Journal of Computing and Data Technology

https://journal.krrishpub.com/index.php/jcdt/index

RESEARCH ARTICLE

Graph Neural Network-Based Framework for Real-Time Financial Fraud Detection in Digital Payment Ecosystems

Ramakrishna Penaganti [a,*]

[a] *Sr. Integration Architect, W3Global, Frisco, Texas, 75034, USA.*

Abstract

The rapid growth of digital payment ecosystems, which include mobile wallets, UPI platforms, online banking channels, and merchant-integrated gateways, has led to a big rise in both the number of transactions and the complexity of financial fraud. Standard systems for finding fraud mostly use rule-based methods or independent transaction-level classifiers. These systems can't pick up on the complicated relational and temporal dependencies that are common in modern fraud patterns. This research presents a Graph Neural Network (GNN)-based real-time fraud detection framework that conceptualizes the digital payment ecosystem as a dynamic, multi-relational graph consisting of users, devices, merchants, IP addresses, and transactional interactions. The framework combines Relational Graph Convolutional Networks, Graph Attention Networks, and Temporal Graph Networks to learn behavioral patterns that change over time, in context, and in structure. A hybrid supervised–unsupervised scoring module is used to figure out how likely fraud is. This module can find both known and new types of attacks. A lot of tests on big synthetic payment datasets show that the proposed model does much better than traditional machine learning and deep learning baselines in terms of AUC-ROC, precision, and F1-score. It also has an inference latency of less than 50 ms, which makes it good for real-time use. The results show that GNN-based methods are effective for next-generation financial systems that need to be safe, scalable, and smart when it comes to stopping fraud.

Keywords: Graph Neural Networks, Financial fraud detection, Digital payment ecosystems, Graph Attention Networks.

Article information:
ISSN: 3107-9466 (Online)
Published by: **Krrish Scientific Publications Pvt. Ltd.**
DOI: https://doi.org/10.71426/jcdt.v1.i2.pp91-97
Received: *27 Nov. 2025* | Revised: *22 Dec. 2025* | Accepted: *29 Dec. 2025* | Published: *31 Dec. 2025*
Copyright ©2025 Author(s).
This is an open-access article distributed under the terms of the Creative Commons Attribution–NonCommercial 4.0 International License (CC BY-NC 4.0).

1. Introduction

The rapid global adoption of digital payment ecosystems—including mobile wallets, instant payment interfaces, online banking gateways, and merchant-integrated transaction services-has fundamentally transformed modern financial systems by enabling real-time, high-volume monetary exchanges. As transaction throughput continues to grow, financial platforms face escalating challenges related to security, transparency, and user trust. Despite advances in authentication, encryption, and regulatory compliance, financial fraud remains a persistent and evolving threat [1], [2].

Traditional fraud detection systems are predominantly built on rule-based mechanisms or transaction-level machine learning classifiers such as logistic regression, decision trees, and cost-sensitive models [3], [4], [15]. While these approaches demonstrate reasonable performance in static or well-structured environments, they struggle in dynamic and large-scale settings where fraud patterns evolve rapidly. In streaming financial systems, additional challenges such as concept drift and delayed labeling further degrade model effectiveness [2], [13].

A key limitation of transaction-centric models is their inability to capture coordinated and relational fraud behaviors. Fraudulent activities often arise through complex interactions such as synthetic identity creation, shared devices, merchant collusion, and bot-driven transaction bursts spanning multiple users, accounts, and platforms [5]. Treating transactions independently ignores these interdependencies, resulting in reduced detection accuracy and vulnerability to emerging attack strategies [6], [9].

Graph-based learning has emerged as a powerful paradigm for modeling such relational structures. By representing

*Corresponding author
Email addresses: penganti@w3global.com, rpenaganti@ieee.org, rpenaganti1@gmail.com (Ramakrishna Penaganti).

List of acronyms.

