

# Online Payments Fraud Detection using Machine Learning

## 1. INTRODUCTION

### 1.1 Project Overview

Online payment systems have become an essential part of modern digital transactions. With the rapid growth of e-commerce, mobile banking, and digital wallets, fraudulent activities in online payments have also increased significantly. Fraudulent transactions lead to huge financial losses for banks, businesses, and customers.

This project, Online Payments Fraud Detection, is a machine learning-based web application that predicts whether a given transaction is fraudulent or genuine. The system analyzes transaction details such as transaction type, amount, old balance, and new balance to determine fraud probability.

The model is trained using historical transaction data and deployed as a web application for real-time fraud prediction.

### 1.2 Purpose

The main purpose of this project is:

- To build a predictive system that identifies fraudulent online transactions.
- To reduce financial losses caused by fraud.
- To deploy a trained ML model as a web application.
- To demonstrate practical implementation of ML in real-world financial systems.

## 2. IDEATION PHASE

### 2.1 Problem Statement

Online payment fraud has become a major issue in digital financial systems. Traditional rule-based fraud detection systems are not efficient enough to handle modern fraud patterns.

#### **Problem:**

Design and develop a machine learning-based system that can accurately detect fraudulent online payment transactions using historical transaction data.

### 2.2 Empathy Map Canvas

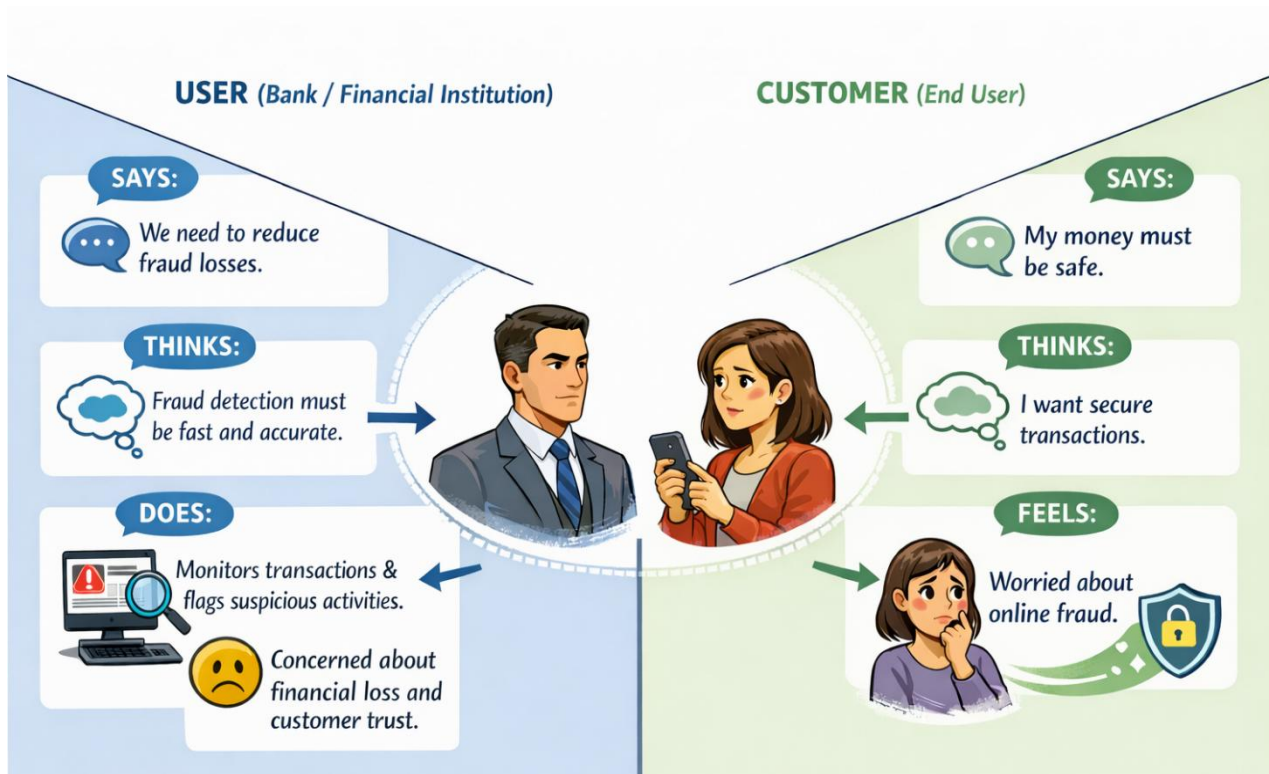
User (Bank / Financial Institution):

- Says: We need to reduce fraud losses.
- Thinks: Fraud detection must be fast and accurate.

- Does: Monitors transactions and flags suspicious activities.
- Feels: Concerned about financial loss and customer trust.

Customer (End User):

- Says: My money must be safe.
- Thinks: I want secure transactions.
- Feels: Worried about online fraud.



## 2.3 Brainstorming

Different approaches were discussed:

- Rule-based fraud detection
- Statistical anomaly detection
- Machine learning models (Logistic Regression, Random Forest, XGBoost)
- Real-time fraud scoring system

Finally, machine learning-based classification was selected as the best solution due to higher accuracy and adaptability.

