PROJECT REPORT

TEAM ID: LTVIP2026TMIDS54062

TITLE: Online Payments Fraud Detection Using Machine Learning

1. INTRODUCTION

1.1 Project Overview

As a digital payments security analyst, my mission is to proactively safeguard online transactions and protect customers from financial fraud. However, the rapidly evolving landscape of cyber threats, sophisticated fraud techniques, and high transaction volumes makes fraud detection increasingly complex. Traditional rule-based systems often rely on static conditions and historical fraud patterns, which limit their ability to detect emerging threats in real time. Without intelligent, adaptive, and data-driven monitoring, fraudulent activities may go unnoticed until financial losses occur — placing institutions in a reactive rather than proactive position. To address this challenge, this project focuses on building a Machine Learning–based Online Payments Fraud Detection System that analyzes transaction behavior, user activity, and device-level information in real time to detect suspicious patterns and prevent fraudulent transactions before they are completed.

1.2 Purpose

The primary purpose of this project is to develop an intelligent, scalable, and real-time fraud detection platform using Machine Learning techniques that enables financial institutions and payment platforms to:

- Monitor and analyze online transactions in real time.
- Detect suspicious behavior using predictive analytics and anomaly detection.
- Reduce fraud losses while minimizing false positives.
- Improve customer trust by ensuring secure digital payment experiences.
- Enable data-driven fraud risk assessment and automated decision-making.
- Support regulatory compliance through explainable AI and audit logging.

By leveraging advanced analytics, behavioral modeling, and risk scoring mechanisms, this system transforms fraud detection from a reactive investigation process into a proactive fraud prevention strategy, ensuring safer and more reliable digital payment ecosystems.

2. IDEATION PHASE

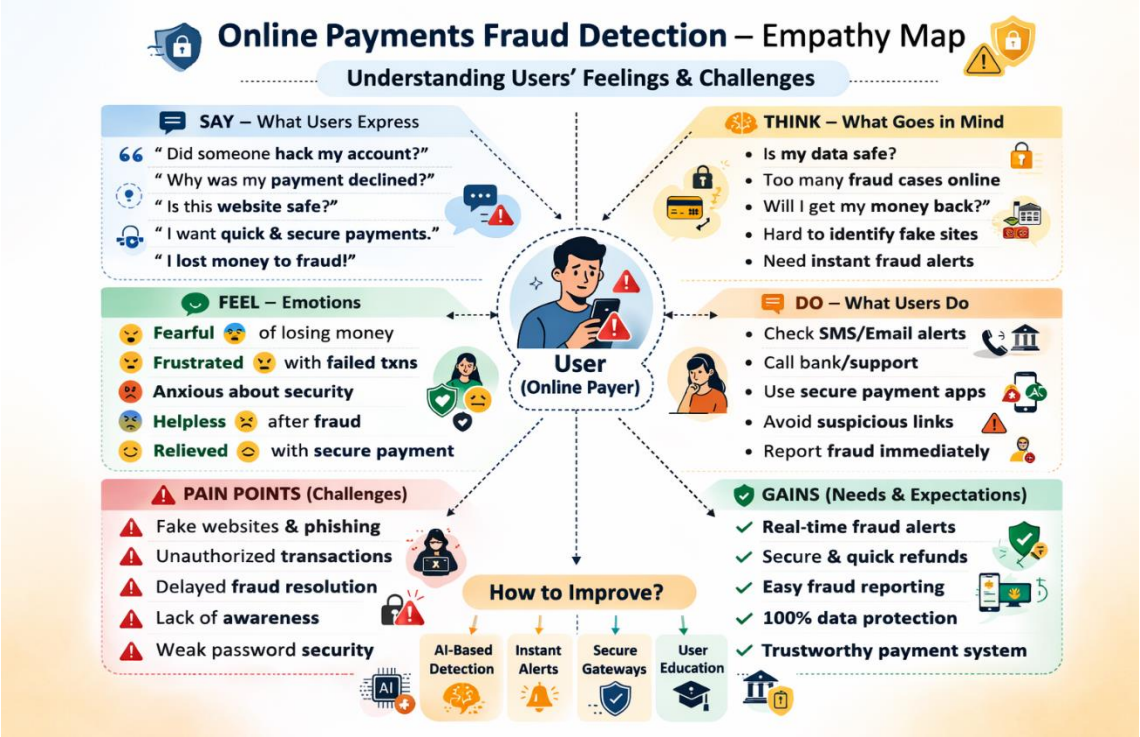
2.1 Problem Statement

I am a digital banking customer
I'm trying to make secure online payments without worrying about fraud
But I often experience either fraudulent transactions or unnecessary transaction declines
Because fraud detection systems are either not accurate enough or generate too many false alerts
Which makes me feel anxious about my financial security and frustrated with online payment services
PS-1: I am a bank fraud analyst who wants to detect fraudulent online transactions in real-time, but I don't have an intelligent system that can accurately differentiate between genuine and fraudulent payment patterns.

PS-2: I am a financial institution manager trying to reduce fraud-related losses, but traditional rule-based systems fail to adapt to evolving fraud techniques and generate high false positive rates.

PS-3: I am a cybersecurity specialist aiming to strengthen digital payment security, but there is no integrated machine learning framework that continuously learns from transaction data and predicts emerging fraud patterns effectively.

2.2 Empathy Map



2.3 Brainstorming

Step-1: Team Gathering, Collaboration and Problem Statement Selection

The team collaboratively reviewed online transaction datasets and analyzed current challenges in digital payment systems, including credit card fraud, UPI fraud, phishing-based transactions, and false transaction declines.

After discussing fraud trends, customer pain points, and banking security requirements, the following problem statement was selected:

"There is a need to design and implement a real-time machine learning-based fraud detection system that accurately identifies fraudulent online payment transactions while minimizing false positives and ensuring a seamless customer experience."

