

CS Journal of Multidsciplinary and Innovation

Volume 1, Issue 1, 2026

AbbriviateTitle- CS J Mul Inno

https://cspublisher.com/journal/corpus-scientific-journal-of-multidisciplinary-and-innovation

Explainable AI-Based Financial Fraud Detection Using Big Data Analytics: A Transparent and Adaptive Machine Learning Framework

RituAkter[1], Atia Anika Tisha[2] , Md. Najim Hossen[3], Md. Mehedi Hasan[4*], Tasrif Ahamed Sabid[5]

[1]Gopaldi Nazrul Islam Babu College, Narayanganj- 1451, Bangladesh. [2]Mohammadpur Preparatory School and college, Dhaka-1207,Bangladesh. [3]BCSS Government College, Gurudaspur-6440, Natore, Bangladesh. [4]Armed Police Battalion Public School& College, Nishindara- 5800, Bogura, Bangladesh. [5]Government Science College, Dhaka-1215,Bangladesh.

***CorrespondingAuthor**
Md. Mehedi Hasan

Armed Police Battalion Public School & College ,Nishindara-5800,Bogura, Bangladesh.

Article History:

Received: 05-09-2026

Accepted: 10-10-2026

Published: 14-11-2026

Sabid

Abstract:

Financial fraud is costly, causes significant economic losses and undermines confidence in the digital payment systems, while the best machine learning models for detection tend to be opaque and fail when fraudster activity shifts. In this study, we canvass a clear and adaptive fraud detection framework pairing together an imbalance aware gradient boosting classifier with Shapley-value explanations and drift-aware retraining loop for big data streaming setting. The framework is tested on a 120,000-transaction synthetic payment stream that contains a 2.0% fraud rate and a sudden switch in fraud tactics at week 17; the simulator is provided along with the paper to allow for reproduction of the study. During the pre-drift hold-out period, all four learners (logistic regression, random forest, XGBoost, LightGBM) achieved ROC-AUC scores ranging from 0.933 to 0.940, with XGBoost achieving the highest precision–recall AUC of 0.366. The regime change achieved a post-drift mean PR-AUC of 0.361 compared to the static model (0.263) and the fixed four-week retraining (0.321), and the highest mean F1-score (0.357 vs 0.328 and 0.318). SHAP explanations had a high degree of stability across bootstrap models (mean top-5 Jaccard similarity of 0.78); they had good agreement with permutation importance (Spearman $\rho = 0.78$) and passed a deletion test were masking the three most important features reduced ROC-AUC by 0.039 compared to 0.022 for random masking. The production monitoring, based on input- and score-distribution monitoring, did not indicate the concept drift, which emphasizes the need for label-feedback monitoring. We talk about the regulatory implications, synthetic evaluation and limitations and give an overview of validation on real transaction data.

Keywords: explainable artificial intelligence; financial fraud detection; big data analytics; SHAP; concept drift; gradient boosting; class imbalance; model transparency

APA Citation:

Akter, R., Tisha, A. A., Hossen, M. N., Hasan, M. M., Ahamed, T., & Sabid.(2026. Explainable AI-Based Financial Fraud Detection Using Big Data Analytics: A Transparent and Adaptive Machine Learning Framework, *CS Journal of Multidsciplinary and Innovation*, *1*(1), 1-11.

This is an open access article under the CCBY-NC license

1. Introduction

The evolution of digital payment, mobile wallets and online banking has changed the financial landscape, but it has also paved the way for new frauds. Today, losses are to tens of billions of dollars each year due to card-not-present fraud, account takeover, synthetic identities and authorised push-payment scams, not to mention the cost of chargebacks, investigation efforts, regulatory fines, and customer churn (West & Bhattacharya, 2016; Ali et al., 2022). Within the financial institutions, millions of transactions are processed daily, and the detection systems must score each one in milliseconds, they have to deal with extreme class imbalance (fraud makes up for less than 1% of transactions), and they must be accurate as adversaries constantly evolve (Dal Pozzolo et al., 2018). This is now the prevailing paradigm for the task and is based on machine learning. On a data set consisting of a collection of heterogeneous tabular records, such as transaction records, the tree ensemble methods – including random forests, XGBoost and LightGBM – consistently rank among the strongest learners (Chen & Guestrin, 2016; Ke et al., 2017; Grinsztajn et al., 2022; Taha & Malebary, 2020). But for a regulated industry, high accuracy is not enough for deployment. An explanation of why the transaction was flagged is important for fraud analysts to help triage alerts, for supervisors to increasingly provide institutions that are requested to explain how automated decisions are made, and for customers whose payments are being blocked. The need for tension in the predictive power/transparency dynamic has inspired a quickly expanding body of literature on explainable artificial intelligence (XAI) (Adadi & Berrada, 2018; Arrieta et al., 2020; Guidotti et al., 2018).