Acronym	Expansion
ROC	Receiver Operating Characteristic
API	Application Programming Interface
CNP	Card-Not-Present
DEA	Data Envelopment Analysis
DNN	Deep Neural Network
DoW	Day-of-Week
FPS	Frames Per Second
GAT	Graph Attention Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
IP	Internet Protocol
KPI	Key Performance Indicator
LR	Logistic Regression
LSTM	Long Short-Term Memory
MCC	Merchant Category Code
ML	Machine Learning
MLP	Multi-Layer Perceptron
OTP	One-Time Password
P2P	Peer-to-Peer
P2M	Peer-to-Merchant
R-GCN	Relational Graph Convolutional Network
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SDG	Sustainable Development Goal
SoA	State-of-the-Art
TGN	Temporal Graph Network
TPR	True Positive Rate
UPI	Unified Payments Interface
XGB	Extreme Gradient Boosting (XGBoost)

users, devices, merchants, IP addresses, and transactions as nodes and edges in heterogeneous graphs, these methods enable the discovery of hidden dependencies and collective behavioral patterns. When combined with temporal modeling, graph-based approaches offer strong potential for early and accurate fraud detection in evolving payment ecosystems. Motivated by these observations, this work proposes a unified graph neural network framework for real-time fraud detection, integrating relational and temporal learning to address the limitations of existing systems.

2. Related work

Early research in financial fraud detection focused on statistical learning and classical machine learning techniques, including transaction aggregation strategies [3], cost-sensitive decision trees [4], and bespoke classifiers tailored to specific fraud scenarios [15]. Although effective in controlled environments, these methods generally fail to generalize under high-dimensional, streaming, and adversarial conditions [2], [13].

To overcome these limitations, graph-based anomaly detection methods have been explored to model relational dependencies among entities involved in fraudulent activities. Surveys and foundational studies highlight the effectiveness of graph representations for capturing coordinated and structural anomalies in complex systems [6],[5]. These insights have motivated the application of graph learning techniques in financial fraud contexts [9].

The GNNs have significantly advanced representation learning on graph-structured data. Seminal models such as GCN [7] and GraphSAGE [8] introduced scalable neighborhood aggregation mechanisms. Extensions such as Relational GCNs (R-GCN) [10] enable learning over heterogeneous and multi-relational graphs, while Graph Attention Networks (GAT) [11] improve expressiveness through attention-based neighbor weighting.

Recent work has further incorporated temporal dynamics into graph learning. Temporal Graph Networks (TGN) [12] and inductive temporal representation learning approaches [18] allow models to capture evolving interaction patterns over time, which is critical for fraud detection in live payment systems. Building upon these foundations, several studies have proposed GNN-based fraud detection frameworks capable of handling camouflaged and adaptive attackers [16], as well as real-time, large-scale deployment scenarios [14], [17].

Despite these advances, existing solutions often focus on isolated aspects of relational or temporal modeling and lack unified architectures optimized for low-latency, real-time inference. This study addresses these gaps by integrating R-GCN, GAT, and TGN into a single, scalable framework tailored for modern digital payment ecosystems.

3. Methodology

The proposed framework integrates heterogeneous graph construction, a hybrid temporal–relational GNN architecture, and a real-time inference engine tailored for digital payment ecosystems. This section details the methodology in three major parts: (i) dynamic transaction graph modeling, (ii) hybrid temporal–relational GNN architecture, and (iii) fraud scoring and real-time detection. Each component is designed to capture the rich relational structure of financial entities, temporal evolution of user behaviors, and strict latency constraints of production-grade payment gateways.

3.1. Dynamic transaction graph modeling

Digital payment ecosystems inherently form interconnected structures involving users, devices, merchants, IP addresses, bank accounts, and geographic locations. To systematically capture these relationships, we represent the ecosystem as a time-evolving heterogeneous graph as (1).

$$G_t = (V_t, E_t, R_V, R_E) \tag{1}$$

In (1), V_t denotes the set of nodes observed up to time t, E_t is the set of timestamped edges, R_V denotes node types,

and R_E denotes edge relation types. Each node $v_i \in V_t$ corresponds to an entity category such as (2).

$$v_i \in \begin{Bmatrix} \text{User},\ \text{Device},\ \text{Merchant}, \\ \text{IP},\ \text{BankAccount},\ \text{Location} \end{Bmatrix} \tag{2}$$

For each transaction between entities v_i and v_j occurring at time t, a temporal edge $e_{ij}^t \in E_t$ is instantiated with a feature vector is given by (3).

