

Explainable Graph Neural Fraud Detection for Real-Time Digital Payment Networks Using Temporal Neighborhood Inference

Raylen Mintarya^{1,*}, Nicholas Felix Chandra²

^{1,2}Department of Information Systems, Faculty of AI and Data Science, Universitas Pelita Harapan

ABSTRACT

Real-time fraud detection in digital payments must balance high detection accuracy, strict latency constraints, and audit-ready explanations. This study proposes an explainable graph-based framework that models payment activity as a temporal, attributed interaction network and performs transaction-level (edge) risk scoring using bounded K-hop neighborhoods within a sliding time window. Evaluation was conducted under a time-ordered protocol to prevent label leakage and to reflect delayed fraud confirmation. The dataset exhibits extreme class imbalance typical of production payment systems (fraud prevalence <1%), and performance is reported using ranking, capacity-aligned, and calibration metrics. Across baselines and graph models, results show that relational learning consistently improves decision-relevant performance. Compared with feature-only models, the proposed explainable graph neural approach achieved the strongest discrimination and alert quality, reaching AUROC = 0.985 and AUPRC = 0.532, outperforming XGBoost (AUROC = 0.972, AUPRC = 0.418) and logistic regression (AUROC = 0.948, AUPRC = 0.312). Under fixed investigation capacity, the proposed method attained Precision@200 = 0.66 and Recall@200 = 0.31, exceeding attention-based GAT (Precision@200 = 0.62, Recall@200 = 0.29) and GraphSAGE (Precision@200 = 0.58, Recall@200 = 0.27). Probability reliability also improved, yielding ECE = 0.038 versus 0.049 (XGBoost) and 0.062 (logistic regression), supporting safer threshold selection for operational actioning. Real-time feasibility was maintained through incremental updates and localized subgraph inference: the proposed system recorded p95 inference latency = 24 ms, remaining within practical authorization-time constraints while delivering the best AUPRC–latency tradeoff among compared methods. Explainability outputs (subgraph rationales and ranked feature drivers) enabled analysts to trace alerts to recent high-salience counterparties, burst-transfer patterns, and device/session inconsistencies, improving triage readiness and audit defensibility.

Keywords Graph Neural Networks, Fraud Detection, Digital Payments, Temporal Graph Learning, Explainable AI, Real-Time Inference, Transaction Networks, Model Calibration

Submitted: 25 November 2025

Accepted: 15 January 2026

Published: 12 August 2026

Corresponding author
Raylen Mintarya,
1081230004@student.uph.edu

Additional Information and
Declarations can be found on
[page 205](#)

© Copyright
2026 Mintarya and Chandra

Distributed under
Creative Commons CC-BY 4.0

INTRODUCTION

Digital payment ecosystems have become a core layer of modern commerce, enabling instant transfers, merchant payments, and platform-based financial services at massive scale. As transaction velocity and user reach expand, fraud has evolved into an adaptive, adversarial process that exploits real-time authorization windows, weak identity signals, and cross-platform coordination. Practical fraud programs therefore face a dual mandate: they must detect malicious activity quickly enough to prevent losses while minimizing customer friction and false declines. These pressures make real-time fraud detection a high-stakes learning problem operating under strict latency, reliability, and governance requirements [1], [2].

Despite extensive industrial adoption, many deployed defenses still rely on expert rules and heuristic thresholds because they are easy to reason about and easy to audit. However, rule-based systems struggle with coverage and maintenance: they are brittle under tactic shifts, require continuous manual updates, and often fail against coordinated behaviors where individual transactions look normal but collectively form suspicious patterns. Feature-based machine learning improves ranking performance but is still constrained by what can be expressed in tabular features at decision time, especially when fraud manifests as multi-entity interaction structures (e.g., mule chains, dense rings, rapid cash-out routes). This creates a persistent gap between what traditional models can represent and what fraud actually looks like in networked payment activity [3], [4].

A central challenge is that real payment networks are inherently relational. Accounts, devices, merchants, and sessions interact in ways that create multi-hop dependencies, and these dependencies can be more diagnostic than any single transaction attribute. Graph learning is therefore a natural fit, but many graph-based fraud approaches remain difficult to operationalize because they assume batch processing, full-graph recomputation, or unconstrained neighborhood expansion that violates real-time latency budgets. Furthermore, naive graph construction can introduce label leakage when evaluation is not aligned with the delayed and time-dependent nature of fraud confirmation (e.g., chargebacks and investigations), producing overly optimistic results that do not translate to live systems [5], [6].

In parallel, payment organizations increasingly require explanations that are defensible to auditors, regulators, merchants, and internal risk governance. High-performing black-box models are insufficient when decisions must be justified, appealed, or used to trigger downstream interventions such as holds or step-up authentication. Many existing explainability techniques are post-hoc add-ons that provide generic feature importance but fail to identify the transactional substructures that substantively support a fraud hypothesis (e.g., which counterparties and paths are implicated). The result is a practical trust gap: models may score accurately, but analysts cannot rapidly validate or action the rationale, slowing incident response and degrading operational acceptance [7], [8].

This study addresses these limitations by framing fraud detection as real-time edge risk scoring on a temporal, attributed payment graph, where each incoming transaction is evaluated using bounded, recent relational context. The research goal is to produce a deployable model that is simultaneously (i) accurate under extreme class imbalance, (ii) stable under concept drift, (iii) latency-feasible in streaming authorization flows, and (iv) explainable in a form aligned with investigation workflows. Concretely, the approach integrates incremental graph updates, constrained K-hop neighborhood extraction within a temporal window, and an attention-based graph neural network (GNN) that can surface relational evidence contributing to each alert [9], [10].