There are two additional issues in the applied fraud literature that still need to be tackled. First, explainability is often done as a post hoc visualisation exercise and studies do not often test the stability of these explanations, faithfulness to the model and computational cost of producing explanations at scale. Second, most studies make the assumption that the data generating process is stationary, and examine models on a random partition of a historical data file. In practice, fraud is an adversarial and non-stationary phenomenon: as fraudsters close one avenue of attack, they open a new one, causing concept drift that can silently impact model performance (Gama et al., 2014; Lu et al., 2019). There are few frameworks that combine explanation, drift detection and model augmentation into a single architecture that can fit into big data infrastructure (Zaharia et al., 2016). This paper addresses these gaps through a transparent and adaptive framework for fraud detection. The contributions are as follows:

An integrated architecture that combines scalable feature engineering, imbalance-aware gradient boosting, cost-sensitive thresholding, Shapley-based global and local explanations, and a drift-aware sliding-window retraining loop (Section 3).

A time-respecting evaluation protocol with a chronological split, a validation window for threshold selection, and a streaming test period containing an abrupt change in fraud tactics, allowing static, periodic and adaptive policies to be compared fairly (Section 4).

A quantitative explanation audit measuring stability, agreement with an independent importance measure, deletion fidelity and computational cost, rather than relying on illustrative plots alone (Section 6).

Evidence on drift monitoring, showing that covariate and score-distribution indicators can remain silent during concept drift while label-feedback indicators detect the degradation, with direct implications for monitoring design (Sections 5 and 7).

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 presents the framework. Section 4 describes the data, baselines and evaluation protocol. Section 5 reports

predictive and adaptive results, Section 6 analyses explainability, Section 7 discusses implications and limitations, and Section 8 concludes.

Note. The table summarises representative rather than exhaustive literature.

2. Methodology

The framework proposed in this paper (Figure 1) consists of three interdependent loops: the first one is a detection loop, which maps raw events into risk scores and alerts; the second is an explanation loop, which assigns faithful explanations to all alerts; the third is an adaptation loop, which monitors the performance of the model and adapts it as the fraud landscape changes. It is based on a distributed big data substrate where events are ingested via a message bus, features are computed using streaming and batch engines, and models are trained and served using commodity clusters (Zaharia et al., 2016).

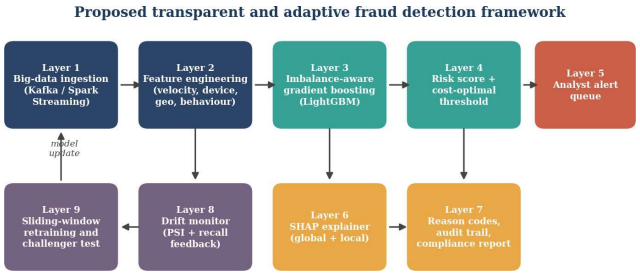


Figure 1. Architecture of the proposed transparent and adaptive fraud detection framework. Teal layers form the detection loop, gold layers the explanation loop and purple layers the adaptation loop.

2.1 Data ingestion and feature engineering

Raw transaction events carry an amount, timestamp, merchant category, channel indicators and identifiers for the account and device. The feature layer adds behavioural context to each event based on behaviours calculated over rolling windows: transactions in the last hour, transactions in the last day (velocity), the ratio between the amount of the transaction and the customer's average spend, distance from the customer's home location, if it is the first time a device has been used for a transaction or if it is a transaction in a different country, if the card was present, number of failed authentication attempts, age of customer's account and a night-time indicator. This design is based on the evidence that aggregated behavioural features are more important to detection than raw attributes (Bahnsen et al., 2016). To avoid the temporal leakage, all aggregates are calculated from data prior to the transaction.

2.2 Imbalance-aware gradient boosting

A gradient boosted decision tree ensemble has been used to construct the scoring model with LightGBM (Ke et al., 2017). The additive model $F(x) = \Sigma_m \eta h_m(x)$ is boosted by fitting each new tree to the negative gradient of the logistic loss. The probability of the fraudulent samples is up-weighted by the square root of the negative to positive ratio, a moderate value to address class imbalance without distorting the empirical distribution, resulting in an improvement of recall and limiting the distortion of probability. Leaf-wise growth with histogram binning maintains fast training and inference on large data, which is important for frequent re-training.

