PROJECT PLAN: FRAUD DETECTION FOR FINANCIAL RISK MANAGEMENT

ABSTRACT

The objective of this project is to design and evaluate a machine learning framework for fraud detection using tabular transaction data. Fraud detection is a critical use case in financial risk management, e-commerce, and payment systems, where identifying anomalies in large-scale datasets can prevent financial loss. This project will involve training and evaluating classical machine learning models specifically Random Forest and Gradient Boosted Trees (GBT) to identify fraudulent transactions. Students will gain hands-on experience in data preprocessing, feature engineering and model evaluation.

1 INTRODUCTION

In the age of digital finance, online transaction fraud poses significant challenges for organizations. Traditional rule-based fraud detection systems struggle to scale with complex and evolving fraud patterns. Machine Learning (ML) offers a superior alternative by learning directly from data, allowing dynamic adaptation to new fraud vectors.

The goals of this project are:

1. To understand the differences between rule-based and ML-based fraud detection systems.
2. To train and evaluate classical ML models (Random Forest, Gradient Boosted Trees) for fraud classification.
3. To perform feature engineering and assess model interpretability and fairness.

2 BASELINE OR INITIAL ANALYSIS

The first step involves understanding and preparing the online transaction fraud dataset (sourced from Kaggle). Students will:

- Load, clean, and preprocess the dataset using Python (pandas, scikit-learn, NumPy).
- Analyze class imbalance (fraud vs. non-fraud) and determine appropriate sampling or weighting techniques.
- Perform exploratory data analysis (EDA) through visualizations showing transaction distributions, correlations, and fraud ratios.
- Compare rule-based vs. ML-based fraud identification to establish a baseline for improvement.

3 FINAL ANALYSIS

The final phase will involve the following components:

A. Model Development

- Train Random Forest and Gradient Boosted Trees (XGBoost) models for fraud classification.
- Use time-series splitting and cross-validation to ensure generalizable results.
- Apply techniques to handle class imbalance (weighted losses or undersampling).

B. Model Evaluation

- Evaluate models using AUPRC (Area Under Precision-Recall Curve), precision, recall, and F1-score.
- Compare model interpretability via feature importance plots and decision tree visualization.

C. Feature Engineering

- Engineer derived features to enhance predictive accuracy.
- Use correlation matrices to identify redundant or biased features.

4 Final Goals & Evaluation

The project will be successful if it demonstrates:

- Technical Accuracy: Models achieve measurable performance improvement (e.g., AUPRC > 0.75).
- Explainability: Clear articulation of how model predictions relate to transaction features.
- Communication: Comprehensive documentation of code, methodology, and findings, presented through visual dashboards and written reports.

5 Related Work

1. Fraud Detection Basics
   a. Dal Pozzolo, A. (2015). Calibrating Probability with Undersampling for Imbalanced Classification. A practical introduction to fraud detection challenges, especially class imbalance - perfect for beginners.
2. Imbalanced Learning
   a. He, H. & Garcia, E. (2009). Learning from Imbalanced Data. A clear overview of why fraud datasets are hard and how to handle imbalance (sampling, weighting, metrics).
3. Tree-Based Models
   a. Breiman, L. (2001). Random Forests. The original Random Forest paper, simple and accessible.
   b. Chen & Guestrin (2016). XGBoost: A Scalable Tree Boosting System. The foundational work behind gradient boosting; widely used in fraud modeling.
4. Explainability
   a. Lundberg & Lee (2017). SHAP: A Unified Approach to Interpreting Model Predictions. Great reference for model interpretation, essential for fraud analytics.

6 Data & Technical Requirements

Datasets:

- Kaggle Online Payments Fraud Detection Dataset- https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset

Software & Libraries:

- Python 3.10, pandas, scikit-learn, NumPy, XGBoost
- Matplotlib / Seaborn for data visualization

Tools for model explainability:

- SHAP or LIME for model explainability