Fraud Detection Engine Using Machine Learning

Simreteab Mekbib

Nov 16, 2025

Abstract

Financial fraud detection is a critical challenge for modern digital payment systems due to the extreme imbalance between fraudulent and legitimate transactions and the continuously evolving behavior of fraudsters. This project presents a Fraud Detection Engine that combines supervised machine learning models, unsupervised anomaly detection, and explainable artificial intelligence (XAI). Using a real-world inspired transaction dataset, multiple classifiers are trained and evaluated, class imbalance is explicitly handled, and SHAP explanations are used to provide transparency for non-technical fraud analysts. The system is deployed as a live web application suitable for real-time transaction monitoring.

1 Introduction

The rapid growth of digital payment platforms has increased exposure to financial fraud. Traditional rule-based systems are often insufficient due to their rigidity and inability to adapt to new fraud patterns. Machine learning approaches offer the ability to learn complex transaction behaviors and dynamically improve detection accuracy.

This project implements an end-to-end Fraud Detection Engine that identifies fraudulent transactions, explains predictions, and supports deployment in live financial systems.

2 Dataset Description

The dataset used in this project is the *Fraud Detection Dataset* obtained from Kaggle:

`https://www.kaggle.com/datasets/amanalisiddiqui/fraud-detection-dataset`

The dataset simulates mobile money transactions and contains labeled examples of fraudulent and legitimate activity.

2.1 Features

Key features include transaction type, transaction amount, sender and receiver balances before and after the transaction, and a binary fraud label (`isFraud`). Identifier variables and rule-based flags (`isFlaggedFraud`) were removed to prevent data leakage.

2.2 Class Imbalance

Fraudulent transactions account for less than 1% of the dataset, making class imbalance a major challenge that must be addressed during model training and evaluation.

3 Exploratory Data Analysis

Exploratory data analysis revealed that fraudulent transactions are highly concentrated in specific transaction types, particularly `TRANSFER` and `CASH_OUT`. Fraud cases are often associated with large transaction amounts and complete depletion of sender balances.

Two engineered features were introduced:

- `balanceDiffOrig`: Difference between sender balances before and after the transaction.
- `balanceDiffDest`: Difference between receiver balances before and after the transaction.

These features capture abnormal balance movement patterns characteristic of fraud.

4 Data Preprocessing

Numerical features were standardized using z-score normalization, while categorical features were encoded using one-hot encoding. A stratified train-test split (70% training, 30% testing) was applied to preserve the fraud ratio. All preprocessing steps were implemented using scikit-learn pipelines to ensure consistency during training and inference.

5 Supervised Machine Learning Models

Three supervised machine learning models were implemented and compared:

5.1 Logistic Regression

Logistic Regression was used as a baseline model due to its interpretability and efficiency. Class weights were adjusted to address class imbalance.

5.2 Random Forest

A Random Forest classifier was used to capture non-linear relationships and feature interactions. Balanced class weights improved recall for fraudulent transactions.

5.3 Neural Network

A Multi-Layer Perceptron (MLP) neural network was implemented to model complex fraud patterns. While effective, this model is less interpretable than linear and tree-based approaches.

6 Model Evaluation

Models were evaluated using metrics appropriate for imbalanced classification:

- Confusion Matrix
- Precision
- Recall
- F1-score
- ROC-AUC

Particular emphasis was placed on recall to minimize missed fraud cases while analyzing the trade-off with false positives.

7 Handling Class Imbalance

Class imbalance was handled using class-weighted loss functions. Model evaluation prioritized fraud-specific metrics over overall accuracy. Decision thresholds were adjusted to reflect real-world risk tolerance.

8 Anomaly Detection

To detect novel fraud patterns not present in labeled data, an Isolation Forest anomaly detection model was implemented. This unsupervised method identifies transactions that deviate significantly from normal behavior.

The anomaly detection output was combined with supervised predictions, allowing a transaction to be flagged if either method indicates high risk.

9 Explainable AI with SHAP

Explainability is critical in financial systems. SHAP (SHapley Additive exPlanations) was integrated to provide feature-level explanations for individual predictions. This enables fraud analysts to understand why a transaction was flagged, improving trust and regulatory compliance.

An example explanation provided to analysts is:

> "The transaction was flagged because the sender balance was fully depleted, the transaction type is high-risk, and the transaction amount is unusually large."

10 System Deployment

The Fraud Detection Engine was deployed as a live web application using the Streamlit framework and hosted on Streamlit Cloud. The deployed application is available at:

`https://fraud-detection-engine-app.streamlit.app/`

The application allows users to input transaction details, receive fraud predictions with probability scores, and view SHAP-based explanations in real time.

10.1 Live Transaction Flow

1. Transaction input
2. Feature engineering and preprocessing
3. Fraud probability prediction
4. Anomaly detection check
5. Alert generation or classification as legitimate

11 Model Maintenance and Adaptation

Fraud detection systems must evolve as fraudsters change tactics. Periodic retraining with recent data, monitoring for data drift, threshold recalibration, and analyst feedback integration are essential for maintaining system effectiveness. Unsupervised anomaly detection provides an additional defense against emerging fraud patterns.

12 Conclusion

This project presents a complete Fraud Detection Engine combining supervised machine learning, unsupervised anomaly detection, and explainable AI. The system achieves effective fraud detection performance while maintaining transparency and deployment readiness. The approach aligns with real-world financial risk management practices and provides a strong foundation for future enhancements.