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map

1. User enters transaction details.
2. Data is sent to backend.

3. ML model processes input.
4. System predicts fraud or genuine.
5. Result is displayed to user.

### 3.2 Solution Requirement

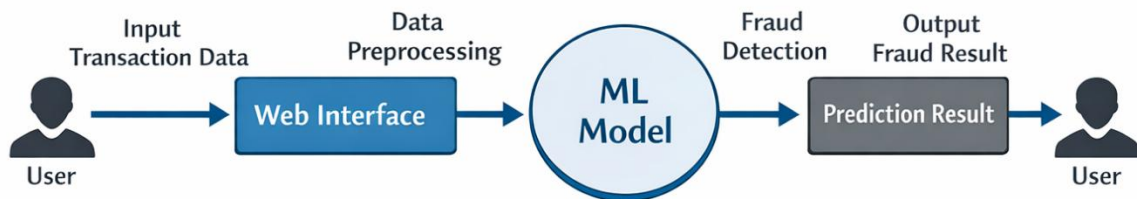
#### Functional Requirements:

- User should input transaction details.
- System should process input data.
- System should predict fraud status.
- Display result clearly.

#### Non-Functional Requirements:

- Fast prediction time.
- High accuracy.
- Secure data handling.
- Easy-to-use interface.

### 3.3 Data Flow Diagram (DFD)



### 3.4 Technology Stack

- Programming Language: Python
- Machine Learning Library: XGBoost / Scikit-learn
- Frontend: HTML, CSS
- Backend Framework: Flask
- Deployment Platform: Render
- Dataset Source: Kaggle Online Payment Fraud Dataset

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit

The problem requires real-time fraud detection with high accuracy. Machine learning models are suitable because:

- They learn complex patterns.
- They improve over time with more data.
- They reduce manual monitoring.

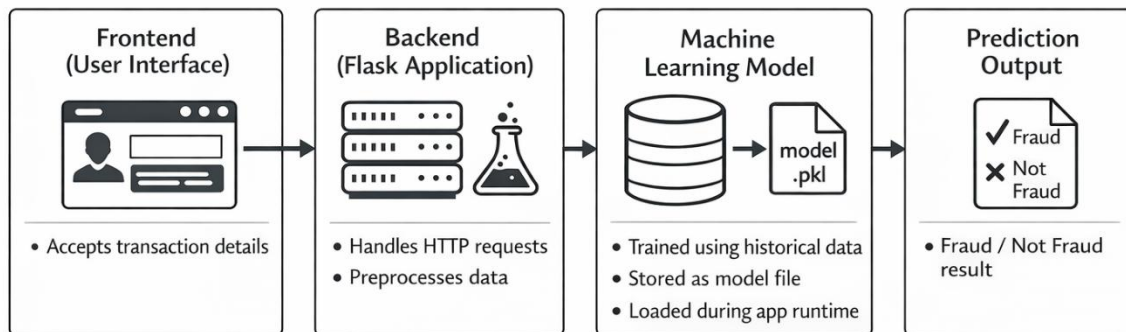
Thus, ML-based classification fits the problem effectively.

### 4.2 Proposed Solution

The proposed system:

- Takes transaction input.
- Converts categorical values to numerical form.
- Applies trained ML model.
- Predicts whether transaction is fraudulent or genuine.
- Displays result with prediction probability.

### 4.3 Solution Architecture



## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning

#### Phase 1 – Research

- Study fraud detection techniques.
- Explore dataset.

#### Phase 2 – Data Preprocessing

- Clean data.
- Handle missing values.
- Encode categorical variables.

### **Phase 3 – Model Training**

- Train multiple models.
- Compare accuracy.
- Select best model.

### **Phase 4 – Web Application Development**

- Build frontend.
- Integrate ML model.

### **Phase 5 – Testing & Deployment**

- Test predictions.
- Deploy on Render.

## **6. FUNCTIONAL AND PERFORMANCE TESTING**

### **6.1 Performance Testing**

#### **Model Evaluation Metrics:**

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

#### **Performance Result:**

- Model achieved high accuracy.
- Good fraud detection rate.
- Low false positive rate.

Prediction response time is fast (less than a few seconds).

## **7. RESULTS**

The system successfully:

- Detects fraudulent transactions.
- Provides real-time prediction.
- Works through web interface.
- Gives reliable fraud probability.

### **7.1 Output Screenshots**

- a. Home Page

## Online Payment Fraud Detection




b. Input Form

## Online Payment Fraud Detection



c. Fraud Prediction Result


## Prediction Result

 **Fraud Transaction**  
Confidence: 99.55%

[Check Another Transaction](#)

d. Genuine Transaction Result

## Prediction Result

 **Legitimate Transaction**  
Confidence: 99.99%

[Check Another Transaction](#)

## **8. ADVANTAGES & DISADVANTAGES**

### **Advantages:**

- Real-time fraud detection
- High prediction accuracy
- Automated system
- Reduces financial losses

### **Disadvantages:**

- Requires quality training data
- May produce false positives
- Needs regular model updates

## **9. CONCLUSION**

The Online Payments Fraud Detection system successfully demonstrates how machine learning can be used to detect fraudulent transactions. The model accurately classifies transactions based on input features and provides real-time results through a web application. This project highlights the importance of AI in financial security and fraud prevention.

## **10. FUTURE SCOPE**

- Integration with live banking systems
- Deep learning-based fraud detection
- Real-time streaming data analysis
- Mobile application integration
- Continuous model retraining

## **11. APPENDIX**

Source Code:

(Attach your Flask + ML model code)

Dataset Link:

Kaggle – Online Payment Fraud Detection Dataset  
(Provide your actual dataset link here)

### **GitHub & Project Demo Link**

GitHub Repository:  
(Add your GitHub link here)

Project Demo (Render):

<https://online-payments-fraud-detection-84m0.onrender.com>