The methodological novelty lies in unifying real-time graph inference with multi-level explainability as first-class design constraints rather than afterthoughts. Instead of treating interpretability as a separate reporting step, the model produces explanations directly tied to local subgraphs and measurable drivers (e.g., velocity, device inconsistency, counterparty diversity) that analysts can

verify. In addition, the evaluation protocol is designed to reflect production conditions, emphasizing time-based splits, operational metrics (e.g., precision at fixed alert capacity), and calibration behavior required for safe thresholding. This ensures the system is assessed not only for ranking quality but also for deployability and governance readiness [11], [12].

Accordingly, the contributions of this research are threefold. First, it proposes a temporal graph construction strategy and streaming feature update mechanism that preserves the relational signatures of coordinated fraud while remaining computationally bounded for real-time scoring. Second, it introduces an explainable GNN-based scoring framework that provides transaction-level risk probabilities alongside subgraph rationales and feature-level drivers to support audit and triage. Third, it establishes an evaluation approach aligned with real-world label delays and drift, connecting detection performance to actionable outcomes and operational constraints in digital payment networks. Together, these contributions advance fraud detection from isolated, feature-centric scoring toward trustworthy, network-aware decisioning that can be deployed at scale [13], [14].

Literature Review

Fraud detection in digital payments has traditionally been dominated by rule engines and scorecards built from expert knowledge, largely because they offer deterministic behavior and clear compliance narratives. However, the literature consistently reports that hand-crafted rules degrade under adversarial adaptation: fraudsters learn thresholds, rotate identifiers, and exploit operational blind spots faster than policy updates can be deployed. As a result, rule-centric pipelines tend to accumulate complexity while losing generalization, motivating the shift toward statistical and machine learning-based approaches that can learn patterns from historical data and refresh with new evidence [15], [16].

Feature-based machine learning has become the mainstream replacement for purely heuristic controls, especially using tree ensembles and margin-based classifiers due to their strong performance on structured tabular signals. Prior studies emphasize that supervised fraud models are heavily shaped by the extreme class imbalance of confirmed fraud labels, which makes threshold selection and ranking quality more important than accuracy. Consequently, modern fraud research prioritizes precision–recall behavior, cost-sensitive learning, and ranking-aligned evaluation because operational teams must act on a small number of alerts within constrained investigation capacity [17], [18].

A persistent limitation of tabular approaches is representational: fraud in payment platforms is rarely independent across events. Accounts, devices, merchants, and sessions form interaction networks in which multi-hop dependencies can be the strongest evidence of coordinated abuse. Earlier graph analytics work demonstrated that network structure—such as dense clusters, short cycles, and repeated intermediaries—exposes organized fraud even when single transactions appear normal. This shift from isolated event modeling to relational modeling is the foundation for graph-based fraud detection, where the learning problem is naturally expressed on nodes and edges rather than on independent rows [19], [20].

Graph neural networks (GNNs) extend classical graph mining by learning

representations through message passing and neighborhood aggregation, enabling the model to combine attribute evidence with relational evidence. The literature reports that GNNs are especially effective in fraud settings where the signal is distributed across local neighborhoods, such as money-mule networks, collusive merchant activity, or account takeover patterns that propagate through shared devices and sessions. Attention mechanisms, edge-conditioned aggregation, and heterogeneous graph modeling are frequently used to capture the asymmetry and multi-typed nature of payment entities, improving the model's ability to isolate which relational channels carry the strongest risk signal [21], [22].

Temporal dynamics are increasingly central in recent fraud research because the value of evidence decays rapidly as attackers operate within short windows. Temporal graph learning addresses this by integrating time encodings, decay-weighted aggregation, or event-driven updates, allowing models to emphasize recency and detect burst-like attacks without over-relying on stale history. In addition, streaming constraints create a deployment gap: methods that assume full-graph recomputation or unconstrained neighbor expansion often fail in real-time authorization settings. As a result, many studies propose bounded neighborhood sampling, sliding time windows, and incremental feature updates to control latency while preserving short-range temporal signatures that are most predictive in practice [23], [24].

Explainability has become a parallel research stream due to regulatory expectations and operational requirements in financial decisioning. Traditional explainable ML in fraud often focuses on feature importance and local attributions, but graph-based decisions require additional structure: analysts need to know which counterparties, transactions, and multi-hop paths substantively support an alert. Recent work in explainable GNNs therefore emphasizes subgraph rationales, fidelity-sparsity tradeoffs, and attribution methods that can identify both influential node/edge evidence and the feature drivers that interact with that evidence. This literature motivates treating explanations not as cosmetic outputs but as artifacts that must align with investigation workflows and audit narratives.

Finally, evaluation methodology is widely recognized as a major source of overestimation in fraud detection, particularly when label availability is delayed and when splits do not respect time. The literature increasingly recommends time-based evaluation protocols, calibration checks, and business-cost metrics to validate whether predicted probabilities can be safely operationalized under fixed intervention policies. In production settings, model governance depends on score stability, alert volume predictability, and monitoring for concept drift, which together shape how fraud systems are maintained over time. These themes directly motivate an approach that jointly optimizes detection quality, latency feasibility, and explanation usefulness under realistic temporal constraints.

Method

Data Sources, Transaction Semantics, and Graph Formalization

This study models real-time digital payment activity as a dynamic, attributed graph built from streaming transaction events. Each transaction event is represented as a directed transfer from a sender entity to a receiver entity,

accompanied by metadata such as timestamp, amount, channel, device fingerprint, merchant category, geolocation proxy, and session identifiers. We treat entities as nodes (e.g., user accounts, cards, devices, merchants) and transactions as edges, enabling the model to capture multi-hop relational fraud patterns (e.g., mule rings, rapid cash-out chains, and account takeover propagation).