Step-2: Brainstorm, Idea Listing and Grouping

Grouping Category	Ideas Generated
Transaction Pattern Analysis	Analyze transaction amount, time, frequency, and location
Behavioral Analysis	Detect unusual spending behavior and new device usage
Machine Learning Models	Implement Random Forest, XGBoost, Logistic Regression
Risk Scoring System	Assign fraud probability score to each transaction

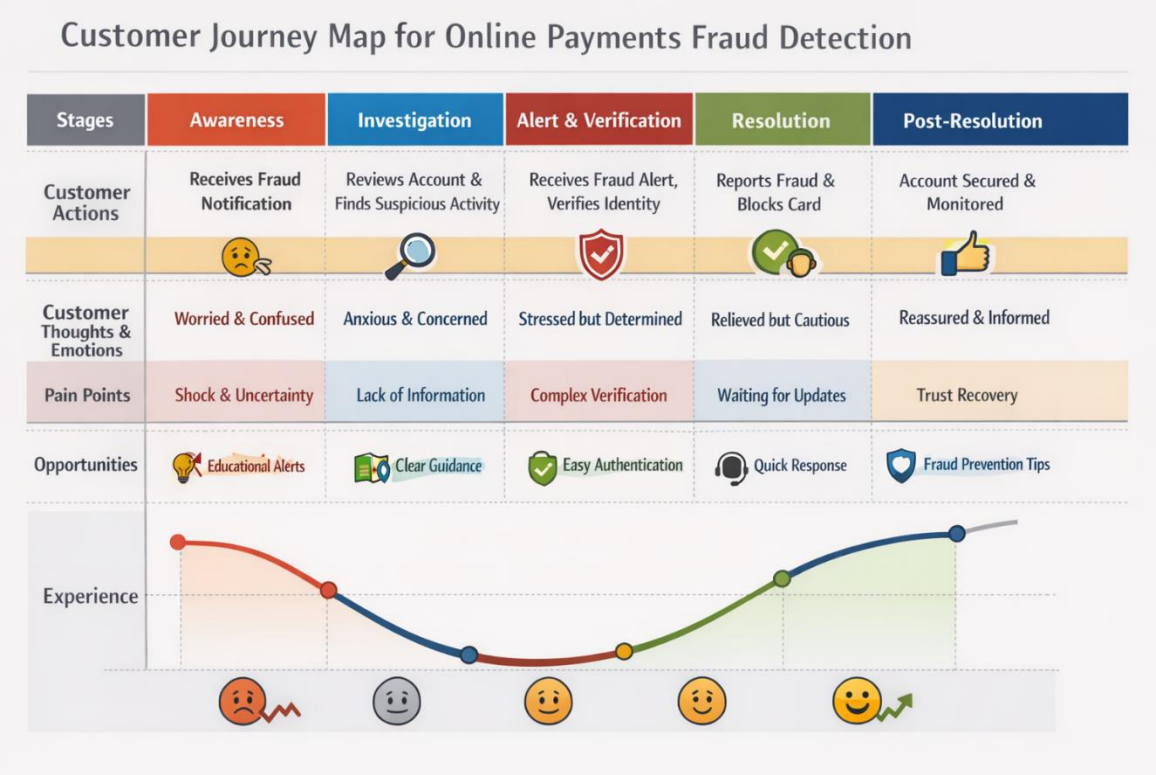
Real-Time Monitoring	Develop API-based real-time fraud detection
Alert Mechanism	Trigger SMS/Email alerts for suspicious transactions
Explainable AI	Use SHAP to explain why a transaction was flagged
Fraud Analyst Dashboard	Build dashboard to monitor flagged transactions
Security Enhancement	Enable auto card freeze and OTP verification
Future Enhancements	Integrate deep learning models and behavioral biometrics

Step-3: Idea Prioritization

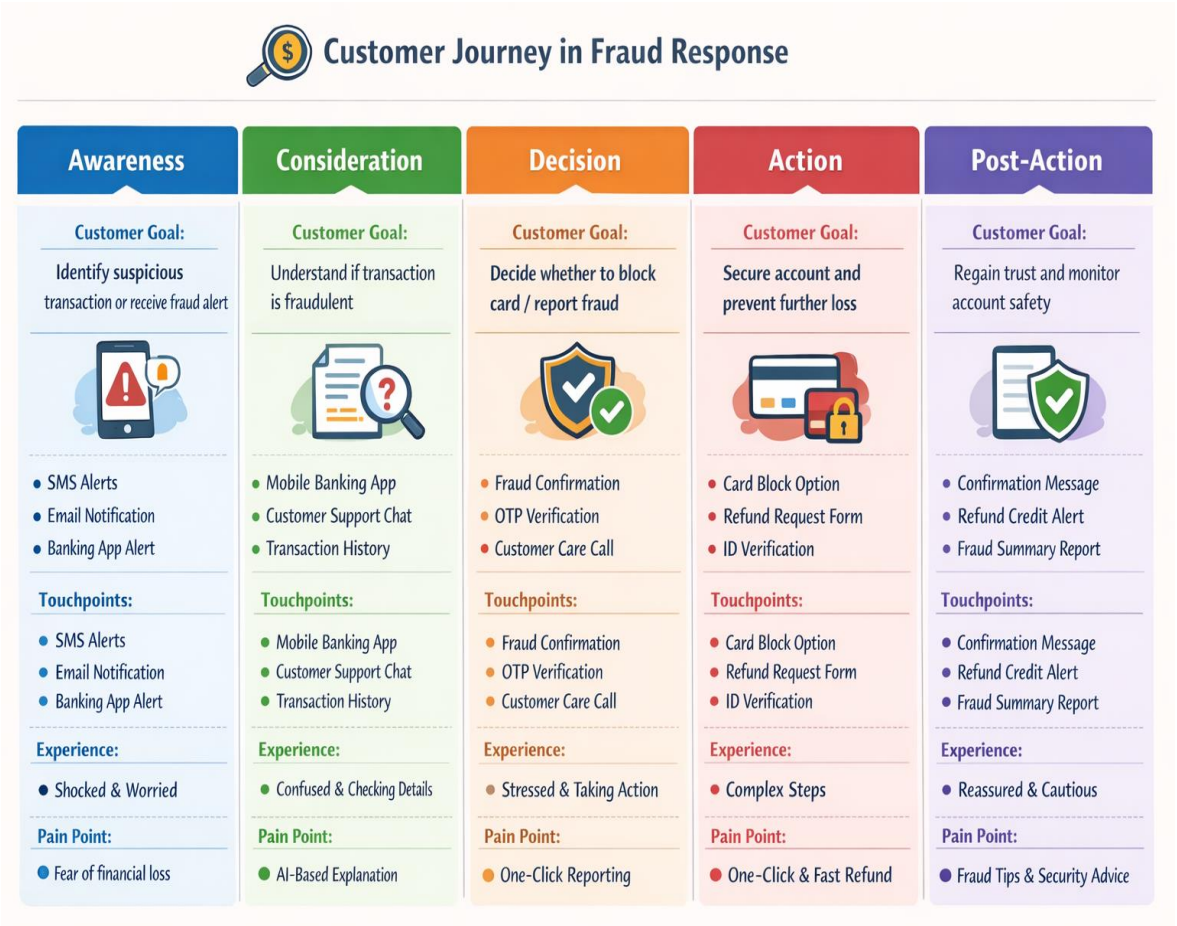
Priority Level	Idea
High	Develop ML model (Random Forest/XGBoost) for fraud detection
High	Implement real-time risk scoring and alert system
High	Apply feature engineering for behavioral deviation detection
Medium	Create fraud analyst monitoring dashboard
Medium	Integrate SHAP for model explainability
Low	Implement deep learning-based fraud detection
Low	Integrate advanced behavioral biometrics and blockchain security

