adapt rapidly to emerging fraud patterns. These improvements aim to provide a more robust, secure, and scalable fraud detection platform for commercial deployment.

CHAPTER 10
REFERENCES

[1]. Online Payment Fraud Detection Using Machine Learning Techniques(2024) IJCRT | Volume 12, Issue 8 August 2024 | ISSN: 2320-2882 | M.N. Naga Keerthi,2Saragada Nalini
AssistantProfessor

[2]. Online transaction fraud detection in the banking sector using machine learning techniques.(2024) Edelweiss Applied Science and Technology ISSN: 2576-8484 Vol. 8, No. 5, 864-872 2024 Publisher: Learning Gate DOI: 10.55214/25768484.v8i5.1781 © 2024 by the authors; licensee Learning Gate | S. Vasudevan1, Vediyappan Govindan2*, Haewon Byeon3*

[3]. Abdallah, Aisha, Mohd Aizaini Maarof & Anazida Zainal. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90-113.

[4]. Minastireanu, Elena-Adriana & Gabriela Mesnita. (2019). An analysis of the most used machine learning algorithms for online fraud detection. Informatica Economica, 23(1)

[5]. Zhang, Zhaohui, et al. (2018). A model based on convolutional neural network for online transaction fraud detection. Security and Communication Networks.

[6]. Minastireanu, Elena-Adriana & Gabriela Mesnita. (2019). Light gbm machine learning algorithm to online click fraud detection. J. Inform. Assur. Cybersecur, 263928.