

## FEATURE ENGINEERING AND EXPLAINABLE MACHINE LEARNING FOR CREDIT CARD FRAUD DETECTION: EVIDENCE FROM TEMPORAL TRANSACTION DATA

**Mahmudova Dilnoza Avaz qizi**

“Business Development Bank” IT, 1st Category Specialist

Email : [mahmudovadilnoza1999@gmail.com](mailto:mahmudovadilnoza1999@gmail.com)

### **Abstract:**

Credit card fraud detection is challenging because individual transaction attributes may not reveal whether current activity is unusual relative to previous behavior. This study evaluates an expanded feature representation for XGBoost under temporal holdout evaluation. Two identically configured models were compared on 1,852,394 transactions: a baseline using four original variables and an engineered model adding ten historical, temporal, geographic, demographic, and merchant-related features. Average Precision increased from 0.6762 to 0.9299, a difference of +0.2537 (paired class-stratified bootstrap 95% CI: [0.2349, 0.2712]). F1-score, MCC, precision, recall, ROC-AUC, and Recall@5% also improved. SHAP ranked transaction amount, seven-day historical average amount, transaction hour, 24-hour transaction frequency, and city population highest. Permutation analysis supported the predictive utility of the current-amount-to-historical-average ratio. Historical features excluded current and simultaneous transactions; test merchant risk used training labels only. The findings support richer feature representation in the evaluated setting. However, all ten engineered features were introduced jointly, preventing attribution of the total improvement to transaction-history features or any single variable. Single-holdout evaluation and unmodeled fraud-label delays limit generalization.

**Keywords:** credit card fraud detection; feature engineering; transaction history; XGBoost; explainable machine learning; SHAP; imbalanced classification.

### **Introduction**

Credit card fraud detection combines severe class imbalance with changing customer behavior, evolving fraud patterns, and delays in confirmed fraud label [1]. A transaction amount or category describes the current event but may not reveal whether it is unusual for the cardholder. Historical aggregation and behavioral profiling can supply this missing context [2]. Their value depends on constructing features only from information available when a transaction is scored and evaluating models on subsequent observations [3].

This study compares two identically configured XGBoost classifiers under a strict temporal holdout. Model A uses four original transaction and customer variables; Model B adds ten historical, temporal, geographic, demographic, and merchant-related variables. Keeping the

classifier and its configuration fixed focuses the comparison on the expanded representation. It does not isolate individual features because all ten additions are introduced jointly [4].

Two research questions guide the analysis: (1) To what extent does the expanded feature representation improve XGBoost fraud detection relative to basic transaction and customer attributes? (2) Which variables contribute most strongly to the engineered model's predictions and predictive performance? Average Precision (AP) is the primary metric, supplemented by ranking, threshold-dependent, and review-budget measures. Paired bootstrap resampling assesses uncertainty in the AP difference, while SHAP and permutation analysis examine fitted-model behavior without making causal claims [5].

## Materials and Methods

### 1. Related work and study design

Whitrow et al. showed that aggregating preceding transactions over different time windows can improve fraud detection. Correa Bahnsen et al. examined aggregation and periodic-behavior features. APATE combined transaction information, network-based extensions, and customer spending behavior, while Jurgovsky et al. found manually aggregated features useful in both sequential and non-sequential approaches. These studies motivate representations that relate each transaction to preceding activity.

Realistic fraud detection also requires attention to non-stationarity, concept drift, and verification latency. Reviews identify feature engineering, preprocessing, imbalance, and model selection as recurring challenges; related strategies have been studied for fund-transfer fraud. The present comparison holds the learner, hyperparameters, and evaluation periods constant, avoiding changes in these factors as competing explanations for the difference between representations.

Metric choice is equally consequential. Precision-Recall evaluation focuses on the rare positive class and complements ROC-based summaries. MCC incorporates all four confusion-matrix components. The study therefore combines AP with ROC-AUC, precision, recall, F1-score, MCC, and Recall@5%, rather than relying on accuracy.

### 2. Dataset and temporal split

The experiment used transaction records contained in fraudTrain.csv and fraudTest.csv. The files were combined and stably sorted according to trans\_date\_trans\_time, producing 1,852,394 chronologically ordered transactions.

An initial split point was defined at 75% of the combined observations. The boundary was then moved to the first occurrence of the corresponding timestamp to prevent transactions with identical timestamps from appearing on opposite sides of the split. A strict temporal condition was subsequently verified so that the final training timestamp preceded the first test timestamp.

The resulting training partition contained 1,389,295 transactions, including 7,891 fraud observations (0.5680%). It covered the period from 1 January 2019 00:00:18 to 23 July 2020 12:09:59. The test partition contained 463,099 transactions, including 1,760 fraud observations (0.3800%), and covered the period from 23 July 2020 12:11:54 to 31 December 2020 23:59:34.

