Fraud Transaction Detection Report

Title

Fraud Transaction Detection Using a Simulated Dataset

Author

Ishan ghosh
Email: ishanghosh0111@gmail.com
Date: March 18, 2025

1. Introduction

The increasing prevalence of financial fraud necessitates robust detection systems to protect customers and institutions. This report presents a machine learning-based approach to classify transactions as fraudulent or legitimate using a simulated dataset. The dataset, designed with specific fraud scenarios, provides a controlled environment to test and validate fraud detection techniques. The objective is to develop a model that accurately identifies fraudulent transactions while leveraging the dataset's simulated patterns, including high-value transactions, terminal compromises, and customer-specific fraud.

2. Dataset Description

The dataset is a simulated collection of 1,754,155 transactions from 183 files, containing original and fraudulent records. Key columns include:

TRANSACTION_ID: Unique transaction identifier.
TX_DATETIME: Date and time of the transaction.
CUSTOMER_ID: Unique customer identifier.
TERMINAL_ID: Unique terminal (merchant) identifier.
TX_AMOUNT: Transaction amount.
TX_FRAUD: Binary label (0 = legitimate, 1 = fraudulent).
TX_FRAUD_SCENARIO: Indicator of the fraud simulation scenario.

Fraud Scenarios

The fraud labels are simulated based on three scenarios:

1. **Scenario 1**: Any transaction with TX_AMOUNT > 220 is marked as fraudulent, serving as a baseline pattern.
2. **Scenario 2**: Two random terminals per day have all transactions fraudulent for the next 28 days, simulating terminal compromise (e.g., phishing).
3. **Scenario 3**: Three random customers per day have 1/3 of their transactions (over the next 14 days) multiplied by 5 and marked as fraudulent, mimicking card-not-present fraud.

These scenarios guide the feature engineering and model evaluation process.

3. Methodology

3.1 Data Preprocessing

The dataset was loaded from .pkl files, ensuring TX_DATETIME was parsed as a datetime object.
Duplicate columns were removed, and the data was sorted by TX_DATETIME for rolling feature calculations.
Invalid TX_DATETIME entries were dropped, resulting in 1,754,155 valid transactions.

3.2 Feature Engineering

Features were engineered to capture the fraud scenarios:

- **Base Features**: TX_AMOUNT, TX_TIME_SECONDS, hour, day_of_week.
- **Terminal Features**:
  - terminal_fraud_count: Cumulative fraudulent transactions per terminal.

- terminal_fraud_28d: Sum of frauds over a 28-day window (Scenario 2).
- terminal_fraud_28d_ratio: Ratio of frauds to total transactions over 28 days.

- **Customer Features**:
  - customer_avg_amount: Mean transaction amount per customer.
  - customer_amount_14d_avg: Mean amount over a 14-day window (Scenario 3).
  - amount_spike_14d: Ratio of TX_AMOUNT to customer_amount_14d_avg, flagging >5x spikes.
  - amount_spike_220: Binary flag for TX_AMOUNT > 220 (Scenario 1).
- **Additional Features**: amount_deviation, terminal_fraud_trend.

Customer and terminal IDs were encoded using LabelEncoder.

3.3 Model Selection and Training

- **Algorithm**: LightGBM, a gradient boosting framework, was chosen for its efficiency with large datasets.
- **Training Split**: 80% train (1,403,324 transactions), 20% test (350,831 transactions), with stratification.
- **Parameters**:
  - Objective: Binary classification.
  - Metric: AUC.
  - Learning rate: 0.03.
  - Scale pos weight: 47.39 (adjusted for class imbalance).
  - Early stopping: 100 rounds.
- **Threshold Tuning**: Evaluated at 0.79, 0.80, 0.81, 0.82, 0.83, and 0.84.

3.4 Evaluation Metrics

- Precision, recall, F1-score (per class), and macro-averaged metrics.
- ROC AUC score for overall performance.

4. Results

4.1 Classification Reports

The model was evaluated at six thresholds. Key metrics for the fraud class (1) are:

Threshold	Precision	Recall	F1-Score	Support
0.79	0.78	0.92	0.85	2936
0.80	0.79	0.92	0.85	2936
0.81	0.81	0.92	0.86	2936
0.82	0.82	0.92	0.87	2936
0.83	0.83	0.92	0.87	2936
0.84	0.85	0.91	0.88	2936

Best Threshold: 0.84, with an F1-score of 0.88, balancing precision (0.85) and recall (0.91).
Non-fraud class (0) consistently achieved 1.00 across all metrics due to the imbalance (347,895 vs. 2,936).

4.2 ROC AUC Score

- **Value**: 0.9857, indicating excellent discrimination between classes.

4.3 Feature Importance

A plot (feature_importance.png) highlights the top 10 features.
Expected key contributors include:
  - amount_spike_220 (Scenario 1).
  - terminal_fraud_28d (Scenario 2).
  - amount_spike_14d (Scenario 3).

5. Analysis

5.1 Performance Evaluation

The F1-score of 0.88 at threshold 0.84 suggests the model effectively detects fraud, with high recall (91%) ensuring most frauds are caught and reasonable precision (85%) minimizing false positives.
The ROC AUC of 0.9857 confirms the model’s robustness, exceeding the baseline expectation for simulated data.
The model aligns with Scenario 1 (high amounts), Scenario 2 (28-day terminal patterns), and Scenario 3 (14-day customer spikes), as reflected in the feature engineering.

5.2 Scenario-Specific Insights

- **Scenario 1**: amount_spike_220 should rank high, validating detection of transactions > 220.
- **Scenario 2**: The 28-day window for terminal_fraud_28d matches the PDF’s specification, likely improving terminal-based fraud detection.
- **Scenario 3**: The 14-day window and 5x spike detection (amount_spike_14d) align with the customer fraud pattern.

5.3 Limitations

- The dataset’s simulated nature may not fully reflect real-world complexities.
- Memory usage with 1.75M rows could be an issue; sampling (e.g., 10%) might be needed for scalability.

6. Conclusions and Recommendations

The developed model successfully classifies fraudulent transactions with an F1-score of 0.88 and ROC AUC of 0.9857, meeting the project’s objective. The feature engineering effectively targets the simulated fraud scenarios, with the 28-day and 14-day windows aligning with the PDF’s guidelines.

Recommendations

Hyperparameter Tuning: Adjust LightGBM parameters (e.g., num_leaves, learning_rate) to potentially improve the F1-score beyond 0.88.
Additional Features: Incorporate temporal patterns or terminal clusters to enhance Scenario 2 detection.
Real-World Validation: Test the model on real transaction data if available.
Deployment: Save the model (fraud_detection_model_optimized_tuned.txt) for integration into a production system.