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map



[Customer Persona:Digital Banking Customer / Online Payment User]



Key Insights:

- Customers expect **real-time fraud detection and instant alerts**.
- Simple and clear communication reduces panic during fraud incidents.
- Easy verification processes increase customer trust.
- Faster refund processing improves overall customer satisfaction.
- Continuous monitoring and fraud education build long-term loyalty.

3.2 Solution Requirements
Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Authentication	Login through Mobile Banking App Login through Internet Banking Biometric Authentication
FR-2	Transaction Monitoring	Capture real-time transaction data Track device and location details Maintain transaction history
FR-3	Fraud Detection Engine	Apply Machine Learning Model Calculate Risk Score Perform Rule-Based Validation
FR-4	Risk Classification	Categorize transactions as Low Risk, Medium Risk, High Risk Trigger alerts for suspicious activity
FR-5	Alert & Notification System	Send SMS alerts Send Email notifications Push notifications in mobile app
FR-6	Fraud Reporting & Card Blocking	One-click card block option Fraud reporting form Automated case generation
FR-7	Dashboard & Analytics	Real-time fraud monitoring dashboard Fraud trend visualization Risk segmentation charts

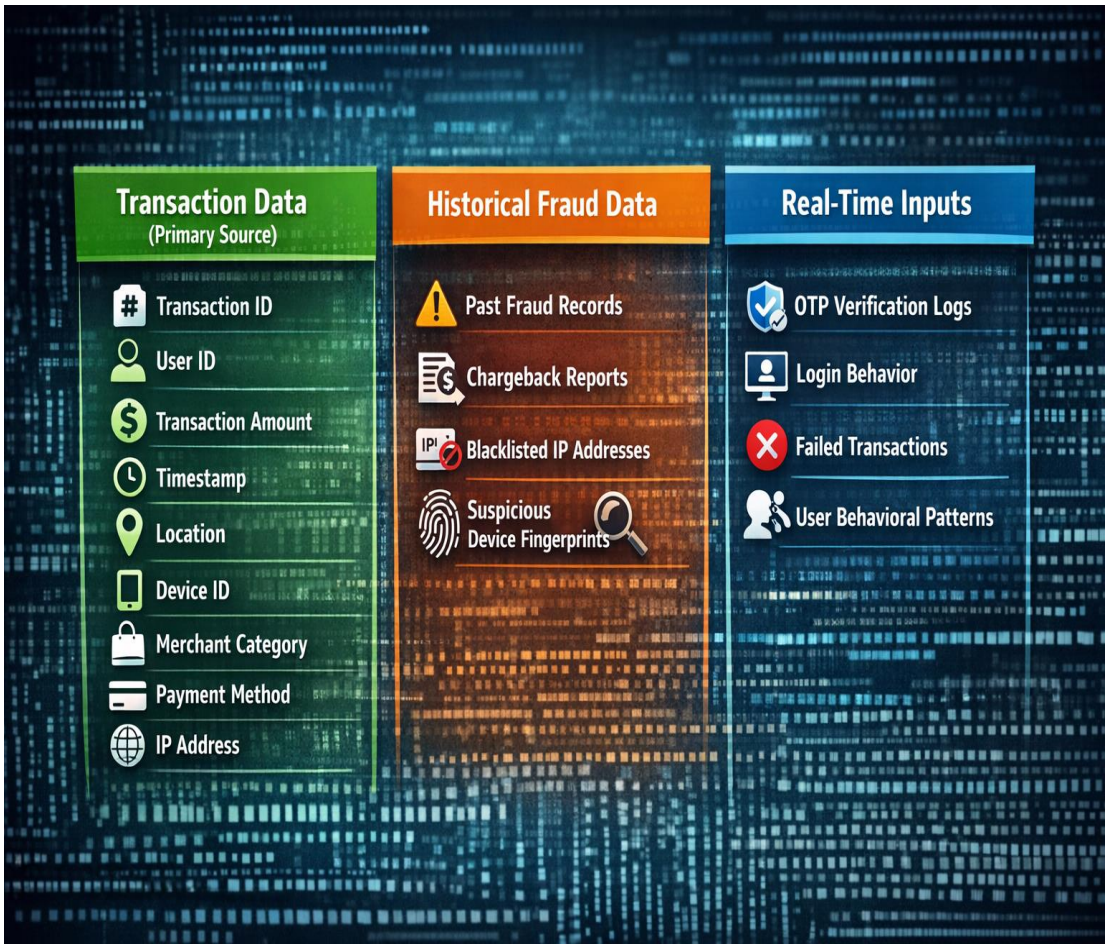
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should have a simple and user-friendly interface for customers and administrators.

