

Project Design Phase

Solution Architecture

Date	
Team ID	LTVIP2026TMIDS40243
Project Name	Online Payments Fraud Detection
Maximum Marks	4 Marks

Solution Architecture

The solution architecture of the Online Payments Fraud Detection System bridges the gap between the growing issue of online fraud and a machine learning-based technical solution. It defines how transaction data flows from user input through preprocessing and feature encoding to model prediction and final fraud classification. The system is built using a trained Decision Tree model, with additional comparisons using Random Forest, SVM, and XGBoost, all integrated into a Flask web application. It clearly outlines the structure for data handling, model training, and real-time prediction of fraudulent transactions. Overall, the architecture ensures secure processing, efficient fraud probability calculation, and supports future enhancements and deployment improvements.

Solution Architecture:

The solution architecture of the Online Payments Fraud Detection System connects business needs with machine learning technology to prevent financial fraud in real-time. Our architecture is designed to:

- Detect fraudulent transactions using Machine Learning models (Decision Tree, Random Forest, SVM, XGBoost).
- Process transaction input data through preprocessing and feature encoding.
- Predict whether a transaction is Fraud or Not Fraud using the selected Decision Tree model.
- Provide real-time prediction results through a Flask-based web application.

The architecture ensures secure data handling, efficient model execution, and fast response time for online transaction fraud detection.

Example - Solution Architecture Diagram:

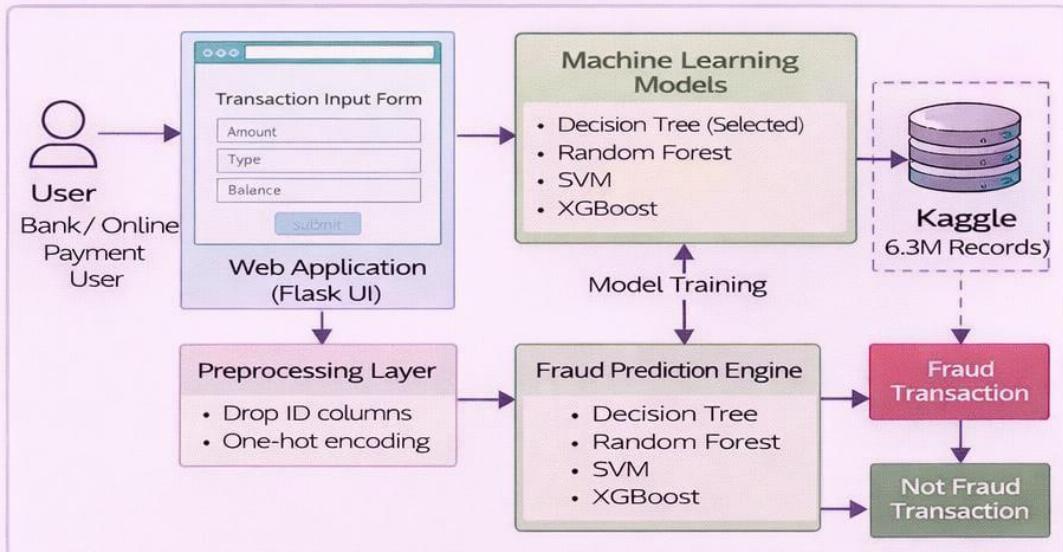


Figure 1: Architecture and data flow of the Online Payments Fraud Detection System

User → Flask Web App → Transaction Input → Data Preprocessing → Machine Learning Model → Fraud Prediction → Fraud Prediction → Display Result ■■■■■

Figure 1: Architecture and data flow of the Online Payments Fraud Detection System