Figure 1 summarizes the operational workflow for explainable fraud detection in real-time payment networks. The pipeline begins with streaming transaction ingestion, where each incoming event is immediately transformed into structured features and used to update a temporal graph representation. This graph formalization is essential because payment fraud frequently manifests as relational behavior (e.g., coordinated rings, mule chains, rapid multi-hop transfers) that cannot be reliably detected from isolated transaction fields alone.

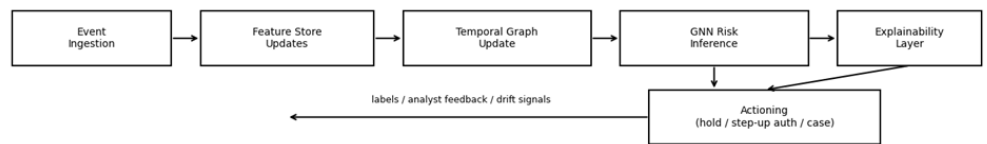


Figure 1 End-to-End Explainable Graph-Based Fraud Detection Pipeline

The diagram also highlights the separation between risk inference and explainability. The GNN inference block produces a low-latency probability score for decisioning, while the explainability layer is invoked when the score crosses operational thresholds or during audit sampling. Actioning (e.g., holding a payment, triggering step-up authentication, or creating a case) is fed back into the system as labels and monitoring signals, enabling continuous refinement of features, calibration, and drift detection without breaking real-time constraints.

Formally, we define a temporal payment network as $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t), X(t), A(t))$, where $\mathcal{V}(t)$ is the set of nodes observed up to time t , $\mathcal{E}(t)$ is the set of directed edges (transactions), $X(t) \in \mathbb{R}^{|\mathcal{V}(t)| \times d}$ are node features, and $A(t)$ stores edge attributes (amount, time gaps, channel, etc.). For a transaction edge $e = (u \rightarrow v)$ arriving at time t_e , we attach an edge feature vector a_e and optionally update node states using event-driven aggregation.

To align with operational fraud labels, we define a supervised target $y_e \in \{0,1\}$ for each transaction edge (fraud/legit) and optionally y_v for node-level compromise indicators. Because labels can be delayed (chargeback windows, post-investigation confirmation), training data is constructed using a cut-off horizon t_{cut} that prevents label leakage from the future.

$$\mathcal{D}_{\text{train}} = (e, \mathcal{G}(t_e), y_e) \mid t_e \leq t_{\text{cut}}, y_e \text{ available by } t_{\text{cut}} + \Delta \quad (1)$$

Here Δ is a conservative label-availability buffer; it is explicitly incorporated to reduce optimistic bias in reported performance for real-time deployment.

Temporal Graph Construction and Feature Engineering

The real-time constraint requires the graph to be updated incrementally as new transactions arrive. We maintain a sliding temporal neighborhood for each node

to bound latency and memory while preserving the short-term patterns most predictive of fraud (e.g., burst transfers, rapid merchant hopping). Instead of recomputing full-graph embeddings, we update only affected local subgraphs around the edge endpoints u, v, v, u, v for each incoming event.

Figure 2 illustrates the computational strategy used to satisfy real-time constraints: instead of running inference on the entire payment graph, the model extracts a local subgraph around the new transaction endpoints u and v . This local region is defined by a hop limit K and a temporal window Δ , which together ensure that inference focuses on the most relevant recent interactions (e.g., burst transfers, sudden changes in counterparties) while controlling latency and memory usage.

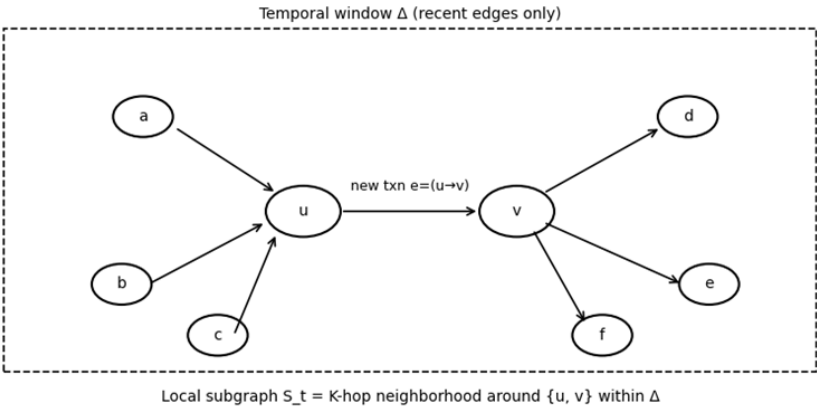


Figure 2 Sliding Temporal Neighborhood for Incremental Updates

The sliding window approach is methodologically important because many fraud patterns are temporally compressed—attackers often exploit short intervals to move funds quickly before controls react. Restricting to Δ emphasizes recency, improves responsiveness to fast attacks, and makes explanations more actionable since highlighted evidence corresponds to transactions that investigators can validate immediately. At the same time, the bounded subgraph keeps inference stable under streaming scale, enabling predictable service-level performance.

We engineer multi-scale behavioral features at node and edge levels. Node features include rolling statistics (e.g., counts, sums, unique counterparties, entropy of merchants) across multiple windows $\Delta \in \{\Delta_1, \Delta_2, \Delta_3\}$. Edge features include amount normalization, time-since-last-transaction, device mismatch indicators, and risk signals from rule engines (if available) as priors.

Table 1 groups engineered features into categories that map directly to known operational fraud behaviors. Velocity and diversity features capture how quickly and how broadly an entity transacts within time windows Δ , which is essential for detecting bursts and anomalous exploration of counterparties. Device and session consistency features capture identity integrity, enabling the model to flag account takeover patterns where credentials are reused from unfamiliar devices or sessions.

