Online Payments Fraud Detection System – Project Report

---

1. Project Title

Machine Learning Based Online Payments Fraud Detection System

2. Project Description

This project implements a machine learning–based fraud detection system designed to identify fraudulent online payment transactions in real-time. The system analyzes transaction details such as amount, time, user behavior, and device information to classify transactions as **Fraudulent** or **Legitimate**.

The system provides an intuitive dashboard for monitoring transactions and integrates a backend API for real-time fraud prediction.

2.1 Problem Statement

Online payment fraud is increasing rapidly due to:

- Growth of digital transactions
- Phishing and cyber attacks
- Card-not-present fraud
- Identity theft

Traditional rule-based systems:

- Fail to detect evolving fraud patterns
- Generate high false positives
- Affect genuine customer experience
- This project aims to provide an intelligent, adaptive fraud detection system using machine learning.

2.2 Solution

A Flask/FastAPI-based fraud detection system that:

- Accepts transaction inputs via API/UI
- Performs preprocessing and feature engineering
- Uses trained ML models to classify transactions

- Provides fraud probability scores
- Triggers alerts for suspicious activities
- Displays monitoring dashboard for analysts

3. Technologies Used

3.1 Backend

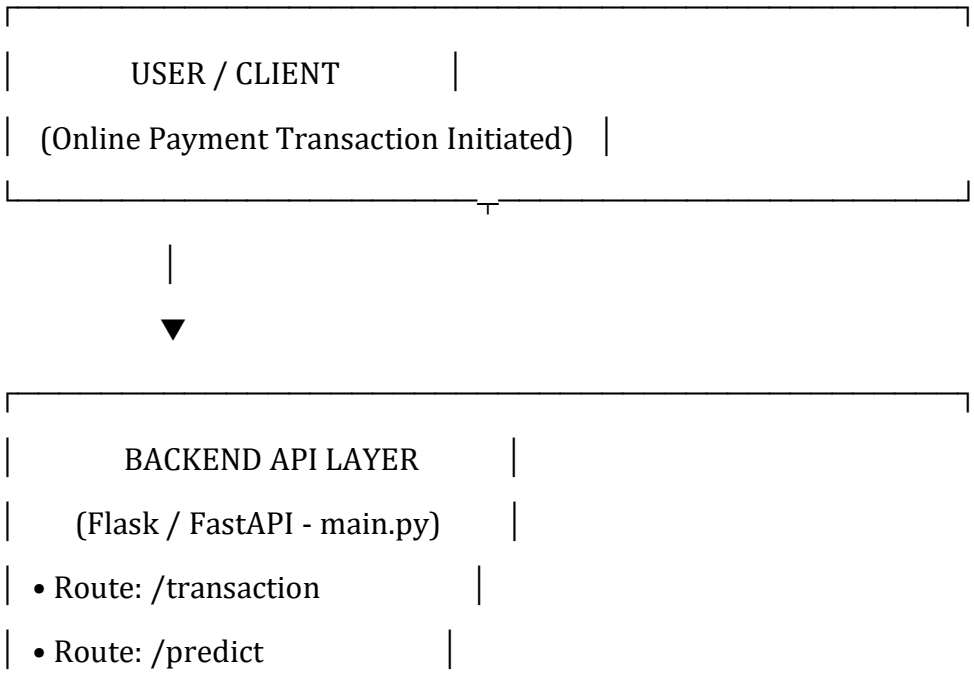
Technology	Purpose
Python 3.x	Programming language
Flask / FastAPI	Web framework
Pandas	Data manipulation
NumPy	Numerical computations
Scikit-learn	Machine learning modeling
Joblib / Pickle	Model serialization

