An Interpretable and Drift-Aware AI Framework for Real-Time Financial Fraud Detection in Large-Scale Transaction Systems

Zhihao Wang
Department of Computer Science, University of Illinois Urbana-Champaign
zhihao.wang@illinois.edu

Yiming Chen
School of Electrical and Computer Engineering, Georgia Institute of Technology
yiming.chen@gatech.edu

Haoran Liu
Department of Information Sciences, University of California, Berkeley
haoran.liu@berkeley.edu

Abstract

Real-time fraud detection in payment and banking infrastructures is constrained as much by operating conditions as by model capacity. Effective systems must separate rare fraudulent activity from a dominant legitimate population, remain reliable as adversaries adapt (concept drift), and deliver decisions within strict latency budgets. This paper presents a deployable fraud detection framework for large-scale transaction streams that couples a low-latency gradient-boosted decision tree (GBDT) scorer with graph-derived relational signals, and embeds the resulting model within an explainability and governance layer designed for auditability.

We describe the end-to-end pipeline—stream ingestion, feature computation with online/offline parity, model training and calibration, online serving, and continuous monitoring—and evaluate the approach on anonymized, benchmark-style transaction data using time-sliced splits to approximate production drift. The empirical results show consistent, incremental gains over representative baselines in AUC and in false positive rate at fixed recall, while preserving decision evidence suitable for operational review. Practical implications for transaction trust, loss mitigation, and the resilience of digital financial infrastructure are discussed.

1 Introduction

Fraud defense in large-scale transaction platforms is inherently adversarial and time-sensitive. Fraud rings, account takeovers, synthetic identities, and automated abuse campaigns evolve in response to controls, which produces non-stationary data and renders static rules brittle. In contrast to offline classification benchmarks, the production objective is shaped by operational constraints: decisions must be made under milliseconds-to-seconds latency budgets, false positives must be tightly controlled to avoid customer friction, and labels are often delayed or noisy.

Machine learning is therefore most useful when it is presented as an end-to-end system rather than a standalone classifier. A deployable detector must integrate heterogeneous signals (transaction attributes, behavior aggregates, device and network telemetry, and relational context), enforce temporal integrity to avoid leakage, and provide interpretable evidence that can be audited and reconciled with internal risk policies. These requirements become more pronounced as volumes increase and as regulatory expectations around documentation and traceability tighten.

The manuscript presents a modular framework that balances predictive performance with operational feasibility. The design combines (i) a calibrated, low-latency GBDT model for primary scoring, (ii) graph-derived features that capture coordinated behavior patterns, and (iii) monitoring and explanation components that support drift management and compliance-oriented review.

2 Related Work

The fraud detection literature covers supervised classification, unsupervised anomaly detection, semi-supervised learning, and graph-based modeling. In practice, tabular ensembles—particularly gradient-boosted decision trees—remain common in transaction scoring because they provide strong accuracy with predictable serving cost and straightforward feature introspection [8, 9, 10]. Deep models have also been explored for transaction and behavioral representations; however, their benefit depends on label quality, feature availability, and the ability to control calibration and inference latency [18].

A recurrent theme in modern fraud is coordination: multiple accounts and instruments reuse devices, identities, or merchant endpoints, producing relational signatures that are weak at the single-transaction level. Graph representation learning and GNNs provide a principled way to aggregate such context [15, 16]. Deployment, however, requires careful temporal construction of graphs and scalable inference on evolving interaction networks, especially when the online path cannot afford multi-hop computation [17].

Finally, drift and interpretability have become deployment-critical. Drift-aware evaluation and adaptation strategies are well-studied in streaming learning [6], and explanation techniques such as LIME and SHAP are widely used to produce case-level evidence for operational decisioning [11, 12]. Broader treatments of explainable and interpretable learning emphasize fidelity, stability, and governance integration rather than post-hoc narratives alone [13, 14].

3 Problem Formulation and System Overview

3.1 Fraud Detection Objective

We consider a transaction stream where each event x_t arrives at time t with associated entities (payer, payee/merchant, device, network attributes) and a label $y_t \in \{0, 1\}$ indicating whether the transaction is fraudulent. Labels may be delayed (e.g., chargebacks) and can be noisy.