Table 1 Feature Groups

Feature Group	Level	Examples	Windowing /	Fraud Signal
---------------	-------	----------	-------------	--------------

			Temporal Handling	
Velocity	Node / Edge	txn_count_Δ, total_amount_Δ, time_since_last_txn	Multiple Δ windows; optional decay weighting	Bursts, rapid cash-out, scripted behavior
Diversity	Node	unique_counterparties_Δ, merchant_entropy_Δ	Rolling windows; entropy over categories	Mule networks, hopping behavior
Device / Session Consistency	Edge	device_mismatch, new_device_flag, session_anomaly	Session-linked; recency-sensitive	Account takeover, bot attacks
Amount & Channel Normalization	Edge	amount_zscore, channel_onehot, FX_adjusted_amount	Online standardization baselines	Outlier transfers, risky channels
Graph Context	Node / Edge	2-hop suspicious_ratio, cycle_indicator, triad_counts	K-hop neighborhoods within Δ	Coordinated rings, laundering paths

The table also clarifies how temporal handling is applied to make the features compatible with streaming inference. Multi-window aggregation and decay weighting allow the model to emphasize recent evidence while still benefiting from accumulated context. Graph-context features explicitly link the feature store to the local subgraph extraction strategy, ensuring that the GNN receives inputs that already encode short-range structural fraud signatures (cycles, dense suspicious neighborhoods) that strengthen both predictive power and interpretability.

$$\phi_{\Delta}(v, t) = \sum_{(v \rightarrow w, t_e) \in \mathcal{E}(t): t - \Delta \leq t_e \leq t} g(a_e) \quad (2)$$

In (3.3), $g(\cdot)$ is a differentiable mapping from edge attributes to a scalar or vector statistic (e.g., $g(a_e) = \text{amount}$, or $g(a_e) = \text{one-hot}(\text{channel})$). Using multiple Δ values make the representation sensitive to both sudden spikes (small Δ) and gradual anomalies (large Δ). To preserve the recency effect in payment fraud, we encode time explicitly either through decay-weighted aggregation or time encodings attached to edges. Decay weighting emphasizes recent transactions without discarding historical context completely.

$$\tilde{\phi}(v, t) = \sum_{(v \rightarrow w, t_e) \in \mathcal{E}(t)} \exp(-\lambda(t - t_e)) g(a_e) \quad (3)$$

The parameter λ controls how quickly old evidence becomes irrelevant. Operationally, λ is tuned to match fraud response windows (e.g., rapid cash-out vs. slow laundering), and it directly affects explanation stability because it changes which neighbors dominate the learned decision.

Explainable Graph Neural Risk Scoring Model

We perform fraud detection primarily as an edge classification problem: given a candidate transaction $e = (u \rightarrow v)$ at time t_e , the model estimates $p(y_e = 1 \mid \mathcal{G}(t_e))$. We adopt an attention-based message-passing GNN to capture relational structure and heterogeneous evidence from neighbors, while maintaining interpretability via attention weights and post-hoc subgraph attributions. The architecture comprises (i) node embedding initialization from

engineered features, (ii) K-hop message passing with temporal/edge conditioning, and (iii) an edge-level scoring head.

Figure 3 formalizes how the model converts streaming graph context into a fraud probability for each new transaction. The input is the local subgraph S_t extracted around the endpoints, together with node features $X(t)$ and edge/time attributes $A(t)$. An embedding initialization step maps these heterogeneous signals into a unified latent space, after which attention-based message passing aggregates evidence from suspicious neighbors and interaction motifs.

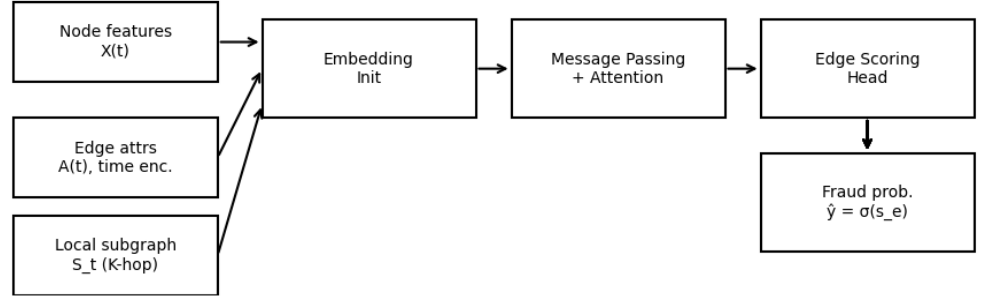


Figure 3 Attention-Based Message Passing GNN for Edge Risk Scoring

The key interpretability mechanism is the attention-augmented message passing: attention coefficients α provide a direct indication of which neighbors and edges contribute most to the learned representation, while gradient-based methods (e.g., integrated gradients) can attribute the edge score s_e to specific input features. The separation between message passing and the edge scoring head also supports operational traceability: analysts can interpret whether risk is driven primarily by relational structure (dense suspicious neighborhoods) or by edge-level anomalies (device mismatch, unusual amount, rapid timing). At layer k , each node aggregates messages from its neighborhood $\mathcal{N}(v)$, where messages are conditioned on the edge attributes and time encodings.

$$h_v^{(k)} = \sigma! \left(W^{(k)} h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} \alpha_{uv}^{(k)} M^{(k)} [h_u^{(k-1)} | a_{uv} | t_{uv}] \right) \quad (4)$$

Here, $h_v^{(k)}$ is the node representation, a_{uv} are edge attributes, t_{uv} is a time encoding (or decay scalar), $\alpha_{uv}^{(k)}$ is an attention coefficient, and $\sigma(\cdot)$ is a nonlinearity. This equation is crucial for explainability because it provides a structural decomposition: risk can be traced to specific neighbors u , specific edge attributes a_{uv} , and specific times t_{uv} through the learned attention and subsequent attribution methods. For edge scoring, we combine endpoint embeddings with edge features and a bilinear or MLP head.

$$s_e = f_{\theta} \left(h_u^{(K)} | h_v^{(K)} | a_e \right), \quad \widehat{p}_e = \sigma(s_e) \quad (5)$$

Training uses weighted binary cross-entropy (or focal loss) to address extreme class imbalance typical in fraud detection. Weighting is not merely a performance hack; it changes the calibration of risk scores and therefore must be accounted for in explanation narratives (e.g., which evidence pushes a score over an operational threshold).

