Phase-2 Submission Template

Student Name: NICKEL

Register Number: 212923106027

Institution: ST.JOSEPH COLLEGE OF ENGINEERING

Department: ELECTRONICS AND COMMUNICATION ENGINEERING

Date of Submission: 29-04-2025

Github Repository Link: **https://github.com/Jo-c-i/NAN-MUDHALVAN.git**

1. Problem Statement

Credit card fraud is a major global issue, costing businesses and consumers billions of dollars every year. Fraudsters use increasingly sophisticated methods—like phishing, identity theft, and card skimming—to make unauthorized transactions. Massive Financial Impact: Global losses from credit card fraud exceed $30 billion annually, affecting banks, retailers, and cardholders alike.

Solving this problem with AI helps create a safer, more trustworthy financial ecosystem while saving billions in losses and reducing the burden on both institutions and customers.

2. Project Objectives

Project Goal:

The primary aim of this project is to develop an AI-powered system that accurately detects and prevents credit card fraud in real-time, minimizing false positives while ensuring legitimate transactions are not disrupted.

Key Outcomes:

1. High-Accuracy Fraud Detection Model:Build and train a machine learning model capable of distinguishing between legitimate and fraudulent credit card transactions with high precision and recall.

2. Real-Time Prediction System:

Implement a system that can flag suspicious transactions as they occur, enabling instant alerts or blocks before financial loss happens.

3. Behavioral Pattern Analysis:

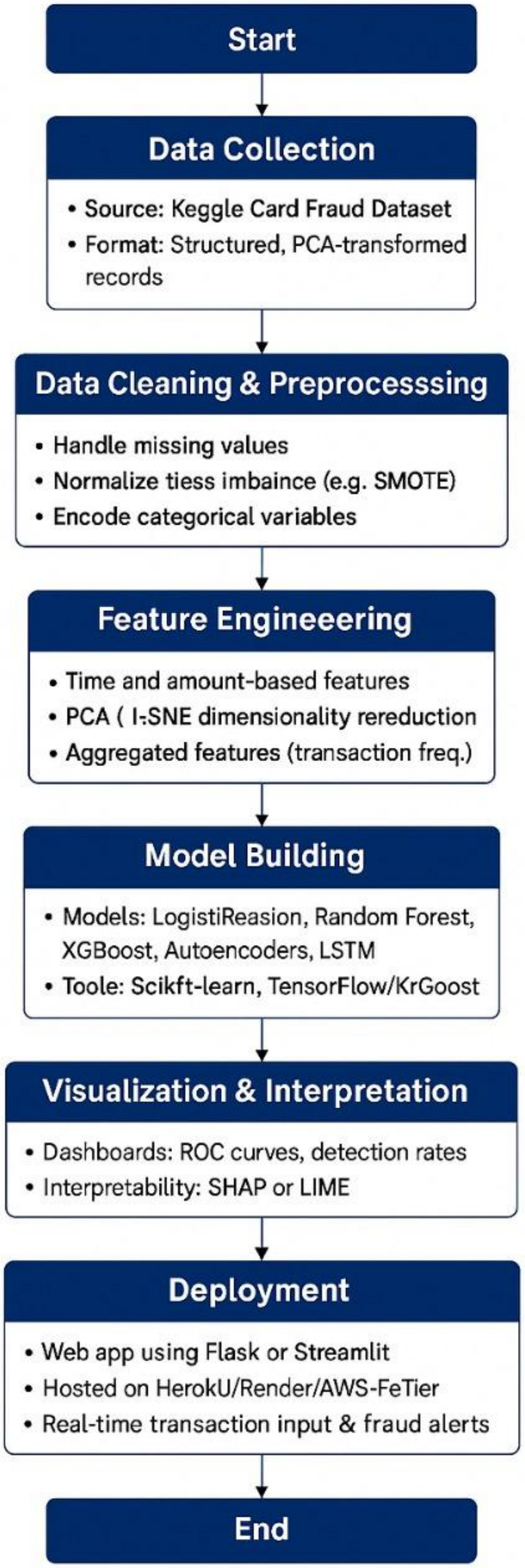
Gain insights into common patterns and behaviors associated with fraudulent activity versus normal user behavior.