All test observations therefore occurred after the training observations. Both model fitting and primary evaluation used their complete partitions; subsampling was restricted to the explainability analyses.

### 3. Feature construction

Model A contained amt, city\_pop, category, and gender. Model B retained these variables and added ten features: haversine\_distance\_km (straight-line geographic distance), age (customer age), hour, day\_of\_week, amt\_to\_city\_pop\_ratio (amount relative to city population), num\_txn\_last\_1h, num\_txn\_last\_24h, avg\_amt\_last\_7d, amt\_ratio\_to\_avg7d, and merchant\_risk\_score.

The same-card history variables used strictly preceding transactions within [t - window, t). The two counts covered 3,600 and 86,400 seconds, respectively; avg\_amt\_last\_7d averaged

amounts over 604,800 seconds. The current transaction and all transactions sharing its timestamp were excluded. The ratio `amt_ratio_to_avg7d` equaled current amount divided by `avg_amt_last_7d` when the historical mean was positive. An unavailable seven-day mean or a nonpositive denominator produced a missing value, subsequently handled by training-median imputation.

For the non-label-based card-history features, later test transactions were allowed to use transaction attributes from earlier test transactions. They could not use the fraud labels of those earlier test transactions. This represents sequential scoring in which observable transaction history is updated as transactions arrive.

#### 4. Merchant risk and leakage controls

Merchant risk used past fraud labels and required separate leakage controls. For a training transaction at time  $t$ , the smoothed score was:

$$\text{Score} = [F_m(<t) + 20 \times p(<t)] / [N_m(<t) + 20]$$

Here:  $F_m(<t)$  is the number of preceding fraudulent transactions for merchant  $m$ ,  $N_m(<t)$  is its preceding transaction count, and  $p(<t)$  is the fraud proportion among all training transactions strictly before  $t$ . The smoothing parameter was 20. Complete timestamp groups were excluded before calculating their scores, so neither the current label nor labels from simultaneous transactions entered the feature.

For a test transaction, the score was

$$\text{Score}(\text{test}) = [F_m(\text{train}) + 20 \times p(\text{train})] / [N_m(\text{train}) + 20]$$

This merchant mapping remained fixed throughout testing. Unseen merchants received  $p(\text{train}) = 7,891 / 1,389,295$ ; test labels were never used.

These rules exclude current-target, future-transaction, and test-label information under the specified design. They nevertheless assume that earlier confirmed fraud labels are available when historical merchant risk is calculated. Fraud-confirmation delays were not modeled, limiting operational interpretation.

#### 5. Preprocessing and model configuration

Missing numerical values were imputed using medians estimated from the training partition only. Categorical variables were one-hot encoded using the training data, with previously unseen test categories handled through `handle_unknown = 'ignore'`. Test observations were transformed using preprocessing parameters learned from training and were not used to fit preprocessing operations.

The resulting data matrices were represented as 32-bit floating-point values. After preprocessing, Model A contained 18 encoded columns and Model B contained 28 encoded columns. All training and test observations were retained.

Both models used XGBoost with `n_estimators = 200`, `max_depth = 5`, `learning_rate = 0.1`, `subsample = 1.0`, `colsample_bytree = 1.0`, `tree_method = hist`, `n_jobs = 4`, `random_state = 42`, `eval_metric = logloss`, and `scale_pos_weight = 1`. No class weighting, oversampling, undersampling, hyperparameter tuning, or early stopping was applied. Both models were fitted on the complete training partition.

#### 6. Performance evaluation

AP and ROC-AUC were calculated from continuous predicted probabilities. Precision, recall, F1-score, and MCC used the fixed decision rule: fraud = 1 if predicted probability  $y > 0.5$ , otherwise 0. No threshold optimization was performed.

For Recall@5%, test transactions were ranked by decreasing fraud probability and the highest-scoring  $k = \text{ceiling}(0.05 \times 463,099) = 23,155$  were selected. Recall@5% equaled the number of frauds in this set divided by 1,760. This is recall under a 5% review budget, not at a 5% false-positive rate.

#### 7. Paired bootstrap analysis

The AP difference was evaluated using 1,000 paired, class-stratified bootstrap repetitions with seed 42. In each repetition, the 1,760 fraudulent and 461,339 non-fraudulent test

observations were separately sampled with replacement. Identical sampled observations and multiplicities were used for both models.  $AP(\text{Model B}) - AP(\text{Model A})$  was calculated for each sample, and its 2.5th and 97.5th percentiles formed the 95% confidence interval.

Neither model was retrained. The interval therefore quantifies test-sample uncertainty for fixed predictions, excluding model-training uncertainty. Transaction-level resampling does not explicitly preserve dependence within cards or neighboring time periods.