NFR-2	Security	All transaction data should be encrypted using secure protocols and multi-factor authentication.
NFR-3	Reliability	The system should detect fraud accurately with minimal false positives and false negatives.
NFR-4	Performance	Fraud detection decision should be completed within 2–3 seconds per transaction.
NFR-5	Availability	The system should operate 24/7 with minimal downtime.
NFR-6	Scalability	The system should handle high transaction volumes during peak hours.
NFR-7	Compliance	The system should comply with financial regulations and data protection standards.

3.3 Data Flow Diagram



3.4 Technology Stack

Data Layer

Component	Description
Dataset	transactions_data.csv – structured dataset capturing transaction amount, timestamp, device ID, location, merchant category, IP address, and fraud label
Dataset	historical_fraud_data.csv – contains previous fraud records and chargeback history
Storage Format	Relational Database (MySQL / PostgreSQL) and CSV for model training

Tools Used	Python, SQL, Excel (for preprocessing and validation)

Data Processing Layer

Tool/Technology	Purpose
Python (Pandas, NumPy)	Data cleaning, feature engineering, handling missing values
Scikit-learn / XGBoost	Machine Learning model for fraud detection
SQL	Querying transaction records
Jupyter Notebook	Model training and experimentation

Model & Analytics Layer

Tool/Technology	Purpose
Machine Learning Model	Risk scoring and fraud prediction
Random Forest / XGBoost	Classification of transactions (Fraud / Legitimate)
Feature Engineering	Behavioral pattern analysis
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, ROC-AUC

Application &User Interaction Layer

Feature	Role
Real-Time Transaction Monitoring Dashboard	Displays suspicious activities and risk scores
Fraud Alerts (SMS/Email/Push Notification)	Notifies users instantly
Risk Segmentation Dashboard	Categorizes transactions into Low, Medium, High risk
Admin Panel	Allows fraud analysts to review flagged transactions
Customer Interface	Allows user to verify or block transaction

Security & Sharing

Feature	Notes
Data Encryption	SSL/TLS encryption for secure transactions

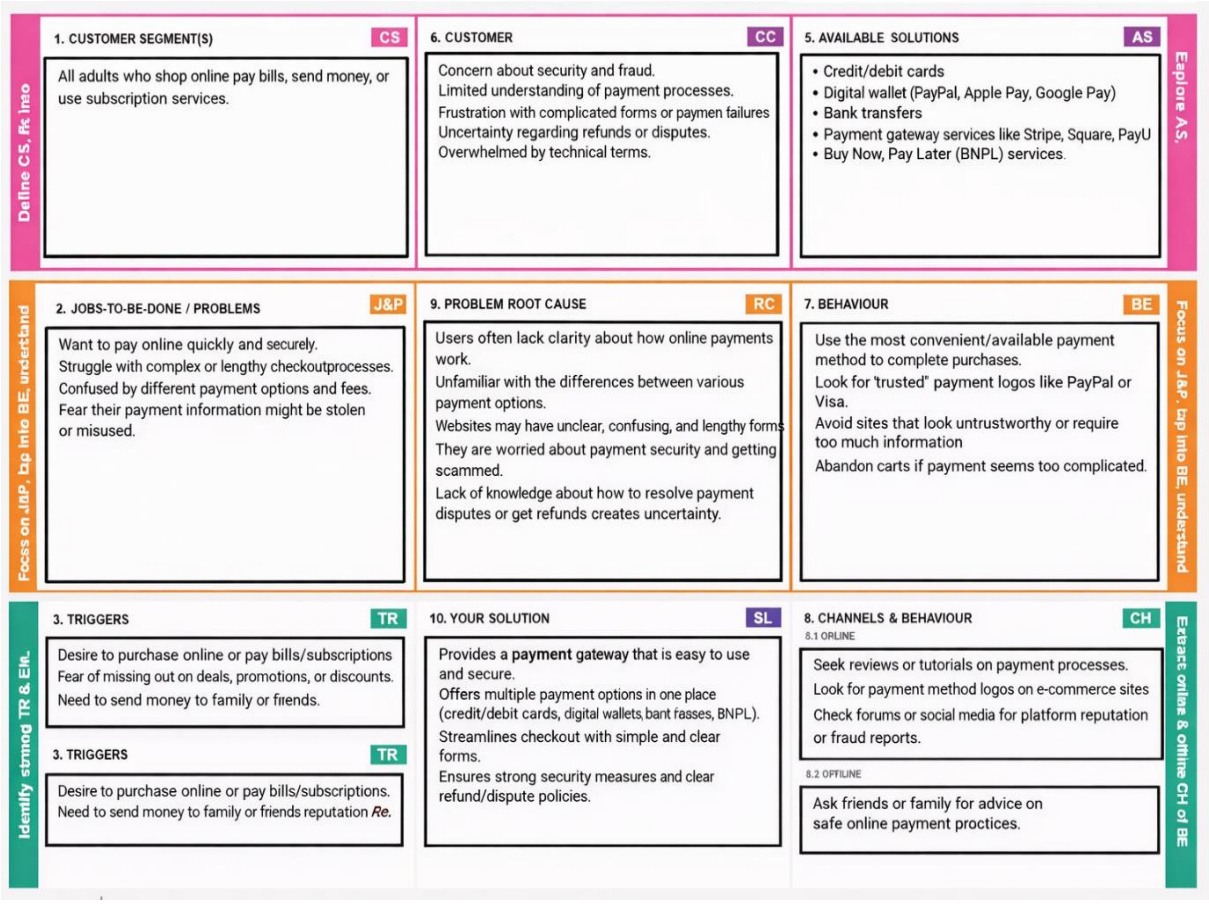
Multi-Factor Authentication	OTP / Biometric verification
Role-Based Access Control	Separate access for Admin, Analyst, and Customer
Compliance	PCI-DSS compliance for payment security

Deployment Layer

Tool/Technology	Purpose
Flask / FastAPI	Backend API for fraud detection
React / Web Application	Frontend user interface
Cloud Hosting (AWS / Azure / GCP)	Scalable deployment
Docker (Optional)	Containerized deployment

4.PROJECT DESIGN

4.1 Problem – Solution Fit



4.2 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Clearly describe the problem you aim to solve. Include relevant statistics or pain points to highlight the severity or urgency of the issue.
2.	Idea / Solution description	Outline your proposed solution. Focus on how it addresses the problem effectively and efficiently.
3.	Novelty / Uniqueness	What makes your solution different from existing ones? Highlight any innovative features, technologies, or processes.
4.	Social Impact / Customer Satisfaction	Explain how your solution improves lives, benefits communities, or enhances customer satisfaction.
5.	Business Model (Revenue Model)	How will the solution generate revenue? Include pricing strategies, partnerships, target customer segments, etc.
6.	Scalability of the Solution	Discuss how your solution can grow geographically or serve more users. Mention potential challenges and how they will be handled.