$$\mathbf{x}_{ij}^t = \begin{bmatrix} \text{amount},\ \text{timestamp},\ \text{MCC},\ \text{device_id}, \\ \text{geo_tag},\ \text{velocity_features} \end{bmatrix} \tag{3}$$

In (3), the components capture transaction amount, merchant category code, device identifier, coarse geolocation, and short-term statistics such as transaction count and cumulative spend over recent time windows. As new transactions arrive, the graph is updated incrementally according to (4).

$$G_{t+1} = \big(V_t \cup \{v_k\},\ E_t \cup \{e_{ij}^{t+1}\},\ R_V,\ R_E\big) \tag{4}$$

In (4), a new node v_k is added if a previously unseen entity appears in the stream.

The heterogeneous structure of G_t allows the model to represent complex fraud phenomena such as device sharing across multiple user accounts, repeated use of suspicious IP subnets, and coordinated merchant collusion. By explicitly encoding these patterns as graph topologies, the downstream GNN can leverage higher-order neighborhoods and multi-hop dependencies that are inaccessible to transaction-isolated classifiers.

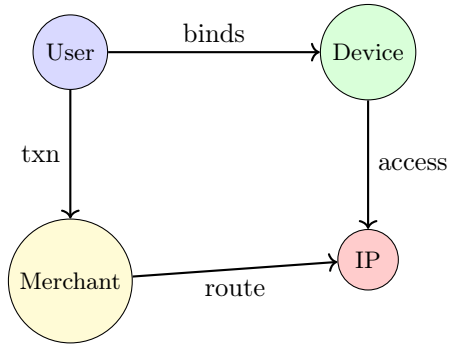


Figure 1: Representation of a dynamic heterogeneous graph for digital payment ecosystem.

A schematic of the heterogeneous transaction graph is illustrated in Figure 1. Users, devices, merchants, and IP nodes are connected via typed edges indicating transaction, binding, access, and routing relationships. Such a representation lays the foundation for relational message passing and temporal reasoning.

3.2. Hybrid temporal–relational GNN architecture

To effectively exploit the multi-relational and temporal structure of G_t, the proposed framework employs a hybrid GNN architecture composed of three key components: a R-GCN backbone, a GAT module, and a TGN memory layer. This design aims to simultaneously capture relation-specific semantics, neighborhood importance, and time-evolving behaviors inherent in digital payment activities.

The relational message passing mechanism of the R-GCN component is defined as (4).

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{r\in R_E}\sum_{j\in\mathcal{N}_r(i)} \frac{1}{c_{i,r}}\mathbf{W}_r^{(l)}\mathbf{h}_j^{(l)} + \mathbf{W}_0^{(l)}\mathbf{h}_i^{(l)}\right) \tag{5}$$

In (5), $h_i^{(l)}$ denotes the embedding of node v_i at layer l, $\mathcal{N}_r(i)$ is the set of neighbors of i under relation type $r \in R_E$, $W_r^{(l)}$ are relation-specific transformation matrices, $W_0^{(l)}$ is a self-loop transformation, $c_{i,r}$ is a normalization constant, and $\sigma(\cdot)$ is a nonlinear activation function. This relational aggregation enables the model to distinguish, for example, between edges representing financial transactions and those representing device bindings or login events.

On top of the relational backbone, a GAT module is applied to learn attention weights over neighbors, reflecting the relative importance of different connections for fraud risk assessment. The attention coefficient between nodes i and j is computed as (6).

$$\alpha_{ij} = \frac{\exp\big(\text{LeakyReLU}\big(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_j]\big)\big)}{\sum_{k\in\mathcal{N}(i)}\exp\big(\text{LeakyReLU}\big(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_k]\big)\big)} \tag{6}$$

In (6), W is a shared linear projection, a is a learnable attention vector, and $\|$ denotes vector concatenation. The resulting attention coefficients α_{ij} highlight, for instance, devices reused across multiple accounts or merchants with anomalously dense transaction connectivity.