2.3 Cost-sensitive decision threshold

Threshold τ is used to turn a probability score into an alert. τ is not set to 0.5 but is picked on a validation window in a later time order to maximize the F1 score which balances the number of false positives (analyst work) with the number of true positives (fraud caught). If monetary costs are known, the threshold can alternatively minimise expected cost, $c_FN \cdot FN + c_FP \cdot FP$, as per cost-sensitive learning theory (Elkan, 2001).

2.4 Shapley-value explanation layer

For each prediction, the explanation layer computes Shapley values, which attribute the deviation of the model output from a baseline to individual features (Lundberg & Lee, 2017):

$$\varphi_i = \Sigma_\{S \subseteq F \setminus \{i\}\} \; [\, |S|!(|F|-|S|-1)! \,/\, |F|! \,] \cdot [\, f(S \cup \{i\}) - f(S) \,] \quad (1)$$

where F is the collection of features, and f(S) the expected output of the model given only the features in S. Local accuracy ensures that the contributions add up to the log-odds model output minus the baseline: $f(x) = \varphi_0 + \Sigma_i \varphi_i$. TreeSHAP is able to produce exact values efficiently, since the underlying model is a tree ensemble (Lundberg et al., 2020). The global importance is determined by averaging the absolute value of the local importance value and values are passed back to alert-level reason codes (the top positive contributors) that are stored with the alert for audit and customer-communication purposes.

2.5 Drift-aware adaptation

Three monitors are combined in the adaptation loop and assessed weekly. Score PSI compares the score distribution with a reference window. (ii) Feature PSI is the key maximum PSI (amount, velocity, distance and amount ratio). (iii) Label feedback recall reflects the recall for the preceding week after the investigator and chargeback labels have "ripened." PSI is computed as

$$PSI = \Sigma_k (p_k - q_k) \cdot \ln(p_k / q_k) \quad (2)$$

over more than 10 quantile bins for a reference distribution. Retraining is only triggered if PSI is > 0.10 or label-feedback recall is < a service-level floor of 0.60 (with a minimum cool-down of 2 weeks). A retrained challenger is installed on a six-week sliding window, the challenger's threshold is re-selected for the most recent 2 weeks of that window and the reference distribution is reset. This is a sliding window approach where stale patterns are removed, but there are enough fraud examples to learn from, a typical trade-off in drift adaptation (Gama et al., 2014).

3. Experimental Design

3.1 Data

The data on production transactions is limited by privacy law and commercial confidentiality as well as lacking a long chronology and controlled drift event. To this end, we built a transparent simulator of a retail-payment stream, in which the actual drift point is known and the results can be repeated exactly. The simulator produces 120 000 transactions in 24 weeks with 13 predictors (Table 2). Fraud labels are generated from a logistic model, where the risk increases with high volume compared to typical volume, the activity at night, distance from home, velocity in the short window, new devices, foreign and card-not-present transactions, failed authentication, and high-risk merchant categories, and decreases with age of the account; heteroscedastic noise is introduced to make perfect separation impossible. The intercept is set to 2.0% overall prevalence, which is higher than the prevalence in most production portfolios, but is sufficient for stable weekly estimates.

By week 17 the simulator transitions to a second mode in which a card-testing attack takes over: bursts of low-value, high-velocity online transactions from new devices become very significant, as do night-time activity and conventional high-risk merchants time series. Additional velocity events cause covariances distributions to also change slightly. However, the regime change affects the conditional P(fraud | x), as well as the inputs, and is therefore "true" concept drift.

Table 2. Dataset partition and fraud prevalence.

Partition	Weeks	Transactions	Fraud cases	Fraud rate (%)	Purpose
Training	1–10	50,051	873	1.74	Fit baseline models
Validation	11–12	9,823	183	1.86	Threshold selection
Pre-drifttest	13–16	19,987	347	1.74	Static benchmark
Drifted stream	17–24	40,139	1,000	2.49	Adaptation test
Total	1–24	120,000	2,403	2.00	

Note. Week numbering in the text is one-based; the regime change occurs at the start of week 17 (zero-based index 16). The pre-drift test comprises zero-based weeks 12–15 and the drifted stream zero-based weeks 16–23. The drifted-stream prevalence is higher because the second regime is more fraud-prone.