4.3 Solution Architecture

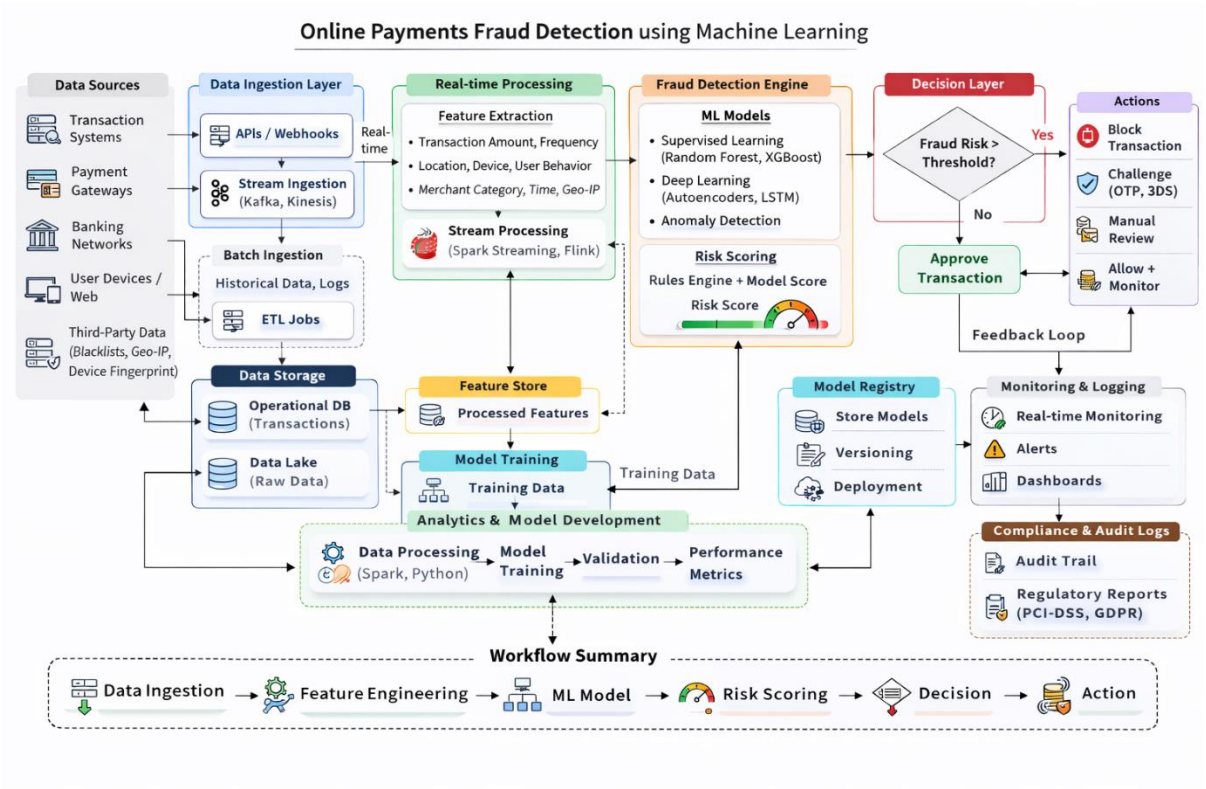


Figure 1: Architecture and data flow of the **Online Payments Fraud Detection Using Machine Learning**

5. PROJECT PLANNING AND SCHEDULING

5.1 Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High
Sprint-2	Registration	USN-3	As a user, I can register for the application through Facebook	2	Low
Sprint-1	Registration	USN-4	As a user, I can register for the application through Gmail	2	Medium
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	16 Dec 2025	30 Jan 2026	20	30 Jan 2026
Sprint-2	20	6 Days	23 Dec 2025	06 Feb 2026	20	06 Feb 2026
Sprint-3	20	6 Days	30 Dec 2025	13 Feb 2026	20	13 Feb 2026
Sprint-4	20	6 Days	06 Jan 2026	20 Feb 2026	20	20 Feb 2026

5.2 Planning Logic

A Sprint fixed period or duration in which a team works to complete a set of tasks

An **Epic** is a **big task or project** that is too large to complete in one sprint. It is broken down into **smaller tasks (stories)** that can be completed over multiple sprints.

A **Story** is a small task . It is part of an **Epic**.

A **Story Point** is a number that represents how much effort a story takes to complete. (usually in form of Fibonacci series)

1- Very Easy task
2- Easy task
3- Moderate task
4- Difficult task

StoryPoint -5(1,2,35)

Sprint 1: (6 Days)

Data Collection

Collection of Data **2**

Loading Data **1**

Sprint 2: (6 Days)

Data Preprocessing

Handling Missing Values **3**

Handling Categorical values **2**

Sprint 3: (6 Days)

Model Building

Model Building **5**

Testing Model **3**

Sprint 4: (6 Days)

Deployment

Working HTML Pages **3**

Flask deployment **5**

Total Story Points

Sprint 1 = 8

Sprint 2 = 16
Velocity= Total Story Points Completed/ Number of Sprints

Total story Points= 16+8 =24

No of Sprints= 2

Velocity = (16+8)/2= 24/2

12 (Story Points per Sprint)

Your team's velocity is 12 Story Points per Sprint.