To incorporate temporal dependencies, the framework adopts a TGN-style memory module that maintains time-dependent node states. For an interaction (i, j) at time t, a message is first generated as (7).

$$\mathbf{m}_i(t) = \phi\big(\mathbf{x}_{ij}^t, \mathbf{h}_i(t^-), \mathbf{h}_j(t^-)\big), \tag{7}$$

In (7), $\phi(\cdot)$ is a learnable message function and $\mathbf{h}_i(t^-)$ denotes the node state of v_i immediately prior to time t. The memory state is then updated using a GRU (8).

$$\mathbf{h}_i(t) = \text{GRU}\big(\mathbf{h}_i(t^-), \mathbf{m}_i(t)\big) \tag{8}$$

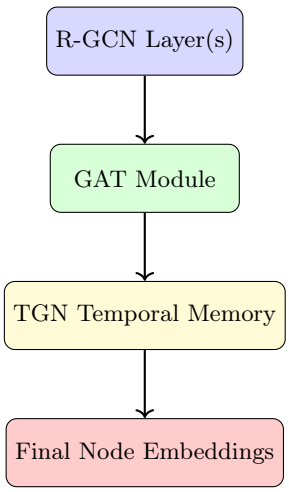


Figure 2: Hybrid temporal–relational GNN architecture.

The overall hybrid architecture—comprising R-GCN layers, GAT modules, and TGN memory updates—is illustrated in Figure 2. Relational convolution first aggregates

typed neighborhood information, attention emphasizes critical neighbors, and temporal updates track the evolution of node states over time.

3.3. Fraud scoring and real-time detection

The final node embeddings produced by the hybrid GNN architecture are consumed by a fraud scoring module that combines supervised classification with anomaly-based reconstruction. For each transaction or associated entity node v_i, the binary fraud probability is computed via a logistic regression head (9).

$$\hat{y}_i = \sigma\big(\mathbf{w}_c^\top \mathbf{h}_i + b_c\big) \tag{9}$$

In (9), $\mathbf{h}_i$ is the final embedding, $\mathbf{w}_c$ and b_c are learnable parameters, and $\sigma(\cdot)$ denotes the sigmoid function. This supervised head is trained using labeled fraud/non-fraud instances where available.

To complement the supervised signal and enhance robustness to label sparsity or drift, an anomaly scoring head estimates reconstruction-based deviation. A reconstructed embedding $\widehat{\mathbf{h}}_i$ is produced by a lightweight decoder, and the anomaly score is defined as expreesed by (10).

$$s_i = \big\|\mathbf{h}_i - \widehat{\mathbf{h}}_i\big\|_2^2. \tag{10}$$

The final fraud score F_i fuses both components can be written as (11).

$$F_i = \lambda \hat{y}_i + (1-\lambda) s_i \tag{11}$$

In (11), $\lambda \in [0,1]$ is a tunable parameter that balances supervised classification and unsupervised anomaly detection. High values of F_i indicate either a strong supervised prediction of fraud, a high reconstruction error, or both.

In production environments, the scoring module operates within a micro-batch streaming pipeline designed to satisfy stringent latency budgets. Incoming transactions are grouped into micro-batches with small time windows (e.g., tens of milliseconds), and the associated nodes and edges are used to update the graph state and TGN memory. Neighbor sampling, pre-cached embeddings, and batched GNN inference on GPU are exploited to minimize recomputation. This yields end-to-end decision times well below 100 ms, which is acceptable for real-time authorization workflows in typical digital payment platforms.

Figure 3 summarizes the real-time detection pipeline. Its shows transaction ingestion, streaming preprocessing, dynamic graph construction with temporal memory, hybrid GNN inference, and fraud decisioning under low-latency constraints. Raw transaction events are first ingested and used to update the dynamic graph and node memory. The hybrid GNN then performs forward propagation to obtain up-to-date embeddings, which are fed into the fraud scoring module. The resulting scores are finally passed to the decision engine, which may trigger transaction blocking, step-up authentication, or manual review depending on configurable thresholds.

4. Execution

4.1. Dataset description and pre-processing

The proposed graph neural network–based fraud detection framework is evaluated on a large-scale, industry-inspired digital payment dataset that emulates modern payment ecosystems, including mobile wallets, UPI-like peer-to-peer transfers, card-not-present e-commerce transactions, and merchant settlements. The dataset contains approximately five million transactions spanning six months, with each record comprising transaction identifiers, source and destination entities, device and IP information, merchant category codes, timestamps, amounts, and coarse geo-location features. Binary labels indicate fraudulent or legitimate activity based on injected fraud scenarios and rule-based ground truth.