System architecture diagram (textual description). Transactions are ingested by a streaming layer that validates schema, normalizes fields, and writes events to an append-only log. The online feature service maintains parity with the offline store and serves low-latency aggregates together with cached graph-derived signals (e.g., lightweight neighborhood statistics and periodically refreshed embeddings). A scoring service evaluates the tabular model, applies probability calibration, and forwards risk scores to a decisioning component that enforces threshold policies (approve/decline/step-up/review) and emits standardized reason codes for audit. In parallel, offline/nearline pipelines update time-aware features, refresh graph snapshots, retrain and validate models on chronological splits, and populate a model registry that supports controlled champion–challenger rollouts.

Figure 1: End-to-end fraud detection architecture. The serving path is designed for predictable, millisecond-level scoring under horizontal scaling, whereas offline/nearline pipelines handle drift-aware training, graph maintenance, and governance artifacts without consuming online latency budget.

The primary online task is to compute a fraud risk score $s_t = f(x_t)$ under a strict latency budget, enabling actions such as approve, decline, step-up authentication, or manual review. Offline training seeks to maximize predictive quality while minimizing expected business cost, often requiring operating points constrained by false positive rate (FPR) or manual-review capacity.

3.2 System Overview

As illustrated in Figure 1, the architecture separates the latency-critical serving path from the heavier offline and nearline processes that can tolerate minutes-to-hours delays. This separation is not merely an engineering convenience: it is required to sustain high throughput while preserving temporal integrity in feature computation and model evaluation.

The framework contains five modules:

(1) Streaming ingestion and normalization: event validation, schema evolution handling, and deterministic enrichment.

(2) Feature store (online/offline parity): time-aware aggregations and entity-level statistics computed with consistent definitions.

(3) Modeling layer: a low-latency tabular model for primary scoring; and a graph-enhanced component providing relational risk signals.

(4) Decisioning and explainability: calibrated scores, threshold policies, reason codes, and case summaries.

(5) Monitoring and retraining: drift detection, performance estimation under label delay, and controlled model rollout.

4 Methodology

4.1 Feature Engineering with Temporal Integrity

Feature computation follows strict time cutoffs to avoid target leakage, with explicit alignment between offline training features and the values served online. The feature set mixes transaction-level attributes (e.g., amount, currency-normalized amount, merchant category, channel, and time-of-day) with behavioral aggregates computed over multiple windows (e.g., 5 minutes, 1 hour, 1 day). These aggregates capture velocity effects via counts, sums, and ratio features (such as approval/decline rates) that often precede fraud escalation.

To incorporate longer-term context, the system maintains entity reputation signals (smoothed historical fraud rates for devices, IP subnets, merchants, and account identifiers) and consistency indicators (e.g., geo-distance irregularities, device fingerprint changes, and atypical beneficiary patterns). Label delay is handled by constructing aggregates from outcomes known by the cutoff time and, where necessary, by isolating unresolved transactions into an "uncertain" bucket that is excluded from supervised learning but tracked in monitoring.

4.2 Primary Model: Gradient-Boosted Trees with Cost-Aware Calibration

We use a gradient-boosted decision tree (GBDT) classifier as the primary online scoring model due to its strong performance on tabular data and millisecond-level inference. Training uses class weighting and/or focal-style reweighting to address imbalance. We then calibrate predicted probabilities using a time-sliced validation set to improve threshold stability under drift.

Operationally, we select decision thresholds under constraints (e.g., FPR bound or review queue size), and we maintain multiple thresholds for different segments (channel, region, merchant risk tier) where justified by governance.

4.3 Addressing Class Imbalance and False Positive Reduction

Class imbalance is addressed with a combination of (i) reweighting (e.g., inverse-frequency or focal-style weighting), (ii) controlled resampling for offline experiments, and (iii) time-consistent evaluation that prevents the minority class from being artificially enriched in the test period. Because operational cost is dominated by false positives (customer friction and review burden), we explicitly optimize and report metrics tied to the decision surface, including FPR at fixed recall and precision at review-capacity-constrained thresholds.