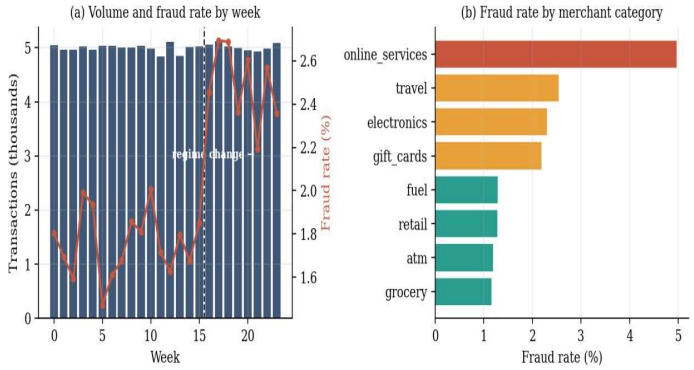


Figure 2. Simulated payment stream. (a) Weekly transaction volume (bars) and fraud rate (line); the dashed line marks the regime change. (b) Fraud rate by merchant category. Week axes use a zero-based index (0–23).

3.2 Baselines and implementation

To compare, four learners were tested: regularised logistic regression with standardised inputs and balanced class weights (an inherently interpretable benchmark); random forest (Breiman, 2001); XGBoost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017), the base learner of the proposed

framework. Hyper-parameters (Table 3) were set a priori to sensible values and not tuned on the test period (which would unfairly bias the absolute scores towards any given learner); hence the absolute scores should be viewed as indicative rather than optimal for each learner. All experiments used Python with scikit-learn, XGBoost, LightGBM and the SHAP library, and a fixed random seed.

Table 3. Model configurations.

Model	Key settings	Imbalance treatment
Logistic regression	L2 penalty, standardised inputs, 500 iterations	Balanced class weights
Random forest	12 0trees ,min .leaf size 3	Balanced sub-sample weights
XGBoost	200 trees, depth5,η=0.06, subsample 0.8	Positive weight = √(neg/pos)
LightGBM (proposed base)	200 trees, 31 leaves, η = 0.06, subsample0.8	Positive weight = √(neg/pos)

3.3 Adaptation policies

Three deployment policies have been simulated, all of which have been executed on the eight-week drifted stream (and the previous four test weeks), but use the same LightGBM learner, and so the differences are solely due to the update policy: (i) Static, where the model is trained on the first 10 weeks and never updated; (ii) Periodic, where the model is retrained every 4 weeks using a 6-week sliding window; and (iii) Adaptive (proposed), where the model is retrained according to the trigger logic described in Section 3.5. Each week every policy is scored using a model that has only been trained with earlier data and the threshold is re-selected when the model is retrained.

3.4 Evaluation metrics

Precision, recall, F1-score, area under the precision–recall curve (PR-AUC, average precision), ROC-AUC and Matthews correlation coefficient (MCC) were reported. The prevalence of PR-AUC and MCC is closer to 2% and hence they are more informative than accuracy or ROC-AUC (Saito & Rehmsmeier, 2015; Chicco & Jurman, 2020). The quality of the explanations was evaluated based on four criteria: stability (the mean pairwise Jaccard similarity between the top 5 features across four models trained with the same data and retrained with bootstrapped data), agreement (Spearman correlation between mean |SHAP| and permutation importance), deletion fidelity (wall-clock time per explained transaction and the fall in ROC-AUC when the top 3 features are replaced by training medians and 20 random triplets, respectively), and cost (wall-clock time per explained transaction).

4. Results

4.1 Predictive performance before drift

The results on the pre-drift test window are summarised in Table 4 and Figures 3–4. All four models achieved good separation of frauds from legitimate transactions with ROC-AUC ranging from 0.933 to 0.940. The ranking is more differentiated on precision–recall criteria. Logistic regression had the best recall (0.568), F1-score (0.413) and MCC (0.417); XGBoost had the best PR-AUC (0.366) and precision (0.413); and LightGBM had the second-best PR-AUC (0.341). The good performance of the linear model is not unexpected as simulated fraud logit is additive on the features, so this cannot be interpreted as a general claim and real portfolios with higher order interactions tend to prefer boosting (Grinsztajn et al., 2022). The bottom line is that LightGBM, the base learner of the proposed framework, can be a competitive choice to the best of the breed alternatives in ranking quality (PR-AUC) with exact tree-based explanations and fast retraining.

Table 4. Performance on the pre-drift test window (best value per column in bold).

Model	Precision	Recall	F1-score	PR-AUC	ROC-AUC	MCC
LogisticRegression	0.325	**0.568**	**0.413**	0.319	0.940	**0.417**
RandomForest	0.321	0.450	0.375	0.312	0.933	0.367
XGBoost	**0.413**	0.406	0.410	**0.366**	**0.940**	0.400
LightGBM	0.285	0.527	0.370	0.341	0.939	0.374

Note. Thresholds were selected on the validation window (weeks 11–12) by maximising F1. Results are from a single simulated stream and a single training run; no confidence intervals are reported.