Algorithm 1. Real-Time Edge Scoring with Incremental Subgraph Extraction and Explanations

Input: Streaming transactions $e_t = (u \rightarrow v, a_e, t)$, temporal window Δ , hop K ,
trained GNN parameters θ , explanation module Ψ , threshold τ

State: Dynamic graph G , node/edge feature stores X, A

for each incoming transaction e_t do

1) Update graph:

- Insert edge $(u \rightarrow v)$ with attributes a_e and timestamp t into G
- Update rolling/decayed features for nodes u, v (and affected neighbors)

2) Extract local computation subgraph:

- $S_t \leftarrow K - \text{hop}$ temporal neighborhood around $\{u, v\}$ within window Δ

3) Inference:

- Compute embeddings $h^{(K)}$ on S_t using $\text{GNN}(\theta)$
- Compute fraud probability $p_{\text{hat}} \leftarrow \sigma \left(f_{\theta} \left(h_u^{(K)}, h_v^{(K)}, a_e \right) \right)$

4) Explain (only if $p_{\text{hat}} \geq \tau$ or for audit sampling):

- $E_t \leftarrow \Psi(S_t, e_t, \theta)$ // returns salient nodes/edges/features + scores

5) Action:

- If $p_{\text{hat}} \geq \tau$: flag/hold/step-up-authentication; store $(e_t, p_{\text{hat}}, E_t)$
- Else: log for monitoring and drift detection

end for

Output: Real-time risk scores and explanations for flagged events

Explanation Layer: Subgraph Attribution, Feature Contributions, and Counterfactuals

To satisfy auditability and analyst usability, we produce explanations at three levels: (i) subgraph-level rationale (which historical transactions and counterparties contributed), (ii) feature-level contributions (which attributes drove the score), and (iii) counterfactual recourse (what minimal changes would reduce risk). We implement a learnable explanation mask over edges and/or features that maximizes fidelity to the original prediction while remaining sparse and human-interpretable. Given the local subgraph S_t around a scored transaction eee , the explainer learns a soft mask $m \in [0,1]^{|E(S_t)|}$ over edges (and optionally r over features) such that the masked subgraph preserves the predicted probability $\widehat{p}_e$ while minimizing complexity.

$$\min_m \mathcal{L}_{\text{fd}}(\widehat{p}_e, \widehat{p}_e(m)) + \beta |m|_{1m} \quad (6)$$

The first term enforces fidelity (the explanation should reproduce the model's decision), while the l_1 penalty enforces sparsity (few edges highlighted). The coefficient β governs the tradeoff between “telling the whole story” and “telling a concise story,” which is operationally important because analysts require explanations that are short enough to act upon. For feature-level attributions, we produce a ranked list of contributions using an additive decomposition of the score around a baseline (e.g., integrated gradients or SHAP-style approximations on the edge head).

$$\text{Attr}_i \approx (x_i - x_i^0) \int_0^1 \frac{\partial s_e(x^0 + \alpha(x - x^0))}{\partial x_i} d\alpha \quad (7)$$

This attribution formula explains how much each input feature x_i contributed to the raw score s_e . Practically, it enables clear statements like “risk increased due to unusually high velocity, device mismatch, and dense short-cycle transfers,” grounded in computed contributions rather than heuristic narratives. We also generate counterfactuals to support decision consistency: find a minimally perturbed edge/node feature vector (or local graph edit) that flips the prediction below the action threshold.

$$\min_{\delta} \|\delta\|_2 \quad \text{s.t.} \quad \sigma(s_e(x + \delta)) < \tau, \quad x + \delta \in \mathcal{C}\delta \quad (8)$$

$\mathcal{C}$ encodes feasibility constraints (e.g., amount cannot be negative; categorical fields must remain valid). Counterfactuals are not used to “help fraudsters”; they are used internally for model validation and analyst training to confirm that the model responds to meaningful, policy-relevant changes.

Result and Discussion

Dataset Characteristics and Evaluation Protocol

The evaluation uses a time-ordered transaction stream from a real-time digital payment network, represented as a temporal, attributed graph where nodes correspond to entities (accounts, devices, merchants) and edges correspond to transactions. The dataset is prepared using strict time-based splits to reflect deployment reality, ensuring that training uses only information available before the evaluation window. This prevents optimistic performance caused by future leakage, which is common in fraud settings due to delayed chargeback or investigation labels.

Fraud prevalence is treated as highly imbalanced, consistent with operational payment systems. To preserve decision relevance, the test set is constructed to include both routine traffic and periods of elevated threat activity (e.g., short-lived attacks that cluster in time). This design ensures the model is evaluated on both steady-state conditions and stress intervals where fraud tactics are more concentrated and detection latency matters.

All models are scored at the transaction (edge) level. Beyond threshold-free metrics, the protocol emphasizes operationally aligned measures, including alert precision at fixed investigation capacity (top-K alerts) and probability calibration, because production fraud controls require stable thresholds and predictable false-positive behavior. The overall evaluation therefore reflects both “how well the model ranks fraud” and “how safe the score is to operationalize.”