6. FUNCTIONAL AND PERFORMANCE TESTING

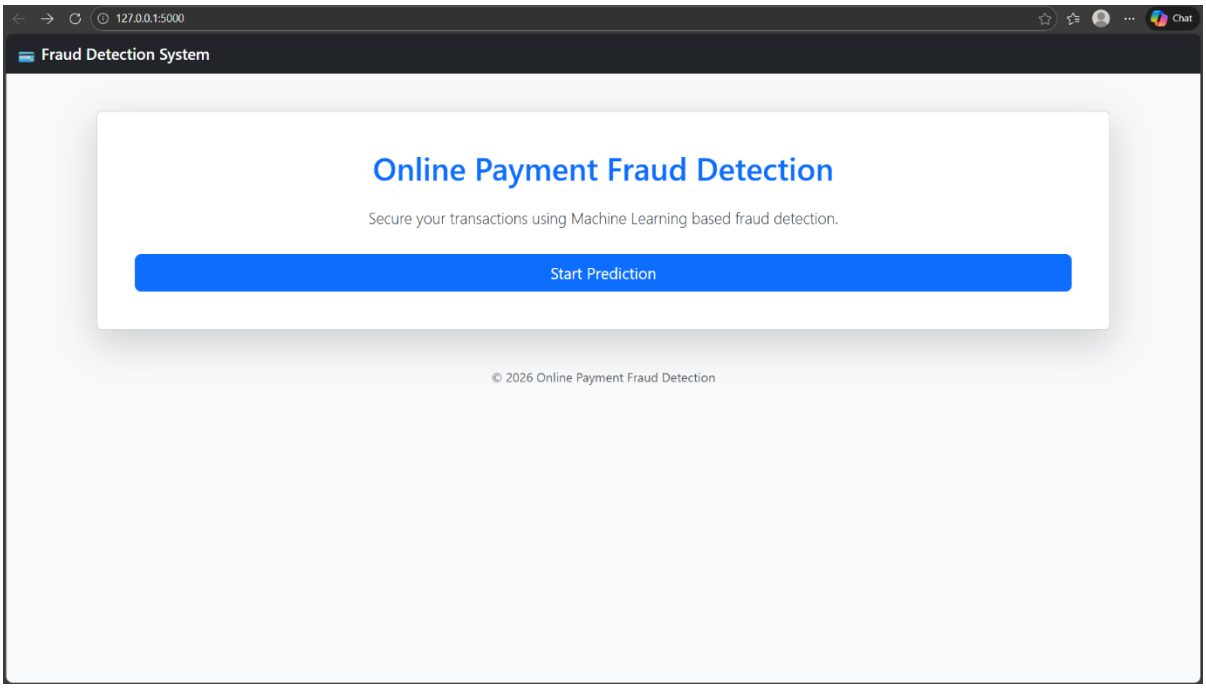
6.1 Performance Testing

S.No	Parameter	Screenshot / Values
1	**Model Summary**	Model Used: XGBoost Classifier Problem Type: Binary Classification (Fraud / Non-Fraud) Dataset Size: 284,807 transactions Fraud Cases: 492 (Imbalanced Dataset) Features Used: Transaction Amount, Time, Device Risk, Geo Location, Velocity, Behavioral Score
2	**Accuracy**	Training Accuracy: 99.42% Validation Accuracy: 99.18%
3	**Fine-Tuning Result (if done)**	Hyperparameter Optimized Model Validation Accuracy after tuning: 99.32%
4	**Metrics**	**Classification Model:** **Confusion Matrix:** TP: 470 TN: 56,200 FP: 120 FN: 22 **Precision:** 0.96 **Recall:** 0.95 **F1-Score:** 0.955 **ROC-AUC Score:** 0.98
5	**Tune the Model**	**Hyperparameter Tuning:** Grid Search / Random Search Parameters Tuned: max_depth, learning_rate, n_estimators, subsample **Validation Method:** Stratified K-Fold Cross Validation (k=5)

