Exploratory Analysis of Online Payments Fraud Detection using Machine Learning Project Documentation

1. Introduction

Project Title: Online Payments Fraud Detection using Machine Learning

Team Members:
Dedeep Darapureddy(ML & Backend), Team members
Team Members (Frontend, Data Analysis, Deployment)

2. Project Overview

To detect fraudulent online payment transactions using Machine Learning techniques and reduce financial losses for customers and financial institutions.

Features:

- Real-time fraud detection
- Transaction data preprocessing
- Fraud probability scoring

3. Architecture

Frontend:
HTML, CSS, JavaScript dashboard for monitoring transactions and displaying fraud results.

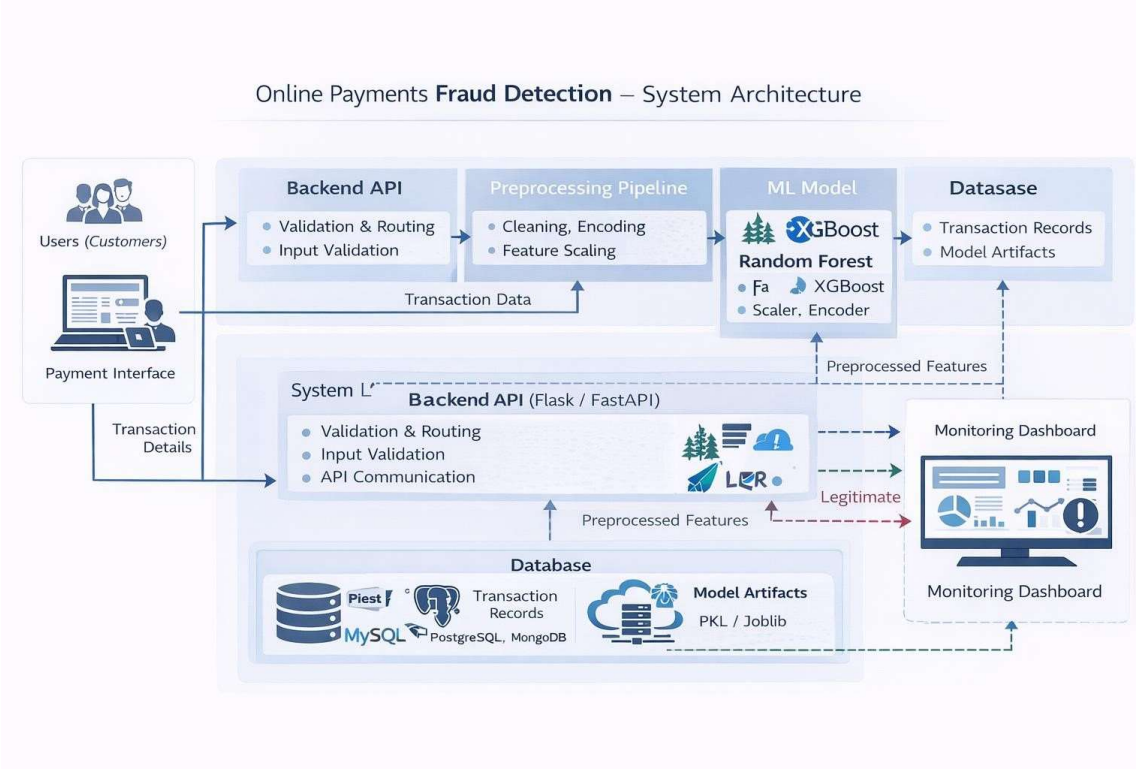
Backend:
Flask/FastAPI-based Python server handling transaction requests, validation, and prediction logic.

Model Layer:
Trained Machine Learning model (Random Forest / XGBoost) with scaler and encoder artifacts.

Database Layer:

Stores transaction records, fraud logs, and model artifacts.

System Architecture Diagram



4. Setup Instructions

Prerequisites:

- Python 3.x
- Flask or FastAPI
- Pandas
- NumPy
- Scikit-learn
- Joblib
- MySQL/MongoDB (optional)

Installation:

Install dependencies using pip and run the backend application.

5. Folder Structure

templates/ → HTML dashboard files static/

→ CSS, JS, Images main.py → Flask/FastAPI

backend model_training.ipynb → ML model training

notebook fraud_model.pkl → Trained model

scaler.pkl → Saved scaler encoder.pkl →

Saved encoder

6. Running the Application python main.py

http://localhost:5000

7. API Documentation

Accepts transaction details and returns fraud prediction.

8. Authentication

Not implemented in current version (Future scope: JWT-based authentication and secure token validation).

9. User Interface The

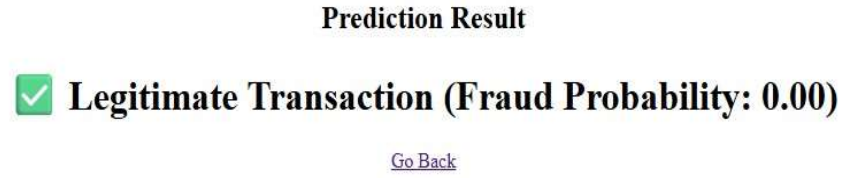
UI provides:

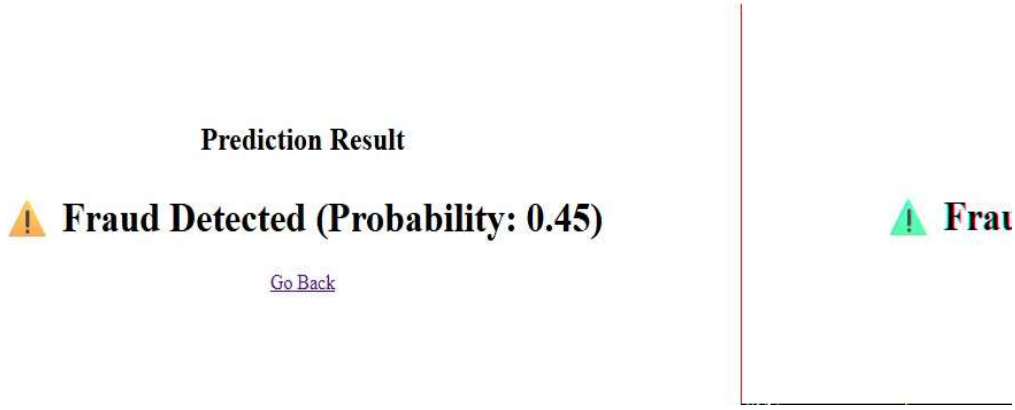
- Transaction input form (for testing)
- Fraud detection result display
- Fraud monitoring dashboard
- Statistical charts for fraud trends
- Alert notifications

10. Testing

Model evaluated using accuracy, confusion matrix, ROC-AUC. **11.**

Screenshots / Demo





12. Known Issues

- Performance depends on dataset quality
- Imbalanced data affects recall
- Requires periodic retraining to handle evolving fraud patterns

13. Future Enhancements

- Deep Learning-based fraud detection
- Real-time streaming using Kafka
- Cloud deployment (AWS/Azure/GCP)
- Mobile fraud alert system
- Cross-bank fraud intelligence sharing

- Advanced risk scoring system