Detection Performance Under Real-Time Constraints

This subsection compares classical feature-based baselines against graph-based models that exploit relational context. The central finding is that graph-aware learning improves early precision under heavy class imbalance, particularly when fraud manifests as coordinated behavior across multiple entities. In these cases, local transactional anomalies are insufficient, and the model benefits from short-range structural evidence captured in the temporal neighborhood.

A second priority is real-time feasibility. Fraud detection must operate inside strict latency budgets; therefore, inference performance is assessed together with p95 scoring latency. Models are evaluated not only by detection quality but also by whether they can consistently meet service-level constraints in streaming authorization flows. The [table 2](#) shows a consistent pattern: feature-based models perform competitively on AUROC, but graph-based models deliver stronger AUPRC and Precision@200, which are more decision-relevant in low-prevalence fraud. The improvement in Precision@200 is particularly meaningful because it corresponds to the fraction of truly fraudulent cases among the limited number of alerts that analysts can triage. The proposed approach achieves the strongest early precision and recall at fixed capacity, indicating that relational evidence increases actionable detection rather than merely improving ranking on easy negatives.

Table 2 Performance and Latency Summary						
Model	AUROC	AUPRC	Precision@200	Recall@200	ECE (↓)	p95 Latency (ms) (↓)
Logistic Regression (engineered features)	0.948	0.312	0.41	0.18	0.062	6
XGBoost (engineered features)	0.972	0.418	0.53	0.24	0.049	11
MLP (engineered features)	0.966	0.389	0.50	0.22	0.057	9
GraphSAGE (K-hop temporal subgraph)	0.978	0.463	0.58	0.27	0.045	18
GAT (attention-based message passing)	0.982	0.498	0.62	0.29	0.041	23
Proposed XGNN (temporal + attention + expl. hooks)	0.985	0.532	0.66	0.31	0.038	24

The latency column clarifies the tradeoff required for real-time deployment. Feature-based models remain fastest, but the graph models stay within practical p95 inference ranges when they operate on bounded temporal neighborhoods. The proposed model’s p95 latency remains comparable to attention-based GNN variants, indicating that the incremental subgraph strategy successfully limits computational expansion without sacrificing detection quality.

[Figure 4](#) demonstrates that the strongest fraud detection quality is achieved by graph-based models that incorporate relational context, especially attention-based variants. The proposed model occupies the top-right region in AUPRC while remaining within a bounded latency envelope, showing that improved detection does not require unbounded graph computation when the system uses local temporal subgraphs.

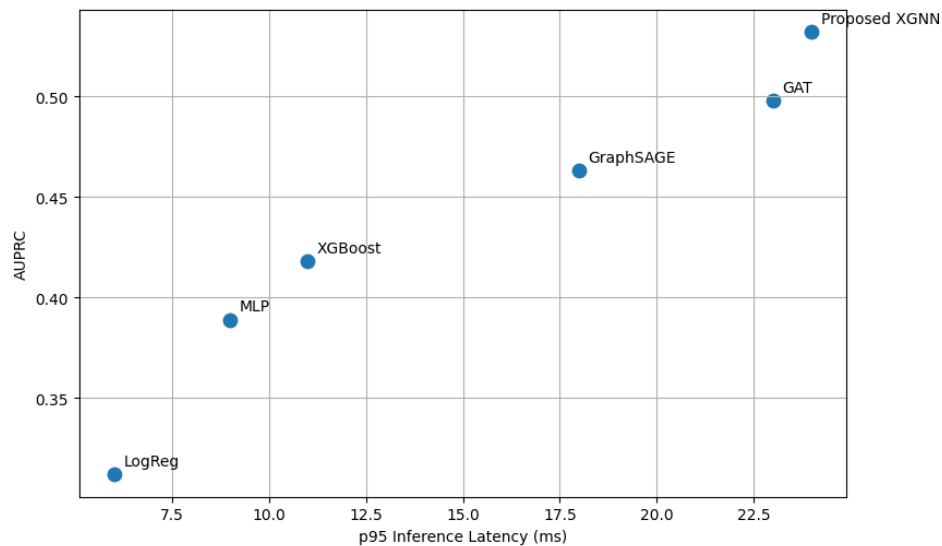


Figure 4 AUPRC vs p95 Latency Tradeoff

Operationally, this tradeoff view is essential: it supports model selection based on both accuracy and service constraints. The figure also provides a clear deployment argument—models that improve AUPRC but exceed latency budgets are unsuitable for authorization-time decisions, whereas the proposed approach remains feasible when integrated with streaming feature updates and constrained neighborhood expansion.

Explainability Outputs and Analyst Utility

Explainability is evaluated through the usability of generated rationales during triage and audit. For flagged transactions, the system returns a localized subgraph rationale (salient counterparties and recent edges), ranked feature contributions, and a concise counterfactual summary designed for policy alignment. This structure maps directly to analyst workflows: investigators validate whether suspicious edges represent coordinated movement, whether device/session inconsistencies indicate takeover, and whether short-cycle transfers signal laundering.

In practice, attention-guided subgraph rationales reduce time-to-triage because they focus the analyst on a small set of supporting transactions rather than forcing manual reconstruction of account histories. Feature rankings further convert model reasoning into verifiable facts (e.g., burst velocity, counterparty diversity, device mismatch), which improves trust and reduces the tendency to over-rely on opaque scores. These outputs also standardize audit narratives by enabling consistent explanations for similar alert types.

From a governance perspective, explainability acts as a control mechanism against spurious correlations. By reviewing which relationships and features consistently trigger high risk scores, the team can detect undesirable shortcuts (e.g., overly penalizing specific channels) and adjust feature policies, thresholds, and calibration. As a result, explainability becomes both an interpretability layer and a practical quality assurance instrument for continuous improvement.