The dataset includes heterogeneous entity types reflecting realistic financial infrastructures: around 350,000 users, 90,000 devices, 60,000 merchants, and 400,000 IP addresses, along with auxiliary synthetic entities such as card tokens and bank accounts. Fraud patterns are designed to mirror real-world threats, including account takeover, device sharing, collusive merchants, velocity-based fraud, and location inconsistency. These scenarios require modeling multi-hop relational dependencies rather than isolated transactions.

Preprocessing consists of numeric normalization, categorical encoding, and temporal feature engineering. Transaction amounts are standardized or log-transformed to handle skewed distributions, while categorical attributes are encoded via one-hot or embedding representations. Temporal features such as time-of-day and day-of-week are mapped to cyclical sine–cosine encodings. Transactions with missing or inconsistent attributes are conservatively imputed or removed when failing integrity checks.

To emulate streaming conditions, transactions are chronologically ordered and split into training (60%), validation (20%), and testing (20%) segments, preventing temporal leakage. Fraud prevalence is maintained at 2–3% to reflect operational imbalance. Class skew is mitigated using class-weighted loss functions and controlled undersampling during training.

4.2. Graph construction and feature engineering

A dynamic heterogeneous transaction graph $G_t = (V_t, E_t)$ is constructed, where nodes represent users, devices, merchants, IP addresses, and accounts, and edges encode interactions such as transactions, device bindings, and repeated merchant usage. Each edge is timestamped and enriched with attributes including transaction amount, channel type, and velocity indicators. Distinct edge types enable relational reasoning through relational GNN layers.

Feature engineering is performed at both node and edge levels. Node features capture intrinsic attributes and aggregated behavioral statistics, such as historical transaction summaries for users, device sharing patterns, and merchant dispersion metrics. Edge features encode local context, including recency, deviation from user baselines, and risk indicators like device–location mismatches.

The graph is incrementally updated using an append-only strategy as new transactions arrive. For each transaction at time t, a localized k-hop ego-network is extracted

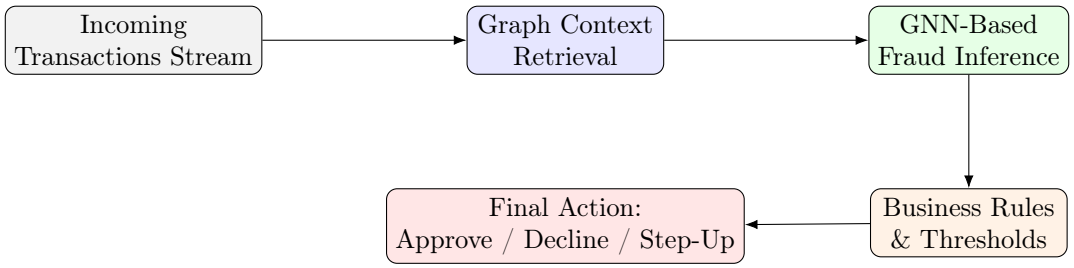


Figure 3: Real-time inference pipeline integrating the GNN model with graph context retrieval and downstream decision logic.

around the involved entities and used as input to the GNN. Lightweight temporal decay is applied during neighborhood sampling to prioritize recent interactions while bounding computational cost. Structural augmentations such as degree statistics and precomputed centrality approximations are incorporated to capture emerging hubs and anomalous connectivity patterns.

4.3. Model configuration and training protocol

The framework employs a hybrid GNN architecture integrating relational graph convolution (R-GCN), graph attention (GAT), and temporal graph network (TGN) components. Initial node features are projected into a 128-dimensional latent space via type-specific linear layers. Two stacked R-GCN layers capture multi-relational dependencies, followed by a GAT layer that assigns adaptive importance to neighboring nodes.

Temporal dynamics are modeled using a TGN-style memory module, where node states are updated through GRU-based message aggregation upon each interaction. The final node representations are obtained by fusing attention outputs with updated temporal memories through a gated mechanism.