#### 8. SHAP and permutation analyses

The fitted engineered model, Model B, was interpreted using SHAP based on the framework introduced by Lundberg and Lee. TreeExplainer was applied to a uniformly sampled set of 1,000 test observations, selected without replacement using random seed 42. The SHAP sample was not class-stratified.

Global importance was the mean absolute SHAP value for each feature across the 1,000 observations. This measures contribution magnitude to the fitted model output; it gives neither the direction of a feature's effect nor a causal estimate.

Permutation analysis used a separate class-stratified sample of 50,000 test transactions containing 190 frauds. Each selected feature was independently permuted three times using seed 42 while the fitted model and other features remained unchanged. AP decrease was defined as unpermuted sample AP minus permuted AP; the mean and standard deviation across the three repetitions were reported. These are fitted-model perturbations, not retraining-based feature ablations, and predictor correlations may affect them. All principal random operations used seed 42.

## Results

### 1. Predictive performance and uncertainty

Table 1 reports performance on all 463,099 test transactions. Model B outperformed Model A across every reported metric. AP increased from 0.6762 to 0.9299, an absolute difference of +0.2537; the paired class-stratified bootstrap 95% CI was [0.2349, 0.2712]. The interval was entirely above zero and applies to the fixed predictions under the implemented resampling procedure [6].

**Table 1.** Predictive performance of the baseline and engineered XGBoost models

| Metric                 | Model A:<br>Baseline | Model B:<br>Engineered | Difference (B - A) |
|------------------------|----------------------|------------------------|--------------------|
| Average Precision (AP) | 0.6762               | 0.9299                 | +0.2537            |
| ROC-AUC                | 0.9945               | 0.9982                 | +0.0037            |
| F1-score               | 0.5905               | 0.8805                 | +0.2900            |
| MCC                    | 0.5979               | 0.8810                 | +0.2832            |
| Precision              | 0.7106               | 0.9233                 | +0.2127            |
| Recall                 | 0.5051               | 0.8415                 | +0.3364            |
| Recall@5%              | 0.9756               | 0.9943                 | +0.0188            |

At probability >0.5, both precision and recall increased, alongside F1-score and MCC. Under the 5% review budget, Model A retrieved 1,717 of 1,760 frauds and Model B retrieved 1,750, placing 33 additional fraud cases among the top 23,155 transactions [7].

### 2. SHAP feature importance

Table 2 presents the five highest-ranked features by mean absolute SHAP. The original amount variable ranked first, followed by seven-day historical average amount, transaction hour, 24-hour transaction frequency, and city population. Thus, three engineered historical or temporal variables appeared among the top four[8].

**Table 2.** Top five features by mean absolute SHAP value

| Rank | Feature          | Mean absolute SHAP |
|------|------------------|--------------------|
| 1    | amt              | 1.9129             |
| 2    | avg_amt_last_7d  | 1.0670             |
| 3    | hour             | 0.5738             |
| 4    | num_txn_last_24h | 0.3109             |
| 5    | city_pop         | 0.2178             |

The ranking indicates that Model B used both original and engineered information. Because these are absolute magnitudes, the values do not show whether increasing a feature raises or lowers predicted fraud risk [9].

### 3. Permutation-based predictive utility

Unpermuted AP on the 50,000-transaction sample was 0.9236, compared with 0.9299 on the full test partition. Table 3 reports the three selected engineered variables and the descriptive variability across their three permutations.

**Table 3.** Permutation-based predictive utility of selected engineered features

| Feature               | Mean AP decrease | Standard deviation |
|-----------------------|------------------|--------------------|
| amt_ratio_to_avg7d    | +0.01844         | 0.00308            |
| amt_to_city_pop_ratio | +0.00108         | 0.00069            |
| haversine_distance_km | -0.00093         | 0.00027            |

Disrupting amt\_ratio\_to\_avg7d produced the largest AP decrease (+0.01844). The amount-to-city-population ratio had a smaller effect (+0.00108). The negative decrease for geographic distance (-0.00093) means sampled AP was marginally higher after permutation. These results describe the fitted Model B and do not estimate performance after removing features and retraining [10].

## Discussion

The controlled comparison answers the first research question: the expanded representation substantially improved predictive performance under the evaluated temporal holdout. The classifier configuration, training period, and test period were fixed, making feature representation the principal experimental distinction. However, all ten engineered features were introduced jointly. The total improvement cannot be attributed to transaction-history features alone or to any single engineered variable; estimating incremental contributions requires retraining-based ablation [11].

For the second research question, SHAP identified current transaction amount, recent spending levels, transaction timing, and recent frequency as prominent inputs to the fitted model. A seven-day average provides a behavioral reference that an absolute amount lacks. The AP reduction after permuting the current-to-historical amount ratio offers complementary evidence of predictive utility. This is consistent with earlier findings on transaction aggregation, periodic behavior, and sequential information, while remaining specific to the present model and evaluation [12].