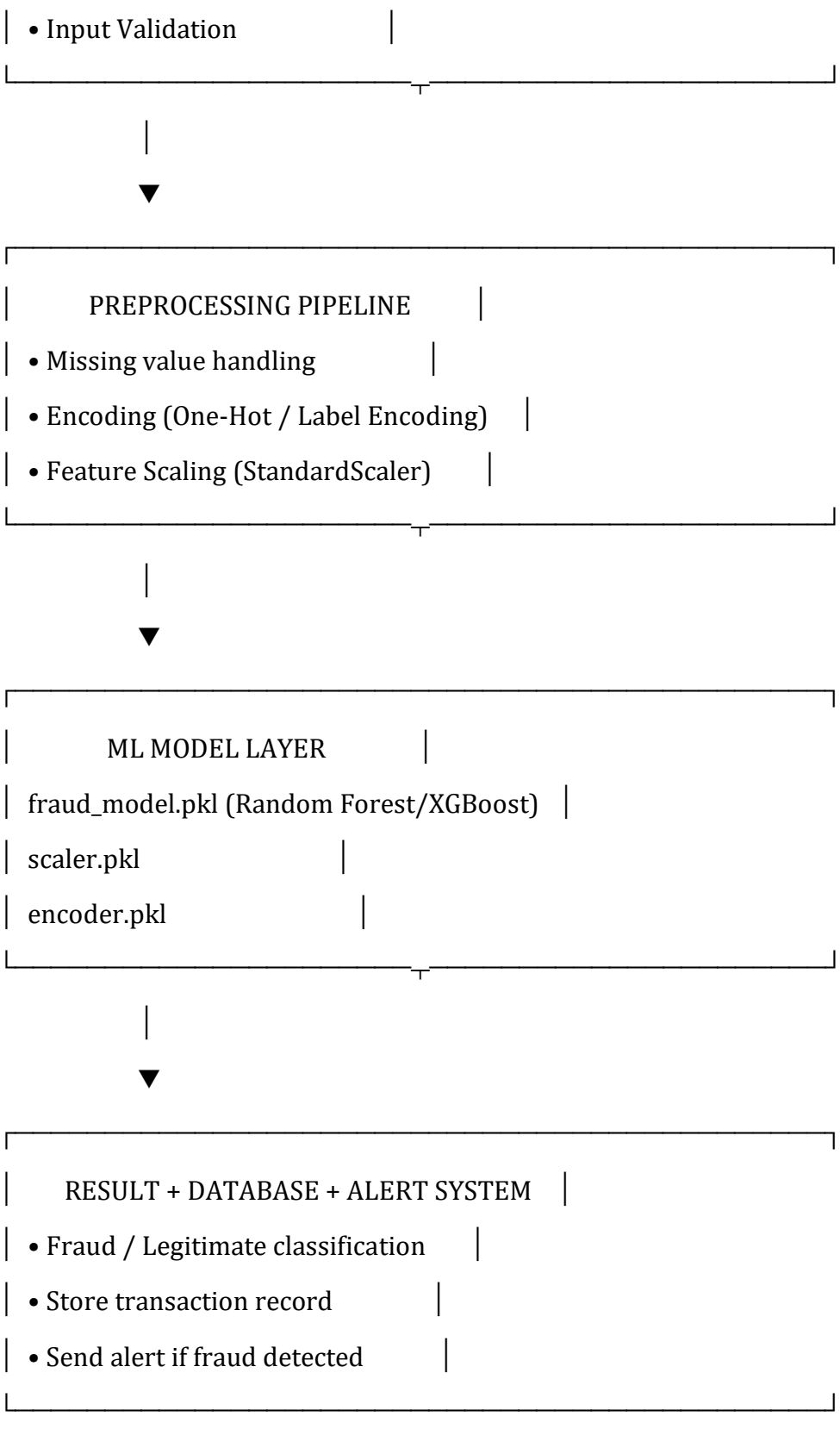
3.2 Frontend

Technology	Purpose
HTML5	Page structure
CSS3	Styling
JavaScript	Interactivity
Chart.js	Fraud trend visualization
Bootstrap	Responsive UI

---

4. System Architecture




• Input Validation
PREPROCESSING PIPELINE
• Missing value handling
• Encoding (One-Hot / Label Encoding)
• Feature Scaling (StandardScaler)
ML MODEL LAYER
fraud_model.pkl (Random Forest/XGBoost)
scaler.pkl
encoder.pkl
RESULT + DATABASE + ALERT SYSTEM
• Fraud / Legitimate classification
• Store transaction record
• Send alert if fraud detected

5. Features Implemented

5.1 Transaction Processing Module

- Accepts transaction details
- Validates input parameters
- Generates fraud probability score
- Stores transaction logs

5.2 Fraud Monitoring Dashboard

- Displays flagged transactions
- Shows fraud percentage trends
- Provides daily/weekly fraud statistics

5.3 Alert System

- Real-time fraud alerts
- Suspicious transaction notification
- case logging

6. Machine Learning Model

6.1 Dataset

- Source: Public Credit Card Fraud Dataset (Kaggle)
- Records: 284,807 transactions
- Fraud Cases: 492 (~0.17%)
- Target Variable: Class (0 = Legitimate, 1 = Fraud)

6.2 Preprocessing

1) Handling imbalanced dataset (SMOTE)
2) Feature scaling (StandardScaler)
3) Train-Test Split (80-20)
4) Feature engineering (transaction frequency, anomaly detection)

6.3 Model Training

- Algorithm Used:
    - Logistic Regression
    - Random Forest
    - XGBoost (Best Performing)
- Cross-Validation: 5-Fold

- Hyperparameter Tuning: GridSearchCV

6.4 Model Performance

Metric	Score
Accuracy	~99%
Precision	~92%
Recall	~85%
F1-Score	~88%
ROC-AUC	~0.97

7. Files Modified/Created

File	Description	Type
main.py	Backend API and prediction logic	Created
model_training.ipynb	Model training notebook	Created
fraud_model.pkl	Trained ML model	Generated
scaler.pkl	Feature scaler	Generated
encoder.pkl	Feature encoder	Generated
templates/dashboard.html	Monitoring UI	Created
static/css/style.css	Styling	Created

8. How to Run

8.1 Prerequisites

- Python >= 3.8
- Flask / FastAPI
- Pandas
- NumPy
- Scikit-learn

8.2 Installation

```
# Clone repository
git clone <repository-url>
cd Fraud_Detection_System

# Create virtual environment
python -m venv .venv
.venv\Scripts\activate

# Install dependencies
pip install flask pandas numpy scikit-learn

# Run application
python main.py
```

9. Application Link

Local Development URL: http://localhost:5000

---

10. Future Enhancements

1. Deep Learning (Neural Networks)
2. Real-time streaming with Apache Kafka
3. Mobile fraud alert integration
4. Multi-bank fraud intelligence sharing
5. Cloud deployment (AWS/Azure/GCP)

11. Conclusion

This project successfully demonstrates the integration of Machine Learning with backend systems to build a real-time fraud detection solution. The application:

- Detects fraudulent transactions accurately
- Reduces financial losses
- Minimizes false positives
- Enhances customer trust in digital payment systems
- Provides scalable and modular architecture

12. References

1. Kaggle Credit Card Fraud Detection Dataset
2. Flask Documentation – https://flask.palletsprojects.com/
3. Scikit-learn Documentation – https://scikit-learn.org/
4. XGBoost Documentation

Report Prepared By: Vasanth Dora

Date: February 2026
Course: Artificial Intelligence and Machine Learning