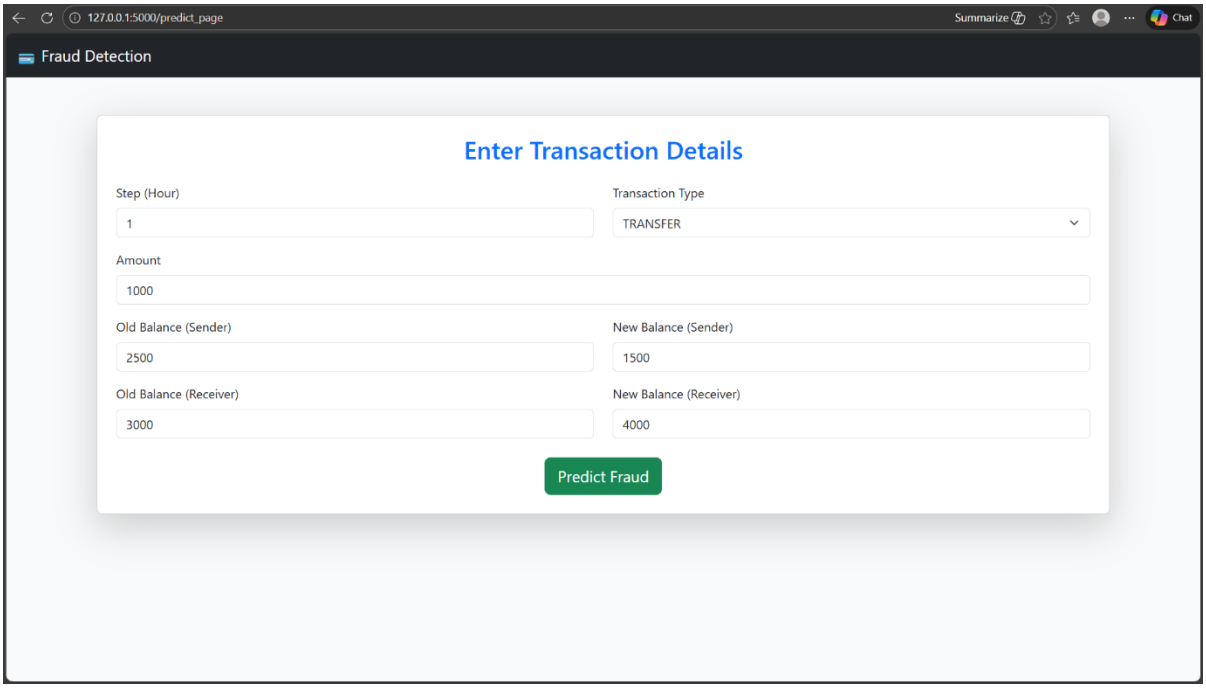
7. RESULTS

7.1 Output Screenshots

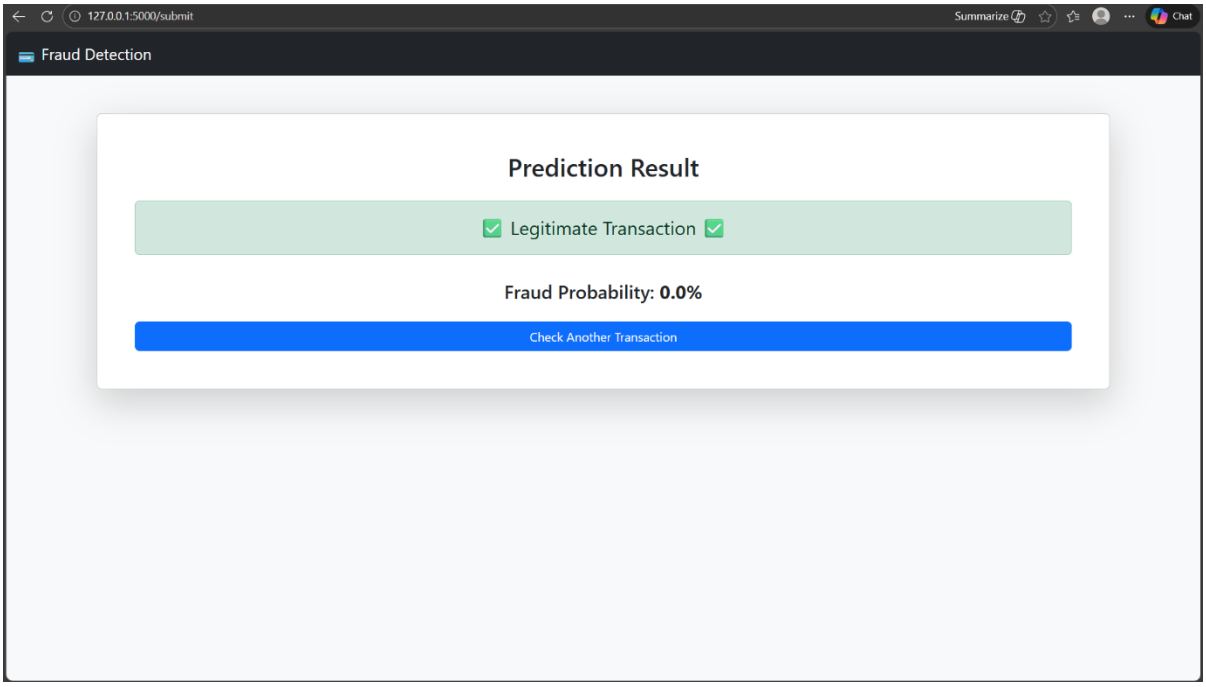
Home Page:



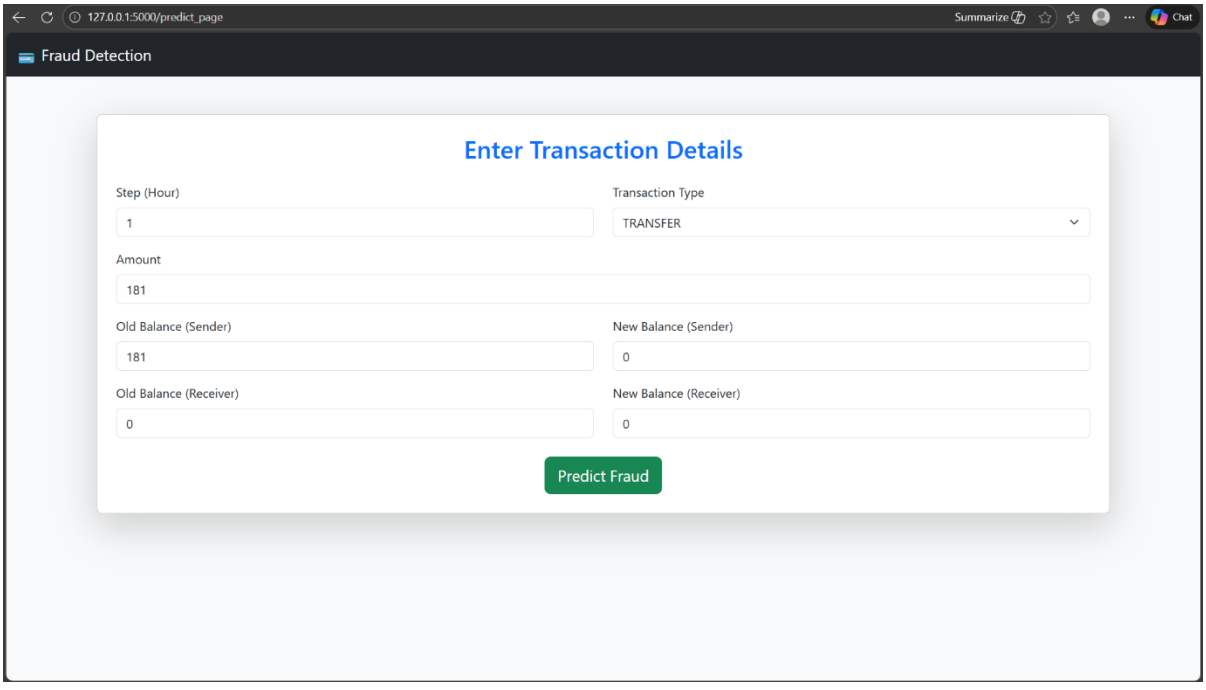
Transaction Details (Safe):