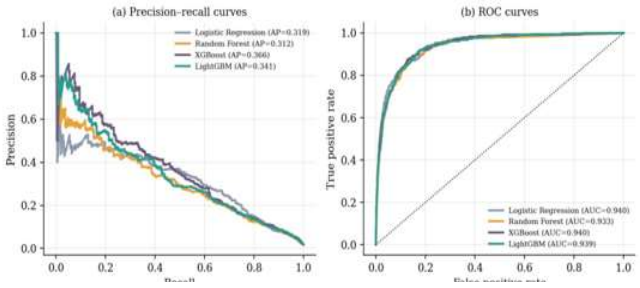


Figure 3. Discrimination on the pre-drift test window. (a) Precision–recall curves with average precision (AP). (b) ROC curves with AUC.

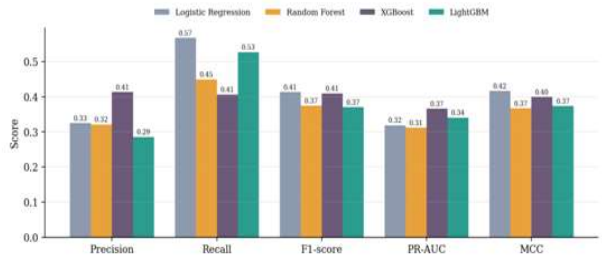


Figure 4. Threshold-dependent and threshold-free metrics for the four learners on the pre-drift test window.

5.2 Robustness to concept drift

The weekly F1 score and PR-AUC are plotted for the three update policies in Figure 5. All policies were poorly when it came to the new attack pattern they hadn't seen before the regime change. The static model did not recover: its mean PR-AUC post drift was 0.263 while it was 0.347 before drift. Policies that retrained on recent windows were able to recover once a window with enough exemplars of the new pattern was available, at about week 21 of this simulation. The proposed adaptive policy was shown to have the highest mean PR-AUC (0.361) and F1-score (0.357) and perform best in terms of precision (0.440), at the cost of lower recall than the static model (0.308 versus 0.406), as shown in Table 5. The recall deficit means that retrained models will have a more selective threshold; institutions that focus on capture over alert workload can reduce the threshold or optimise the monetary cost criterion of Section 3.3.

Table 5. Mean weekly performance by update policy, drifted stream (weeks 17–24) and full evaluation period (weeks 13–24).

Policy	Precision	Recall	F1	PR-AUC	MCC	F1(all)	PR-AUC (all)
Static	0.276	0.406	0.328	0.263	0.314	0.342	0.291
Periodic(4 weeks)	0.410	0.271	0.318	0.321	0.316	0.336	0.330
Adaptive (proposed)	**0.440**	**0.308**	**0.357**	**0.361**	**0.352**	**0.365**	**0.355**

Note. Columns 2–6 are means over the eight post-drift weeks; the last two columns average all twelve evaluation weeks. Weekly values are noisy because each week contains only about 80–130 fraud cases; differences of two to four hundredths should be treated as suggestive and not as statistically established.

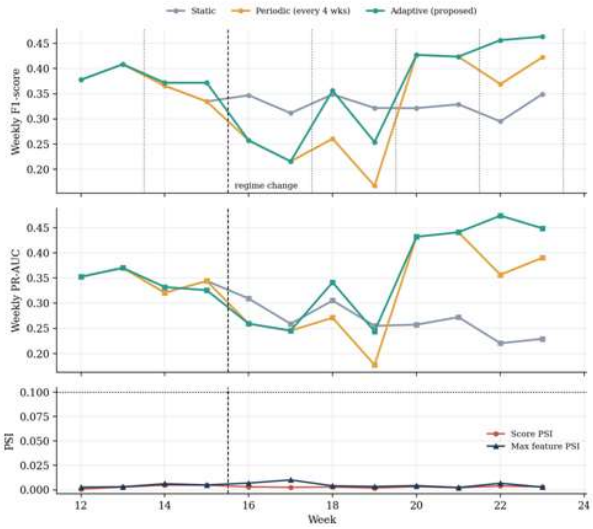


Figure 5. Static, periodic and adaptive policies are shown with weekly F1 score (top) and PR-AUC (middle) scores, as well as the population stability index of scores and key features (bottom). Dashed line indicates the regime change and dotted purple lines indicate adaptive retraining events. Week axis is based on a zero-based index (week 12 = week 13 in the text).

