**Student Name:** KALIMUTHU M

**Register Number:** 732323106023

**Institution:** SSM COLLEGE OF ENGINEERING

**Department:** Electronics And Communication Engineering

**Date of Submission:** April 29, 2025

**Github Repository Link:** <https://github.com/kalimuthu721/Guarding-transactions-with-Al-powered-credit-card-fraud-detection-and-prevention>

Project Title:

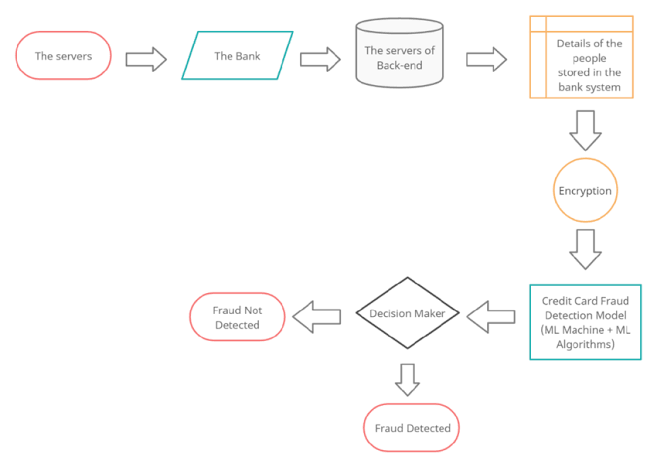
Guarding Transactions with AI-Powered Credit Card Fraud Detection and Prevention

1. Problem Statement

With the surge in digital financial transactions, credit card fraud has escalated, posing significant risks to both financial institutions and consumers. Traditional rule-based systems are inadequate in adapting to evolving fraud strategies, often resultisng in delayed detection and false positives. This project addresses the need for a real-time, adaptive, and intelligent solution by leveraging artificial intelligence and machine learning to accurately detect and prevent fraudulent transactions while minimizing disruption to legitimate ones.

2. Objective

To develop an AI-powered fraud detection system capable of identifying fraudulent credit card transactions in real-time, using machine learning models trained on transactional data. The system will provide scalable and efficient detection to enhance security in digital payments.

3. Project Workflow Flowchart

4. Data Description

* Dataset Source: European cardholders dataset (e.g., Kaggle).
* Records: ~284,807 transactions
* Imbalance: Fraudulent cases < 0.2%
* Features: Numerical-only (PCA-transformed for anonymization)
* Target Variable: Class (0 = Non-Fraud, 1 = Fraud)
* Preprocessing Needs: Scaling, handling imbalance (SMOTE/under-sampling)

5. Data Preprocessing Steps

* Import libraries and dataset
* Handle nulls and duplicates
* Apply SMOTE for class imbalance
* Feature scaling using StandardScaler
* Train-test split for model training

from imblearn.over\_sampling import SMOTE

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

6. Exploratory Data Analysis (EDA)

* Class distribution: Visualized using countplot
* Amount & Time distributions: Identify fraud-related trends
* Correlation matrix: Identify predictive features
* Distribution comparison: Fraud vs. Non-fraud for key variables
* Dimensionality reduction: PCA/t-SNE for 2D visual inspection

7. Model Building Strategy

* Problem Understanding: Real-time fraud detection with evolving threats
* Preprocessing: Feature engineering, scaling, anomaly handling
* Algorithms Considered:
  + Logistic Regression, Random Forest
  + XGBoost, Neural Networks
  + Autoencoders, LSTM, GNNs for advanced patterns
* Evaluation Metrics: Precision, Recall, F1-Score (focus on Recall)
* Deployment Stack: Flask/FastAPI + Docker + AWS/GCP + Kafka for streaming

8. Visualization & Model Insights

1. Confusion Matrix
2. ROC Curve
3. Precision-Recall Curve
4. Feature Importance Plot
5. SHAP Summary for Interpretability
6. Time-Based Fraud Trends

9. Tools & Technologies

* Languages/Frameworks: Python, Flask, scikit-learn, XGBoost
* Libraries: pandas, seaborn, matplotlib, imblearn, SHAP
* Platforms: AWS/GCP for cloud deployment, Kafka for streaming
* Visualization Tools: Matplotlib, Seaborn, SHAP

10. Team Members & Roles

| Name | Role Description |
| --- | --- |
| Krishna Priya .S | Data collection, model building, and deployment |
| Kalimuthu .M | Data analysis and visualization |
| Logesh .K | Feature engineering and model evaluation |
| Lavanya .R | Web interface design and implementation |