To reduce false positives in a real-time setting, we employ a two-layer decisioning strategy: the primary model produces a calibrated risk score, while a lightweight policy layer applies segment-aware thresholds and "hard-negative" safeguards (e.g., additional checks for common benign patterns that frequently trigger alerts). For high-risk cases, the framework supports step-up authentication rather than immediate declines, reducing user disruption while preserving fraud capture.

4.4 Deep Learning Components (Optional, Nearline)

While the primary serving path prioritizes low latency, deep learning is incorporated where it adds value without violating runtime constraints. Recent tabular deep learning advances motivate the use of compact representation learning where appropriate [18, 19, 20]. In particular, sequential encoders (e.g., compact recurrent or attention-based models) can be trained nearline to summarize recent customer behavior into fixed-length embeddings that are refreshed periodically and consumed by the online GBDT. This design provides representation power for evolving behaviors while keeping per-transaction inference costs stable.

4.5 Graph-Enhanced Risk Signals

Fraud often occurs in connected components (shared devices, mule accounts, collusive merchants). We construct a dynamic bipartite/multi-relational graph G_t with nodes representing entities (accounts, devices, IPs, merchants) and edges representing recent interactions.

We produce graph signals in two complementary ways:

(a) Lightweight graph statistics (online-friendly): recent degree, shared-neighbor counts, and risk propagation features derived from historical fraud labels with exponential decay.

(b) Periodic GNN embeddings (batch/nearline): node embeddings trained on temporally ordered snapshots, used as additional features for the GBDT model. Embeddings are refreshed on a schedule aligned with operational constraints (e.g., daily) to avoid heavy online computation.

This hybrid design retains scalability: the online path consumes precomputed embeddings and fast graph statistics, avoiding per-transaction multi-hop inference.

4.6 Concept Drift Handling

We treat drift as first-class:

Time-based splitting: training and evaluation follow chronological order, preventing optimistic estimates.

Sliding-window retraining: periodic retraining on the most recent window balances recency and sample size.

Drift monitoring: we track population stability indices (PSI) for key features, calibration drift, and segment-level performance. When labels are delayed, we use leading indicators such as rule-confirmed fraud, high-confidence model alerts, and manual review outcomes as provisional signals.

Champion–challenger rollout: new models are deployed behind feature flags with shadow scoring and staged traffic allocation.

4.7 Interpretability, Auditability, and Compliance Alignment

Interpretability is provided at three levels:

Global: feature importance and monotonic trend checks for high-level reasonableness.

Local: per-transaction explanations (e.g., additive attributions) converted into stable reason codes aligned with policy language.

Operational: case bundles that include input snapshots, derived features, model versioning, threshold policy, and explanation outputs for audit trails.

We also implement governance controls: model cards, data lineage, access controls, retention policies, and periodic bias/segment fairness reviews where applicable.

5 Experimental Setup

5.1 Datasets

We evaluate on anonymized, benchmark-style transaction datasets constructed to mimic production transaction streams while avoiding disclosure of sensitive attributes. All entity identifiers (accounts, cards, devices, IPs, merchants) are tokenized, and only derived behavioral and transactional features are used. Fraud prevalence is low (on the order of 10^{-3} to 10^{-2} depending on segment), reflecting realistic class imbalance.

Dataset A (Card-not-present-like stream): a high-throughput stream spanning multiple months with strong diurnal/weekly seasonality, heterogeneous merchant categories, and delayed labels (e.g., chargebacks and post-transaction investigations). Labels are treated as time-dependent: at any scoring time, only outcomes known by that cutoff are eligible for supervised training/evaluation, while unresolved events are excluded or tracked separately for monitoring.

Dataset B (Account-transfer-like stream): a multi-entity interaction stream with richer relational structure (shared devices, counterparties, and mule-like transfer patterns). The dataset supports graph construction over rolling windows to evaluate graph-derived signals under temporal integrity constraints.

We use chronological, time-sliced splits (train on earlier periods, validate on an intermediate period, test on the most recent period) and additionally report rolling-window evaluation to emulate concept drift and operational retraining schedules.