4. False Positive Reduction:

Optimize the model to reduce the number of false alarms, improving customer experience by avoiding unnecessary transaction declines.

By the end of the project, the solution should empower financial institutions with a robust, intelligent layer of defense against credit card fraud, ultimately reducing losses and enhancing trust in digital payment systems.

3. Flowchart of the Project



Start

Data Collection

- Source: Keggle Card Fraud Dataset
- Format: Structured, PCA-transformed records

Data Cleaning & Preprocesssing

- Handle missing values
- Normalize tiess imbaince (e.g. SMOTE)
- Encode categorical variables

Feature Engineeering

- Time and amount-based features
- PCA (I-SNE dimensionality rereduction
- Aggregated features (transaction freq.)

Model Building

- Models: LogistiReasion, Random Forest, XGBoost, Autoencoders, LSTM
- Toole: Scikft-learn, TensorFlow/KrGoost

Visualization & Interpretation

- Dashboards: ROC curves, detection rates
- Interpretability: SHAP or LIME

Deployment

- Web app using Flask or Streamlit
- Hosted on HerokU/Render/AWS-FeTier
- Real-time transaction input & fraud alerts

End

4. Data Description

Dataset Description: For this project, I will use the "Credit Card Fraud Detection" dataset available on Kaggle.

Source: Kaggle https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Access: Publicly available

Ownership: Provided by ULB (Université Libre de Bruxelles)

Type: Static (downloaded once and does not update in real-time)

Number of Records: 284,807 transactions

5. Data Preprocessing

Data Preprocessing Pipeline

Load Data

Load the Kaggle fraud dataset and inspect for nulls or anomalies.

Normalize Features

Scale Amount and Time using StandardScaler.

Handle Class Imbalance

Use SMOTE to oversample the minority (fraud) class.

Train-Test Split

Apply Stratified Split to preserve class distribution.

6. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA)

Analyzed class distribution to highlight severe imbalance between fraudulent and legitimate transactions.

Visualized feature distributions (e.g., Amount, Time) to detect patterns and outliers.

Used correlation heatmaps to explore relationships between features.Applied box plots and histograms to compare fraudulent vs. non-fraudulent transaction characteristics.

Identified time-based trends in fraudulent activity for better feature insight.

7. Feature Engineering

- Time-based features to capture temporal patterns
- Amount-based features to detect anomalous spending
- Behavioral pattern features to model user spending habits
- Aggregated features for historical context
- Dimensionality reduction techniques
- Approaches to handle class imbalance

8. Model Building

A clear breakdown of baseline and advanced models

Effective training approaches for fraud detection problems

Strategies for handling the imbalanced nature of fraud data

Key evaluation metrics specifically suited for fraud detection

Methods for ensuring model interpretability

A step-by-step implementation strategy

9. Visualization of Results & Model Insights

- Key exploratory visualizations to understand fraud patterns
- Performance visualization techniques to evaluate model effectiveness
- Interpretability visualizations to explain model decisions
- Business-focused dashboards to demonstrate value
- Real-time monitoring visualizations for operational use
- Interactive reporting for investigation and analysis

10. Tools and Technologies Used

Programming Language:

Python (main language for data analysis, modeling, and deployment)

Notebook / IDE:

Jupyter Notebook

Google Colab (optional for cloud-based execution)

Libraries:

Data Processing: pandas, numpy

Visualization: matplotlib, seaborn, plotly (optional)

Modeling: scikit-learn, xgboost, lightgbm, imbalanced-learn

Interpretation: shap

11. Team Members and Contributions

- Project Manager – NICKEL SUN A
- Data Scientist – MAGESH S
- QA & Testing Lead – JO C I