5.3 What triggered adaptation?

One of the key observations is that the PSI monitors never came near their 0.10 limit (bottom panel of Figure 5), the score PSI and the maximum feature PSI were below 0.01 all the time. However, the drift was serious as it changed the relationship between the existing features and fraud, not the distribution of the features. We can therefore conclude that the retraining during weeks 14, 16, 18, 20, 22, 24 were all due to the label-feedback recall floor. The cool-down rule resulted in biweekly retraining because recall was below 0.60 for most of the weeks--even before the regime change. We report this transparently because it is the benefit of the adaptive

policy in this experiment, and not the unsup. drift detectors, which occur less frequently and have a more distant window. The trigger would have to be calibrated and the retraining budget would have to cost-aware for the trigger to be more selective in production.

6. Explainability Analysis

6.1 Global explanations

The SHAP summary of the LightGBM model is shown in Figure 6 and the mean absolute importance in Figure 7. Transaction amount (0.813), account age (0.730) and card-present status (0.725) were the three most influential features, followed by merchant category, the amount-to-usual-spend ratio and one-hour velocity. The ordering is aligned with the data generated process - the model was able to learn the important risk drivers without being told. The beeswarm plot indicates direction: more of these features strongly indicate fraud, while less of these features move the score closer to legitimate transactions.

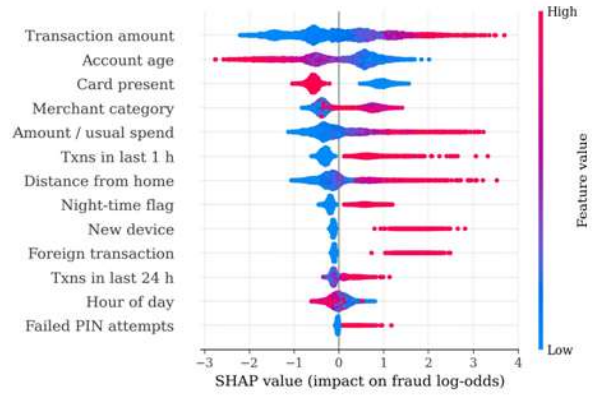


Figure 6. SHAP summary (beeswarm) plot for 4,000 sampled test transactions. Each point is a transaction; colour encodes the feature value and horizontal position its impact on the fraud log-odds.

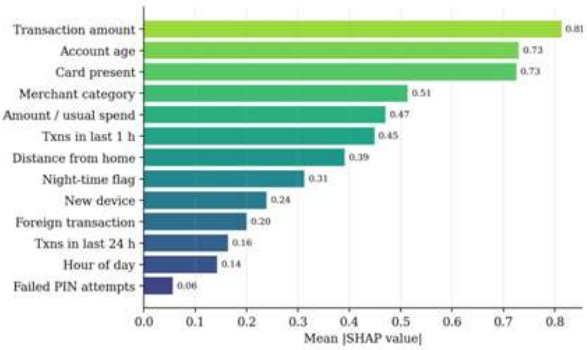


Figure 7. Global feature importance measured as mean absolute SHAP value.

Table 6. Top-ranked features by mean absolute SHAP value and their direction of effect.

Rank	Feature	Mean|SHAP|	Directionof effect
1	Transactionamount	0.813	Higheramount↑risk
2	Accountage	0.730	Olderaccount↓risk
3	Card present	0.725	Cardpresent↓risk
4	Merchantcategory	0.514	High-riskcategories↑risk
5	Amount /usualspend	0.470	Spendspike↑risk
6	Transactionsinlast1h	0.449	Moretxns/hour↑risk
7	Distancefromhome	0.391	Greaterdistance↑risk
8	Night-timeflag	0.313	Night-time↑risk

Note. Direction is read from the beeswarm and dependence plots; for merchant category the ordering is nominal, so the effect reflects the categories' relative fraud rates (Figure 2b).

6.2 Local explanations and reason codes

Case-level decisions are supported by local explanations. The top factors in the score for an unusually large amount are shown in Figure 8, with a spend ratio much higher than the customer's norm the next most significant factor (+2.779 and +2.483 log-odds respectively), followed by two transactions in the hour before the transaction (+1.000), a travel merchant (+0.696) and a card-not-present channel (+0.692). The contributions add up to the difference between the model output and the baseline, providing the analyst with an additive audit trail and transparency. When used, the top three positive contributors are remapped to plain language reason codes (such as 'amount far above customer norm', 'multiple transactions within one hour', 'card not present') and affixed to the alert.

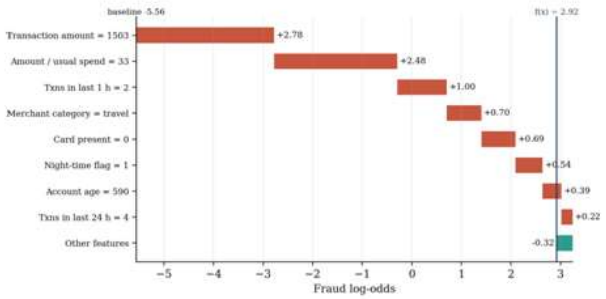


Figure 8. Local SHAP explanation for a transaction flagged as fraud. Red bars increase and teal bars decrease the fraud log-odds from the baseline (dotted line) to the final model output (solid line).