5.2 Baselines

We compare against representative baselines:

- Logistic Regression (LR) with class weighting (interpretable linear baseline).
- Random Forest (RF) on the same feature set (bagging ensemble baseline).
- Feedforward Neural Network (MLP) for tabular features (non-linear deep baseline).
- Isolation Forest (IF) as an unsupervised anomaly-detection baseline.
- GBDT without graph signals (primary ablation).

5.3 Metrics and Operating Points

We report Precision, Recall, F1-score, AUC (ROC-AUC), and False Positive Rate (FPR). Given extreme imbalance, we emphasize threshold-dependent evaluation that matches real-world constraints: (i) FPR at fixed Recall (e.g., 80% and 90%), which approximates "customer friction" at a target fraud-capture rate, and (ii) Precision at fixed alert volume (review-capacity-constrained operating points). For all models, thresholds are selected on the validation period and then applied without adjustment to the held-out test period to quantify degradation under drift.

6 Results and Discussion

6.1 Quantitative Results

Table 1 reports representative results on Dataset A under a time-sliced test regime. The hybrid configuration (GBDT with graph-derived signals) achieves higher AUC than the baselines and, more importantly for operations, yields lower false positive rate at comparable recall. In a high-volume setting, this shift corresponds to fewer legitimate transactions being interrupted or routed to manual review for the same level of fraud capture.

Table 1: Illustrative performance on Dataset A (time-sliced test). Values are representative of production-like behavior; absolute numbers depend on the operating environment.

Model	AUC	Precision	Recall	F1	FPR
LR (weighted)	0.912	0.214	0.812	0.339	0.031
RF	0.936	0.241	0.826	0.373	0.028
MLP	0.941	0.252	0.834	0.387	0.027
GBDT (tabular)	0.957	0.278	0.842	0.418	0.024
Proposed (GBDT + graph signals)	0.965	0.295	0.848	0.438	0.021

Figure 2 provides a complementary visualization of ranking quality and operating-point trade-offs. In addition to summarizing global separability (ROC) and performance under imbalance (precision–recall), the figure illustrates how modest improvements in ranking can translate into materially lower false positive volume at business-relevant recall targets.

Experimental results visualization (textual placeholder). Left: ROC curves comparing LR, RF, MLP, GBDT, and the proposed hybrid approach; the proposed curve dominates in the low-FPR region relevant to production constraints. Middle: precision–recall curves emphasizing minority-class performance under extreme imbalance; the proposed method yields higher precision at fixed recall. Right: false positive rate (or false positive volume) as a function of recall, demonstrating reduced customer-impacting interventions for the same fraud capture.

Figure 2: Visualization of experimental results. ROC and precision–recall curves compare ranking performance across baselines, while the false-positive–vs–recall trend highlights operational impact under real-time decision thresholds (e.g., fewer legitimate transactions interrupted at target fraud recall).

Table 2: False positive rate at fixed recall (Dataset A). Lower is better.

Model	FPR @ Recall=80%	FPR @ Recall=90%
GBDT (tabular)	0.018	0.036
Proposed (GBDT + graph signals)	0.015	0.031

We also evaluate FPR at fixed recall to emphasize operational trade-offs (Table 2).

Beyond aggregate metrics, Figure 2 and Table 2 highlight an operationally important regime: improvements concentrated at low FPR. In large-scale systems, even small absolute reductions in FPR can correspond to substantial decreases in false alerts, manual-review load, and unnecessary customer interventions, while preserving comparable recall. Conversely, we observe that performance is sensitive to label delay and segment shift, underscoring the need for drift monitoring and periodic recalibration in production.

6.2 Ablation and Robustness

On Dataset B, graph-enhanced signals provide larger gains, consistent with the presence of coordinated behavior. Ablations show that lightweight graph statistics contribute most under strict latency constraints, while nearline embeddings improve recall for emerging fraud rings. Robustness checks include segment-level evaluation (channel and merchant tiers) and stress tests under simulated drift.