The two importance analyses answer different questions. SHAP summarizes attribution magnitude; permutation assesses changes in predictive performance when selected values are

disrupted. Neither establishes causation or the isolated contribution of a feature to the overall Model A-Model B gap. Feature dependence and redundancy can affect permutation results, and only three repetitions were used. The small negative geographic-distance effect should therefore not be interpreted as evidence that distance is intrinsically harmful or protective [13].

The baseline already had ROC-AUC = 0.9945, leaving little numerical room for improvement despite its lower AP. AP distinguished the representations more strongly than ROC-AUC, consistent with the relevance of Precision-Recall analysis under severe imbalance [14]. At the fixed threshold, recall improved together with precision, rather than through a loss of precision. These results do not establish an operationally optimal threshold because neither decision costs nor threshold optimization were evaluated.

Recall@5% offers a complementary capacity-based view. Its smaller absolute improvement reflects the high baseline recall within this review budget, although Model B still brought 33 additional frauds into the selected set. A 5% review budget concerns the fraction of all transactions reviewed and must not be confused with a false-positive-rate constraint.

The leakage controls preserve chronology in historical aggregates, exclude simultaneous transactions, and freeze test merchant risk using training labels. Observable earlier test transactions can update non-label-based histories, consistent with sequential scoring. However, historical merchant risk assumes availability of prior confirmed labels. Verification latency is a recognized operational problem, and delayed labels could change the utility of this feature [15].

Generalization remains limited by a single dataset, one temporal holdout, and one fixed model seed and configuration. The bootstrap captures uncertainty in fixed test predictions, not alternative training samples, temporal periods, initialization, or parameter selection; card-level and temporal dependence are not explicitly preserved. SHAP relies on 1,000 uniformly sampled, non-stratified observations, and permutation uses only 190 frauds within its 50,000-observation sample. Its standard deviations are descriptive, not confidence intervals.

Substantive interpretation also requires care: `city_pop` is a population count, not population density, and `haversine_distance_km` is straight-line distance, not actual travel distance. These simplified contextual measures may help prediction without supporting broader claims about customer behavior or geography [16].

## Conclusion

Enriching four basic transaction and customer variables with ten engineered features improved XGBoost fraud detection under a strict temporal holdout. Model B achieved AP = 0.9299 versus 0.6762 for Model A, with a difference of +0.2537 and paired class-stratified bootstrap 95% CI [0.2349, 0.2712]. All other reported metrics improved. SHAP and permutation analyses identified historical spending and transaction-frequency information as meaningful components of the fitted model alongside original attributes.

The evidence supports the combined representation in this setting, not attribution of the entire gain to transaction history or any individual feature. Future evaluation should use additional datasets and temporal periods, model fraud-label delays, and apply retraining-based ablation. Alternative historical windows and operational thresholds also warrant investigation. Leakage-aware feature construction and explicit limits on explanatory claims are essential when translating these findings into fraud-screening practice.

## References

- [1] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, 'Learned Lessons in Credit Card Fraud Detection from a Practitioner Perspective', 2014.
- [2] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, 'Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy', 2018.
- [3] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, 'Transaction Aggregation as a Strategy for Credit Card Fraud Detection', 2009.
- [4] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, 'Feature Engineering Strategies for Credit Card Fraud Detection', 2016.
- [5] V. Van Vlasselaer *et al.*, 'APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection Using Network-Based Extensions', 2015.
- [6] J. Jurgovsky *et al.*, 'Sequence Classification for Credit-Card Fraud Detection', 2018.
- [7] J. Davis and M. Goadrich, 'The Relationship Between Precision-Recall and ROC Curves', 2006.
- [8] T. Saito and M. Rehmsmeier, 'The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets', 2015.
- [9] D. Chicco and G. Jurman, 'The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation', 2020.
- [10] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', 2016.
- [11] S. M. Lundberg and S.-I. Lee, 'A Unified Approach to Interpreting Model Predictions', 2017.
- [12] W. Hilal, S. A. Gadsden, and J. Yawney, 'Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances', 2022.
- [13] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine, 'Credit Card Fraud Detection in the Era of Disruptive Technologies: A Systematic Review', 2023.
- [14] Y.-Y. Hsin, T.-S. Dai, Y.-W. Ti, M.-C. Huang, T.-H. Chiang, and L.-C. Liu, 'Feature Engineering and Resampling Strategies for Fund Transfer Fraud with Limited Transaction Data and a Time-Inhomogeneous Modi Operandi', 2022.
- [15] V. Van Vlasselaer *et al.*, 'APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection Using Network-Based Extensions', 2015.
- [16] T. Saito and M. Rehmsmeier, 'The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets', 2015.