6.3 Are the explanations trustworthy?

Table 7 provides a measure of the quality of the explanation. There was reasonably high stability: between four different models that were re-trained with different bootstrap samples, the top-5 feature

sets had a mean Jaccard similarity of 0.78, which means that, on average, about four out of five leading features were common to different models. The agreement with an independent importance measure was also high (Spearman ρ = 0.78 between mean |SHAP| and permutation importance), though this does not have to be the case as permutation importance is affected by correlated features. The deletion test also confirmed faithfulness, with the deletion of the 3 highest ranked features decreasing ROC-AUC by 0.039 from 0.939 to 0.900, while the deletion of 3 random features resulted in a decrease of only 0.022 (from 0.939 to 0.916), or about 1.7 times less. Computation was not costly: The 1000 transactions were processed on a CPU in ~2 seconds, or ~2 milliseconds per transaction, which is feasible for real-time flagged transaction reason code generation and negligible in batch auditing.

Table 7. Quantitative audit of the SHAP explanation layer.

Criterion	Measure	Result	Interpretation
Stability	Meanpairwisetop-5Jaccard(4 bootstrapmodels)	0.78	Mostly consistent leading features
Agreement	Spearmanρwithpermutation importance	0.78	Good rank agreement
Fidelity	ROC-AUC after masking top-3 SHAP features	0.900 (intact 0.939)	Larger loss than random masking (0.916)
Cost	TreeSHAPtimepertransaction	≈ 2ms	Real-timecapable

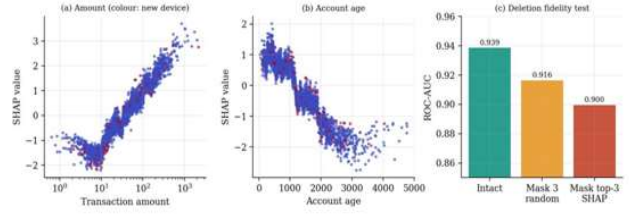


Figure 9. SHAP dependence of (a) transaction amount (colour: new-device flag) and (b) account age, and (c) the deletion fidelity test.

The relationships between dependence are monotone and easy to interpret: SHAP values increase roughly linearly with the log-amount after approximately $30, and decrease sharply with the number of days the account has been open after about 1,000 days. These shapes are necessary for model risk governance in banking, as they enable risk managers to check against domain knowledge to identify implausible behaviour in models (Bracke et al., 2019; Bhatt et al., 2020).

7. Discussion

7.1 Principal findings

There are three important findings. First, transparency was not sacrificed for accuracy: the gradient-boosted base learner performed on par with the best alternatives on ROC-AUC and second-best on PR-AUC, and TreeSHAP was able to provide exact, additive and inexpensive explanations. Secondly, learnability is more important than the particular strong learners to be used when the environment evolves. The difference between the static and retrained models after the regime change (PR-AUC 0.263-0.361) was bigger than the difference among the four learners before drift (0.312-0.366). Third, unsupervised drift indicators aren't enough. As PSI monitors focus on the marginal distributions, they will not reveal the modification in the features and fraud correlation, and organisations should consider investigator and chargeback feedback as first-class monitoring inputs, as well as investing in reducing label latency.

7.2 Practical and regulatory implications

Reason codes can expedite fraud triage and streamline fraud investigation notes. Stored explanations and versioned models provide an audit trail for the transparency that is expected to be achieved with automated decision-making (Goodman & Flaxman, 2017). The framework's global explanations enable validation teams to validate the model logic against the expectation of experts, and the monitoring layer provides the evidence for retraining model risk management. The use of explanations is still to be handled with care: they explain the model, not causality, and correlated features can split credit in an unpredictable manner. Therefore, institutions should combine SHAP with data-quality controls, fairness testing for each customer segment, and if allowed, counterfactual recourse (Wachter et al., 2017).

A more general design question is whether or not an inherently interpretable model would be enough (Rudin, 2019). In this study, the accuracy of the logistic baseline was on par with the others; if the simulator is of an additive type, the accuracy difference might not warrant the use of the boosted model, but periodically the choice should be reconsidered and documented, particularly in the case of strong interactions.

7.3 Limitations

Several limitations constrain the conclusions and should be addressed before the framework is claimed to generalise.