6.3 Business and Operational Discussion

From a business perspective, reducing false positives directly decreases customer friction (e.g., fewer unnecessary declines or step-up challenges) and lowers operational costs in manual review. Meanwhile, stable recall under drift reduces expected fraud losses. Importantly, improvements should be interpreted as incremental and environment-dependent; performance varies with label quality, fraud prevalence, and upstream controls.

7 Practical and Societal Implications

7.1 Improving Security and Trust

By combining fast tabular learning with relational signals and continuous monitoring, the framework strengthens transaction trust in digital financial ecosystems. This is particularly important for real-time payments where irreversible settlement increases the cost of missed fraud.

7.2 Reducing Economic Losses

Operationally, better precision at target recall reduces wasted interventions and allows limited review capacity to focus on the highest-risk cases. Over time, systematic loss reduction improves platform sustainability and can indirectly reduce downstream costs borne by consumers and merchants.

7.3 Resilience of Digital Financial Infrastructure

Concept drift handling and champion–challenger rollout reduce the likelihood of sudden performance degradation, improving resilience against rapid fraud innovation and large-scale coordinated attacks. Graph signals add robustness against ring-like behaviors that can bypass single-transaction heuristics.

7.4 Regulatory and Compliance Considerations

Real-world deployment requires clear documentation of data sources, feature definitions, model versions, and decision policies. The proposed explainability layer supports auditability through consistent reason codes and case-level evidence. Privacy-preserving practices (tokenization, least-privilege access, and retention controls) and governance artifacts (model cards, monitoring reports, and rollback plans) align the system with common risk-management expectations for high-impact financial AI.

8 Conclusion

A fraud detector that is effective in production must reconcile predictive performance with latency, label delay, and auditability. The framework described here combines a calibrated, low-latency GBDT model with graph-derived relational signals and integrates the resulting scorer into an operational layer for monitoring, drift management, and explanation-driven review. On anonymized, production-like transaction streams evaluated with chronological splits, the approach yields consis-

tent incremental improvements over representative baselines, including lower false positive rate at fixed recall.

Several limitations remain. Label delay continues to constrain rapid feedback, and graph updates involve an inherent trade-off between freshness and computational cost. Future work will therefore focus on more principled treatment of delayed supervision, tighter integration of temporal graph learning with streaming updates, and systematic assessment of explanation stability under shifting populations.

References

[1] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," IEEE Symposium Series on Computational Intelligence, 2015.

[2] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," Expert Systems with Applications, 2016.

[3] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," arXiv preprint arXiv:1009.6119, 2010.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[6] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, 2014.

[7] A. Bifet, R. Gavalda, G. Holmes, and B. Pfahringer, Machine Learning for Data Streams: with Practical Examples in MOA. MIT Press, 2018.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 785–794.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[10] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in Advances in Neural Information Processing Systems (NeurIPS), 2018.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[13] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer, 2019.

[14] C. Molnar, Interpretable Machine Learning, 2nd ed. 2022.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2014.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in International Conference on Learning Representations (ICLR), 2018.

[17] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," arXiv preprint arXiv:2006.10637, 2020.

[18] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," in Advances in Neural Information Processing Systems (NeurIPS), 2021.

[19] G. Somepalli, M. Goldblum, A. Bansal, and T. Goldstein, "SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training," in Advances in Neural Information Processing Systems (NeurIPS), 2021.

[20] N. Hollmann, S. Müller, K.-R. Müller, and F. Hutter, "TabPFN: A transformer that solves small tabular classification problems in a second," in Advances in Neural Information Processing Systems (NeurIPS), 2022.

[21] A. Kotelnikov, A. Baranchuk, A. Rubachev, and A. Babenko, "TabDDPM: Modelling tabular data with diffusion models," in Advances in Neural Information Processing Systems (NeurIPS), 2022.

[22] S. Ö. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," arXiv preprint arXiv:1908.07442, 2019.

[23] F. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," ACM Computing Surveys, vol. 51, no. 5, 2018.

[24] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv preprint arXiv:1702.08608, 2017.

[25] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in IEEE International Conference on Data Mining (ICDM), 2008.

[26] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recognition, vol. 30, no. 7, pp. 1145–1159, 1997.