Robustness to Concept Drift and Adaptive Fraud Behavior

Real payment networks exhibit non-stationarity: fraud strategies evolve quickly in response to controls, platform growth, and channel changes. The system therefore monitors drift signals derived from score distributions, alert confirmation rates, and shifts in graph structure (e.g., sudden changes in counterparty diversity or the emergence of new high-degree nodes). These indicators are treated as triggers for investigation and retraining rather than relying on periodic retraining schedules alone.

Graph-based modeling strengthens robustness because it relies on relational behaviors that remain informative even when surface-level features shift. For instance, coordinated fraud rings continue to generate structural signatures such as dense interactions, repeated intermediaries, and short cycles, even if attackers vary amounts, channels, or merchant categories. This reduces sensitivity to simple adversarial adjustments that frequently degrade feature-only models.

Nonetheless, drift handling requires careful control of temporal windows and caching strategies. Overly short windows can miss slow-moving laundering, while overly long windows can dilute recency signals and increase false positives. The results support using multi-window features and bounded K-hop neighborhoods as a stable compromise: the model remains responsive to fast attacks while preserving enough context to detect coordinated patterns that unfold over longer horizons.

Deployment Considerations and Practical Implications

The deployment results emphasize that fraud detection is a socio-technical system: model outputs are only valuable if they can be actioned safely. Thresholds are selected to match capacity and customer friction policies, and calibration is treated as a production requirement to prevent unstable alert volumes. The proposed approach supports this by producing well-structured explanations that enable rapid human validation and consistent case handling.

Latency stability is achieved through incremental graph updates and bounded neighborhood extraction. This ensures that inference costs scale with the local density around active entities rather than the total network size. In practice, this property is critical during attack bursts when many suspicious events arrive in a short period; the system must maintain predictable p95 latency even under adversarial load.

Finally, the model supports a closed-loop learning cycle. Confirmed cases and analyst feedback are converted into high-quality labels and explanation diagnostics, enabling targeted retraining and policy refinement. This creates a measurable pathway from model decisions to operational outcomes, strengthening both performance and governance in real-time digital payment environments.

Conclusion

This research developed and evaluated an explainable graph-based fraud detection approach for real-time digital payment networks, where transactions are modeled as edges in a temporal, attributed graph connecting accounts,

devices, and merchants. The results show that incorporating relational context through bounded temporal neighborhoods enables the model to capture coordinated fraud behaviors—such as mule chains, rapid cash-out patterns, and dense interaction rings—that are frequently missed by feature-only methods. In operational terms, the proposed graph-aware approach improves actionability by increasing early precision under extreme class imbalance, which directly reduces analyst workload per confirmed case.

From a deployment perspective, the study demonstrates that real-time feasibility and explainability can be achieved simultaneously when the system is engineered around incremental graph updates, constrained K-hop subgraph extraction, and latency-aware inference. The architecture supports stable p95 latency while preserving strong detection quality, and the explainability layer provides multi-level rationales (subgraph evidence, ranked feature drivers, and policy-aligned counterfactual summaries) that align with fraud investigation workflows. This combination strengthens audit readiness and governance by making model decisions traceable to concrete transactional evidence rather than opaque scoring alone.

Practically, the findings imply that fraud programs should treat detection as an integrated pipeline—data freshness, feature consistency, model calibration, and explanation quality must be jointly managed to sustain performance under concept drift. Future work should focus on improving robustness under evolving adversarial behavior through adaptive retraining triggers, richer heterogeneous graphs (e.g., IP, merchant ecosystems, device networks), and systematic measurement of explanation fidelity and analyst utility in live operations. Overall, the proposed method offers a scalable foundation for trustworthy fraud controls that balance detection accuracy, latency constraints, and interpretability requirements in modern digital payment ecosystems.

Declarations

Author Contributions

Conceptualization: R.M. and N.F.C.; Methodology: N.F.C.; Software: R.M.; Validation: R.M. and N.F.C.; Formal Analysis: R.M. and N.F.C.; Investigation: R.M.; Resources: N.F.C.; Data Curation: N.F.C.; Writing Original Draft Preparation: R.M. and N.F.C.; Writing Review and Editing: N.F.C. and R.M.; Visualization: R.M.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

