PROJECT ABSTRACT

Credit Card Fraud Detection using XGBoost

Topic ID: K-1213 | Branch: AI & Machine Learning | Type: Software | Availability: Built-to-order

01. Title

Credit Card Fraud Detection using XGBoost — imbalance-aware fraud scoring on 284,807 card transactions.

02. Introduction / Background

Card fraud is a textbook imbalanced problem: in the public ULB dataset, 492 frauds hide among 284,807 legitimate transactions (0.172%). A model predicting 'legitimate' every time scores 99.83% accuracy and catches nothing — which is why accuracy is a meaningless metric here and the field optimizes area under the precision–recall curve instead.

This project builds the real solution: an XGBoost classifier with class-weighted loss, tuned on AUPRC, with the operating threshold chosen explicitly from the precision–recall trade-off — because every false positive blocks a genuine customer. Features V1–V28 are PCA-anonymized, so the model learns statistical shape rather than semantics, exactly as in production systems.

03. Problem Statement

Score each card transaction for fraud in milliseconds, catching the maximum fraud while keeping false alarms — blocked legitimate customers — acceptably low, under extreme class imbalance and drifting fraud tactics. The project must also make every decision auditable (which features drove the score) and select its operating threshold deliberately rather than defaulting to 0.5.

04. Objectives

- Train an XGBoost classifier on the ULB dataset with class weights (scale_pos_weight ≈ 577) and AUPRC tuning.
- Select the operating threshold from the validation precision–recall curve (design target: 0.91 recall at 0.94 precision).
- Build a live screening demo: transaction stream, per-transaction risk gauge and top-feature evidence.
- Visualize fraud structure: PCA-space scatter and hour-of-day fraud distribution.
- Evaluate once on a stratified holdout: confusion matrix, PR curve, per-threshold tables.
- Document imbalance theory, thresholding and a retraining playbook for the viva.

05. Proposed Methodology

- Load the ULB dataset (284,807 × 30 features: Time, Amount, V1–V28); stratified train/test split.
- Scale features; handle imbalance with XGBoost class weights (no naive oversampling).

- Train with early stopping; tune by cross-validated AUPRC — never by accuracy.
- Choose the operating threshold from the validation PR curve at the target recall/precision trade-off.
- Evaluate the holdout once: confusion matrix, AUPRC, per-threshold analysis for the report.
- Serve the saved model in the web demo with feature-level evidence per decision.

06. System Architecture / Working

Ingestion → scaling → class-weighted XGBoost (max_depth 6, early stopping) → threshold policy from the PR curve → scoring service → demo UI with risk gauge and evidence bars. The threshold policy is a first-class component: it converts the model's probability into allow/review/block decisions at the documented trade-off point.

Dataset: Kaggle 'Credit Card Fraud Detection' (ULB, Brussels) — 284,807 transactions, 492 frauds, Sep 2013. Task: binary classification under extreme imbalance. Model: XGBoost, tuned for AUPRC. Metrics: AUPRC, recall@precision, F1 — design targets, never claimed as measured.

07. Hardware / Software Requirements

- Hardware: standard laptop/desktop, 8 GB RAM; CPU training in minutes; ~9 ms inference per transaction.
- Software: Python 3.10+, XGBoost, scikit-learn, pandas, NumPy, Matplotlib, Jupyter.
- Demo: any modern browser; single-file web app runs offline after download.

08. Expected Outcomes

- Tuned imbalance-aware XGBoost model with inference code.
- Training notebook: class-weight study, PR-curve threshold selection, holdout evaluation (0.91 recall / 0.94 precision design targets).
- Live screening demo with per-transaction risk evidence.
- Report, PPT and viva Q&A; on AUPRC vs accuracy, class weights, thresholding and concept drift.

09. Applications

- Bank transaction-monitoring: real-time scoring before authorization.
- E-commerce fraud teams: review-queue prioritization by risk score.
- Payment gateways: merchant-level fraud screening as a service.
- Fintech risk engines: feature-store compatible scoring component.
- Insurance claim screening (adjacent tabular-fraud task with the same methodology).

10. Future Scope

- Graph features: card–merchant–device networks for ring-fraud detection.
- Online/incremental learning as fraud tactics drift.

- Real-time streaming deployment (Kafka + model server).
- SHAP-based explanations for analyst-facing evidence.
- Federated learning across banks without sharing raw transactions.