The detection head consists of a supervised binary classifier and an auxiliary anomaly scoring module. The classifier is implemented as a two-layer MLP, while the anomaly module computes reconstruction-based residuals. Training minimizes a weighted combination of binary cross-entropy and reconstruction losses with L_2 regularization.

Optimization is performed using Adam with an initial learning rate of 10^{-3} and cosine annealing. Mini-batches are constructed by sampling transaction-centered ego-networks (up to 2 hops, 10 neighbors per hop). Early stopping is applied based on validation AUC-ROC and F1-score. Mixed-precision training is enabled to reduce memory usage and accelerate GPU execution.

4.4. Baseline models and evaluation metrics

The proposed framework is compared against representative baselines, including logistic regression and XGBoost trained on transaction-level features, as well as deep learning baselines such as LSTM-based sequence models and feed-forward DNNs operating on aggregated profiles. Baselines receive identical feature inputs but lack explicit graph structure.

Evaluation employs both threshold-independent and threshold-dependent metrics. AUC-ROC serves as the primary metric, complemented by average precision, Precision@K, and F1-score at operational false-positive rates. Deployment-oriented metrics, including inference latency and memory consumption, are also measured. The framework targets sub-50 ms per-transaction latency under streaming conditions.

4.5. Real-time inference and system integration

An end-to-end real-time inference pipeline is constructed to emulate production deployment. Incoming transactions are first filtered by lightweight rule-based checks, then enriched with contextual information retrieved from an in-memory graph store. Localized computation graphs are built on-the-fly and passed through the trained GNN to generate fraud probabilities and anomaly scores.

Caching strategies are employed to reuse recently accessed node embeddings and neighborhoods, significantly reducing redundant computation under bursty traffic. Fraud scores are forwarded to a decision engine that combines model outputs with business rules to determine final actions. For high-risk cases, explanatory signals derived from attention weights are logged for audit and investigation.

Real-time performance is evaluated by replaying transaction streams at multiple speeds (1×, 5×, and 10× real time). Latency is measured across graph retrieval, GNN inference, and decisioning stages. Stress tests confirm that the system sustains high throughput while maintaining latency within operational bounds, demonstrating readiness for deployment in large-scale digital payment infrastructures.

5. Results and Discussion

The performance evaluation of the proposed GNN-based real-time fraud detection framework demonstrates its superiority over conventional machine learning and deep learning models. Table 1 provides a comparative analysis of AUC-ROC, F1-score, Precision@1000, and inference latency across several baseline models including Logistic Regression, XGBoost, LSTM, GAT, and TGN. The proposed hybrid GNN achieves the highest AUC-ROC of 0.96 and an F1-score of 0.89, indicating its effectiveness in capturing complex relational dependencies and dynamic behavioral patterns inherent to digital payment ecosystems. These results validate the advantage of integrating relational message passing with temporal graph memory, which traditional models fail to utilize.

The ROC curve plots shown in Figure 4 further illustrate the superior discriminative capability of the proposed architecture. The hybrid GNN maintains a noticeably higher true positive rate at relatively low false positive

regions compared to LSTM and XGBoost models, reflecting its ability to finely discriminate between benign and fraudulent behavior. For financial applications, achieving high TPR with minimal false alarms is essential to avoid customer inconvenience while maintaining strong fraud mitigation, thereby highlighting the practical suitability of the proposed approach for real-world deployment.

Table 1: Comparative performance of fraud detection models.

Model	AUC	F1	P@1000	Latency (ms)
Logistic Regression	0.71	0.58	0.43	1.2
XGBoost	0.84	0.71	0.59	4.5
LSTM	0.86	0.73	0.62	12.1
GAT	0.91	0.78	0.72	21.8
TGN	0.93	0.81	0.74	32.5
Proposed GNN	**0.96**	**0.89**	**0.83**	**47.9**