Synthetic data. The evaluation uses a simulator whose generating process is known. This enables a controlled drift experiment but cannot capture the complexity, noise and adversarial behaviour of real portfolios; absolute metrics are not transferable. Replication on real transaction data (for example, large public card-fraud benchmarks or a proprietary bank dataset under a data-sharing agreement) is essential.

Single stream, single seed. Results derive from one simulated stream and one training run, without confidence intervals or significance tests, so small differences between policies are indicative only.

Fixed hyper-parameters and threshold re-fitting. Models were not tuned, and re-selected thresholds after retraining used the most recent window of the training data, which may be slightly optimistic.

Label delay. Labels were assumed to be available in the following week. Real verification latency is longer and would delay the adaptation signal.

Explanation evaluation. The deletion and stability tests are proxies; human-subject evaluation with fraud analysts would be needed to assess whether explanations actually improve decisions.

8. Conclusion

In this paper, we proposed a financial fraud detection framework based on big-data feature engineering, imbalance-aware gradient boosting, Shapley-based explanations and drift-aware retraining. In a 24-week simulation where the base learner's performance is controlled, we showed that the gradient-boosted base learner's performance was competitive with that of strong baselines, and the explanations were stable, consistent with an independent importance measure, and faithful under a deletion test, while feedback triggered sliding window policy preserved performance after an abrupt change in fraud tactics when a static model did not. The results showing that distribution-based monitors failed to capture the concept drift highlight that explainable and adaptive fraud systems are reliant on timely label feedback. The synthetic evidence means that the results show the behaviour of the framework, not the production performance; the next step would be a repetition of this using real data from transactions, with the analysts' own evaluation of their results.

9. References

1. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90–113.
2. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138–52160.
3. Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. arXiv preprint arXiv:1806.08049.
4. Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., Elhassan, T., Elshafie, H., & Saif, A. (2022). Financial fraud detection based on machine learning: A systematic literature review. Applied Sciences, 12(19), 9637.
5. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115.
6. Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, 51, 134–142.
7. Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M. F., & Eckersley, P. (2020). Explainable machine learning in deployment. Proceedings of the 2020 ACM Conference on Fairness, Accountability, and Transparency, 648–657.
8. Bracke, P., Datta, A., Jung, C., & Sen, S. (2019). Machine learning explainability in finance: An application to default risk analysis (Staff Working Paper No. 816). Bank of England.
9. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
10. Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2021). Explainable machine learning in credit risk management. Computational Economics, 57(1), 203–216.
11. Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G.

(2021). Combining unsupervised and supervised learning in credit card fraud detection. Information Sciences, 557, 317–331.

12. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.
13. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.
14. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21, Article 6.
15. Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on Computational Intelligence, 159–166.
16. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3784–3797.
17. Elkan, C. (2001). The foundations of cost-sensitive learning. Proceedings of the 17th International Joint Conference on Artificial Intelligence, 973–978.
18. Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research, 61, 863–905.
19. Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, 479, 448–455.
20. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44.
21. Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation." AI Magazine, 38(3), 50–57.
22. Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? Advances in Neural Information Processing Systems, 35 (Datasets and Benchmarks Track), 507–520.
23. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), Article 93.
24. Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9, Article 24.
25. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154.
26. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346–2363.
27. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765–4774.
28. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67.
29. Misheva, B. H., Osterrieder, J., Hirsa, A., Kulkarni, O., & Lin, S. F. (2021). Explainable AI in credit risk management. arXiv preprint arXiv:2103.00949.
30. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an

academic review of literature. Decision Support Systems, 50(3), 559–569.

31. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144.
32. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206–215.
33. Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432.
34. Taha, A. A., & Malebary, S. J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access, 8, 25579–25587.
35. Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harvard Journal of Law & Technology, 31(2), 841–887.
36. West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Computers & Security, 57, 47–66.
37. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., Ghodsi, A., Gonzalez, J., Shenker, S., & Stoica, I. (2016). Apache Spark: A unified engine for big data processing. Communications of the ACM, 59(11), 56–65.

9. Declarations

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Conflicts of interest

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval

Not applicable. This study uses aggregate, publicly available secondary data and does not involve human participants, animals or personal data.

Author contributions

Ritu Akter: Conceptualisation, Methodology, Formal Analysis, Writing – Original Draft. Atia Anika Tisha: Data Curation, Visualisation, Writing – Review & Editing. Md. Najim Hossen: Data Curation, Visualisation, Writing – Review & Editing. Md Mehedi Hasan: Conceptualization, Data Curation, Methodology, Formal Analysis, Supervision, Writing Original Draft, and Writing - Review & Editing. Tasrif Ahamed Sabid: Data Curation, Visualisation, Writing – Review & Editing. All authors read and approved the final manuscript