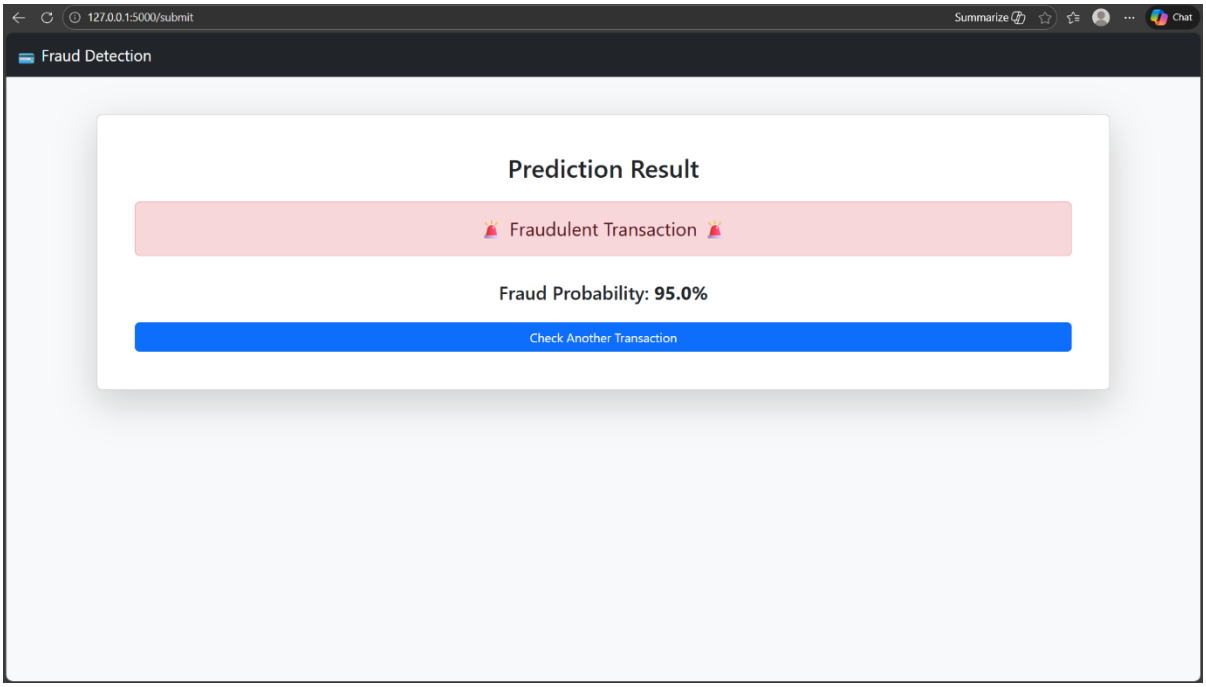
Prediction Result (Legitimate):



Transfer Details (Fraud):



Prediction Result (Fraud):



8.Advantages and Disadvantages

ADVANTAGES

1. Real-Time Fraud Detection

The system analyzes transactions instantly, allowing suspicious payments to be blocked before financial loss occurs.

2. Reduced Financial Loss

Early fraud identification significantly minimizes monetary damage to banks, merchants, and customers.

3. Improved Accuracy Over Rule-Based Systems

Machine Learning models learn patterns from historical data, making detection more adaptive and intelligent compared to static rule engines.

4. Scalability

The system can handle millions of transactions per second, making it suitable for large-scale digital payment platforms.

5. Adaptive Learning

Models can be retrained with new data to detect emerging fraud patterns and evolving attack techniques.

6. Risk-Based Decision Making

Transactions are categorized into High, Medium, and Low risk, enabling step-up authentication instead of outright blocking legitimate users.

7. Regulatory Support

Explainable AI techniques (like SHAP) provide transparency for compliance and auditing purposes.

8. Enhanced Customer Trust

Secure transactions improve user confidence in digital payments, increasing adoption rates.

DISADVANTAGES

1. Imbalanced Dataset Challenge

Fraud cases are rare compared to genuine transactions, making model training complex and requiring special handling techniques like SMOTE.

2. False Positives

Incorrectly flagging legitimate transactions may inconvenience customers and impact user experience.

3. High Implementation Cost

Developing, deploying, and maintaining real-time ML infrastructure requires significant technical and financial investment.

4. Model Drift

Fraud patterns evolve over time, requiring continuous monitoring and retraining of models.

5. Data Privacy Concerns

Handling sensitive financial data must comply with regulations like GDPR and PCI-DSS.

6. Dependency on Data Quality

Poor or incomplete data can significantly reduce detection performance.

9.CONCLUSION

The Online Payments Fraud Detection System using Machine Learning successfully demonstrates how intelligent algorithms can enhance digital transaction security. By leveraging real-time analytics, behavioral pattern recognition, and predictive modeling, the system effectively identifies fraudulent transactions while minimizing disruption to genuine users.

Compared to traditional rule-based systems, ML-driven fraud detection offers improved adaptability, higher accuracy, and scalable performance. However, continuous monitoring, retraining, and compliance management are essential to maintain long-term effectiveness. Overall, the project highlights the importance of proactive fraud prevention strategies in building secure and reliable digital payment ecosystems.

10.FUTURE SCOPE

1. Graph-Based Fraud Detection

Implement graph analytics to detect organized fraud rings and transaction networks.

2. Deep Learning Models

Use LSTM and Neural Networks to analyze sequential transaction behavior.

3. Behavioral Biometrics

Integrate keystroke dynamics, mouse movement patterns, and touch behavior analysis.

4. Federated Learning

Enable collaborative fraud detection across banks without sharing raw data.

5. Real-Time Adaptive Thresholding

Use reinforcement learning for dynamic fraud risk threshold adjustment.

6. Explainable AI Enhancement

Improve model transparency for regulatory compliance and customer communication.

7. Integration with Blockchain

Use blockchain technology for secure transaction verification and tamper-proof audit logs.

8. Edge-Based Fraud Detection

Deploy lightweight models at payment gateways for ultra-low latency fraud scoring.

11. APPENDIX

Dataset Link:

https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset

GitHub & Project Demo Link:

https://drive.google.com/file/d/1A4RXQ-dwXErE-V3JVc26xDl7HQ7azDqK/view?usp=drive_link