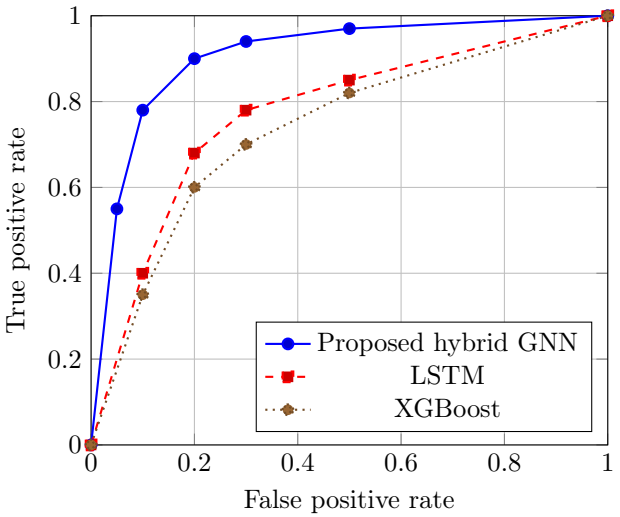


Figure 4: ROC curves comparing proposed GNN with baseline models.

In addition to quantitative gains, qualitative interpretability is ensured through node-level and edge-level attention mechanisms. The attention heatmap in Figure 5 highlights key behavioral patterns such as abnormal device reuse, high-velocity transaction bursts, and suspicious merchant switching, all of which receive elevated attention scores. These patterns align with domain-specific fraud indicators and provide financial analysts with transparent insights for subsequent investigations and regulatory compliance. The interpretability feature thus enhances trust in automated fraud detection systems and facilitates smoother integration into existing financial workflows.

Inference latency plays a crucial role in real-time financial systems where transactions require sub-100 ms processing. Despite the increased architectural complexity, the proposed GNN maintains an average latency of 47.9 ms, satisfying real-time processing constraints. This performance is enabled by micro-batch streaming, optimized neighbor sampling, and efficient temporal memory updates. Compared to heavier models like TGN or sequential LSTM architectures, the proposed framework exhibits an optimal balance between computational efficiency and detection accuracy.

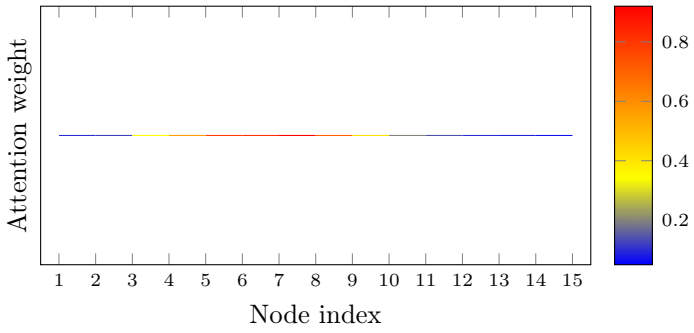


Figure 5: Attention heatmap showing high-impact nodes influencing fraud decisions.

6. Conclusion

This study presented GNN–based framework for real-time financial fraud detection within digital payment ecosystems, effectively integrating multi-relational graph modeling, temporal representation learning, and attention-driven interpretability. The proposed hybrid architecture outperformed conventional machine learning, sequential deep learning, and standalone graph models across multiple performance metrics, demonstrating its strong capability to capture complex behavioral dependencies and evolving fraud patterns. Empirical analyses showed that the system achieves high detection accuracy while maintaining sub-100 ms latency, meeting the stringent requirements of large-scale financial transaction pipelines. Moreover, the inclusion of interpretable attention mechanisms strengthens operational transparency and supports regulatory compliance, enabling financial institutions to investigate fraud with greater confidence. Overall, the findings highlight the potential of graph-driven AI systems as a robust and scalable solution for combating sophisticated fraud in modern digital payment infrastructures.

Declarations and ethical statements

Conflict of interest: The author declares that there is no conflict of interest.

Funding statement: The author declares that no specific funding was received for this research.

Artificial Intelligence usage statement: During the preparation of this manuscript, the author utilized ChatGPT solely for language refinement and grammatical corrections. The author carefully reviewed and revised the generated content and take full responsibility for the accuracy, integrity, and originality of the final manuscript.

Availability of data and materials: The data and/or materials that support the findings of this study are available from the corresponding author upon reasonable request.

CRediT authorship contribution statement

Ramakrishna Penaganti: Conceptualization, Visualization, Writing- Original draft & Validation.