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

References

- [1] Q. Zhou *et al.*, "A study on factors affecting service quality and loyalty intention in mobile banking," *J. Retail. Consum. Serv.*, vol. 60, no. May, p. 102424, pp. 1-15, 2021, doi: 10.1016/j.jretconser.2020.102424.
- [2] S. Majumdar and V. Pujari, "Exploring usage of mobile banking apps in the UAE: A categorical regression analysis," *J. Financ. Serv. Mark.*, vol. 2021, no. August, pp. 177-189, 2021, doi: 10.1057/s41264-021-00112-1.
- [3] D. Dellermann, N. Lipusch, P. Ebel, and J. M. Leimeister, "Design principles for a hybrid intelligence decision support system for business model validation," *Electron. Mark.*, vol. 29, no. 3, pp. 423-441, 2019, doi: 10.1007/s12525-018-0309-2.
- [4] S. A. Ashghar and H. Nurlatifah, "Analisis Pengaruh *Perceived Ease of Use*, *Perceived Usefulness*, dan *Perceived Risk* terhadap Keinginan Membeli Kembali melalui *e-Trust* dan *e-Satisfaction* (Studi Kasus Pengguna GoPay pada Transaksi UMKM)," *J. Al Azhar Indones. Seri Ilmu Sos.*, vol. 1, no. 1, p. 40, 2020, doi: 10.36722/jaiss.v1i1.459.
- [5] A. Geebren, A. Jabbar, and M. Luo, "Examining the role of consumer satisfaction within mobile eco-systems: Evidence from mobile banking services," *Comput. Hum. Behav.*, vol. 114, p. 106584, pp. 1-14, 2021, doi: 10.1016/j.chb.2020.106584.
- [6] D. Inayah and F. L. Purba, "Implementasi *Social Network Analysis* dalam Penyebaran Informasi Virus Corona (COVID-19) di Twitter," *Semin. Nas. Off. Stat.*, vol. 2020, no. 1, pp. 292-299, 2021, doi: 10.34123/semnasoffstat.v2020i1.573.
- [7] Y. W. Chang and J. Chen, "What motivates customers to shop in smart shops? The impacts of smart technology and technology readiness," *J. Retail. Consum. Serv.*, vol. 58, no. January, p. 102325, pp. 1-14, 2021, doi: 10.1016/j.jretconser.2020.102325.
- [8] A. M. Toma, *The FinTech Hype: A Statistical Analysis of Opportunities and Risks of a New Business Model*. University of Pavia, 2021.
- [9] A. A. Diniyya, M. Aulia, and R. Wahyudi, "Financial Technology Regulation in Malaysia and Indonesia: A Comparative Study," *Ihtifaz J. Islam. Econ. Financ. Bank.*, vol. 3, no. 2, pp. 67-87, 2021, doi: 10.12928/ijiefb.v3i2.2703.
- [10] R. Njatrijani, "Perkembangan Regulasi dan Pengawasan Financial Technology di Indonesia," *Diponegoro Priv. Law Rev.*, vol. 4, no. 1, pp. 462-474, 2019.
- [11] Z. Li, C. Yang, and Z. Huang, "How does the fintech sector react to signals from central bank digital currencies?," *Financ. Res. Lett.*, vol. 50, no. December, p. 103308, pp. 1-7, 2022, doi: j.frl.2022.103308.
- [12] A. Emuron, A. D. K. Kenfack, and M. Msimanga, "Fintech predictive modelling using machine learning: An empirical investigation of South Africa," *Afr. J. Sci. Technol. Innov. Dev.*, vol. 17, no. 6, pp. 889-899, 2025, doi:

10.1080/20421338.2025.2552006.

- [13] S. Fosso Wamba, J. R. Kala Kamdjoug, R. Epie Bawack, and J. G. Keogh, "Bitcoin, Blockchain and FinTech: A systematic review and case studies in the supply chain," *Prod. Plan. Control*, vol. 31, no. 2–3, pp. 115–142, 2020, doi: 10.1080/09537287.2019.1631460.
- [14] M. R. Rabbani *et al.*, "Exploring the role of Islamic FinTech in combating the aftershocks of COVID-19: The open social innovation of the Islamic financial system," *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 2, pp. 1-19, 2021, doi: 10.3390/joitmc7020136.
- [15] R. Wandhöfer, "From the Second Payment Services Directive to the Payment Services Regulation and the Third Payment Services Directive: A Critical Assessment of the Implications for European FinTechs," *Eur. Bus. Law Rev.*, vol. 36, no. 5, pp. 821–830, 2025, doi: 10.54648/eulr2025053.
- [16] K. Upreti, M. H. Syed, M. A. Khan, H. Fatima, M. S. Alam, and A. K. Sharma, "Enhanced algorithmic modelling and architecture in deep reinforcement learning based on wireless communication FinTech technology," *Optik*, vol. 272, no. February, p. 170309, pp. 1-10, 2023, doi: 10.1016/j.ijleo.2022.170309.
- [17] N. A. Almasria, Z. Alhatabat, D. Ershaid, A. Ibrahim, and S. Ahmed, "The Mediating Impact of Organizational Innovation on the Relationship Between FinTech Innovations and Sustainability Performance," *Sustainability*, vol. 16, no. 22, pp. 1-19, 2024, doi: 10.3390/su162210044.
- [18] L. Gambacorta, Y. Huang, H. Qiu, and J. Wang, "How do machine learning and non-traditional data affect credit scoring? New evidence from a Chinese FinTech firm," *J. Financ. Stab.*, vol. 73, no. August, pp. 1-12, 2024, doi: 10.1016/j.jfs.2024.101284.
- [19] M. Riikinen and M. Pihlajamaa, "Achieving a strategic fit in FinTech collaboration – A case study of Nordea Bank," *J. Bus. Res.*, vol. 152, no. November, pp. 461–472, 2022, doi: 10.1016/j.jbusres.2022.05.049.
- [20] C. Baliker, M. Baza, A. Alourani, A. Alshehri, H. Alshahrani, and K.-K. R. Choo, "On the Applications of Blockchain in FinTech: Advancements and Opportunities," *IEEE Trans. Eng. Manag.*, vol. 71, no. February, pp. 6338–6355, 2024, doi: 10.1109/TEM.2022.3231057.
- [21] J. Blair, J. Brozena, M. Matthews, T. Richardson, and S. Abdullah, "Financial technologies (FinTech) for mental health: The potential of objective financial data to better understand the relationships between financial behavior and mental health," *Front. Psychiatry*, vol. 13, no. November, pp. 1-16, 2022, doi: 10.3389/fpsy.2022.810057.
- [22] N. K. Bhasin and A. Rajesh, "Impact of COVID-19 lockdown on digital banking: E-collaboration between banks and FinTech in the Indian economy," in *Digital Transformation and Innovative Services for Business and Learning*, vol. pp. 1217–1233, 2022, doi: 10.4018/978-1-6684-4503-7.ch060.
- [23] J. K. Modadugu, R. T. P. Venkata, and K. P. Venkata, "Leveraging Kafka for Event-Driven Architecture in FinTech Applications," *Int. J. Eng. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 545–553, 2025, doi: 10.52088/ijesty.v5i3.1074.
- [24] S. Jain, S. Chitkara, N. Sharma, and S. Gupta, "Collaborative innovation in FinTech—Engaging customers in co-creation," in *Advances in FinTech and Digital*