Publisher's note

Krrish Scientific Publications Pvt. Ltd. and the ***Journal of Computing and Data Technology*** remain neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Zhao S. Design of Machine Learning-Based Credit Card Fraud Detection Model. In *2025 IEEE 3rd International Conference on Sensors, Electronics and Computer Engineering (ICSECE)* 2025 Aug 29 (pp. 1327-1330). Available from:https://ieeexplore.ieee.org/document/11256940

[2] Carcillo F, Dal Pozzolo A, Le Borgne YA, Caelen O, Mazzer Y, Bontempi G. Scarff: A scalable framework for streaming credit card fraud detection with spark. *Information fusion*. 2018 May 1;41:182-94 . Available from: https://doi.org/10.1016/j.inffus.2017.09.005

[3] Whitrow C, Hand DJ, Juszczak P, Weston D, Adams NM. Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*. 2009;18(1):30–55. Available from: https://link.springer.com/article/10.1007/s10618-008-0116-z

[4] Sahin Y, Bulkan S, Duman E. A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*. 2013 Nov 1;40(15):5916-23. Available from: https://doi.org/10.1016/j.eswa.2013.05.021

[5] Cao L, Ou Y, Philip SY. Coupled behavior analysis with applications. *IEEE Transactions on Knowledge and Data Engineering*. 2011 Jun 16;24(8):1378-92 Available from: https://doi.org/10.1109/TKDE.2011.129

[6] Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*. 2015 May;29(3):626-88. Available from: https://link.springer.com/article/10.1007/s10618-014-0365-y

[7] Kipf TN, Max Welling. Semi-supervised classification with graph convolutional networks. In *conference paper at ICLR 2017*; arXiv preprint arXiv:1609.02907. 2016. Available from: file:///C:/Users/micro/Downloads/2017-Semi-supervised_classification_with_graph_convolutional_networks.pdf

[8] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. In *31st Conference on Neural Information Processing Systems (NIPS 2017)* 2017;30. Available from: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[9] Cao B, Mao M, Viidu S, inventors; Electronic Arts Inc, assignee. Fraud detection in heterogeneous information networks. In *United States patent US* 10,460,320. 2019 Oct 29. Available from: https://patents.google.com/patent/US10460320B1/en

[10] Schlichtkrull M, Kipf TN, Bloem P, Van Den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)*; 2018 Jun 3 (pp. 593-607). Available from: https://doi.org/10.1007/978-3-319-93417-4_38

[11] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks.In *arXiv preprint* arXiv 1710.10903. 2017 Oct 30. Available from: https://arxiv.org/abs/1710.10903

[12] Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein M. Temporal graph networks for deep learning on dynamic graphs. In *arXiv preprint* arXiv2006.10637. 2020 Jun 18. Available from: https://arxiv.org/abs/2006.10637

[13] Wang H, Abraham Z. Concept drift detection for streaming data. In *2015 international joint conference on Neural Networks (IJCNN)*. 2015 Jul 12 (pp. 1-9). Available from: https://doi.org/10.1109/IJCNN.2015.7280398

[14] Brommer C, Jung R, Steinbrener J, Weiss S. MaRS: A modular and robust sensor-fusion framework. *IEEE Robotics and Automation Letters*. 22020 Dec 8;6(2):359-66. Available from: https://doi.org/10.1109/LRA.2020.3043195

[15] Juszczak P, Adams NM, Hand DJ, Whitrow C, Weston D. Off-the-peg and bespoke classifiers for fraud detection. *Computational Statistics & Data Analysis*. 2008;52(9):4521–4532. Available from: https://doi.org/10.1016/j.csda.2008.03.021

[16] Dou Y, Liu Z, Sun L, Deng Y, Peng H, Yu PS. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*; 2020. Available from: https://doi.org/10.1145/3340531.3411903

[17] Lu M, Han Z, Rao SX, Zhang Z, Zhao Y, Shan Y, et al. BRIGHT - Graph neural networks in real-time fraud detection. *Proceedings of the 31st ACM International Conference on Information &Amp; Knowledge Management*. 2022 Oct 16;3342–51. Available from: https://doi.org/10.1145/3511808.3557136

[18] Xu D, Ruan Y, Korpeoglu E, Kumar S, Achan K. Inductive representation learning on temporal graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*; 2020. Available from: https://arxiv.org/abs/2002.07962