Journal of Computing and Data Technology

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

RESEARCH ARTICLE

Multisource Fraud Intelligence Using Graph Neural Networks for Merchant, Account, and Transaction Risk Assessment

Yeswanth Dintakurthy [a,*]

[a] *Department of Computer Science, University of North Texas, 1155 Union Circle, Texas 76203-5017 Denton, USA.*

Abstract

Fraud intelligence in payment platforms is inherently relational: a transaction belongs to an account, reaches a merchant, and inherits risk from prior behavior on both sides. Yet graph models can underperform strong tabular learners when fraud labels are scarce or when new accounts appear after training. This paper evaluates that trade-off with a leakage-controlled heterogeneous graph experiment on a 10,000-transaction public BankSim extract containing 6,021 customer accounts, 899 merchants, 16 categories, and 78 frauds. Transactions are ordered by simulation step; steps 1–110 train the models, steps 111–145 select thresholds and graph-fusion weight, and steps 146–180 form a locked future test with 2,000 transactions and 16 frauds. Causal account, merchant, and pair histories are computed before each transaction. A two-layer inductive GraphSAGE encoder is trained on the account–merchant graph and combined with a causal XGBoost score. Standalone GraphSAGE is unstable and weak (test 0.0746), whereas transaction-only XGBoost reaches 0.4251. Validation-selected graph fusion improves to 0.4500 and F1 to 0.4103 while detecting 8/16 frauds with 15 false alerts at the frozen operating threshold. Aggregated risk reaches 0.4502 for accounts and 0.4654 for merchants. The results support graph representations as a complementary relational signal, not a guaranteed replacement for transaction models.

Keywords: Financial fraud detection, Graph Neural Networks, GraphSAGE, Merchant risk, Account risk.

Article information:
Published by: **Krrish Scientific Publications Pvt. Ltd.**
DOI: https://doi.org/10.71426/jcdt.v1.si1.pp36-42
Received: *26 Sep. 2025* | Revised: *02 Nov. 2025* | Accepted: *13 Nov. 2025* | Published: *20 Nov. 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

In a database, digital-payment fraud rarely appears as a single record. A transaction originates from an account, is transmitted to a merchant, is given a commercial category, and is incorporated into a history of prior interactions. Hence, fraud detection is at the crossroad of rare-event classification and relational inference. Surveys on financial fraud detection have long pointed out the challenges associated with severe class imbalance, changing attack behavior, and the difficulty in obtaining representative labeled data [1]. More recent work based on BankSim also relies on synthetic payment traces, since it is difficult to release realistic account-level financial data because of privacy and regulatory constraints [2]. The resulting benchmark problem is useful precisely because it highlights a persistent tension: a detector should exploit entity context without conflating synthetic validation for evidence of real-bank deployment.

One of the main responses to that tension has been learning graphs. As illustrated in the systematic review by Motie and Raahemi, graph neural networks (GNNs) are increasingly employed since they can propagate information across the structural relationships that tabular models discard [3]. Cheng et al. also categorize financial-fraud GNN studies based on graph construction, aggregation, heterogeneity, imbalance, and explainability [4]. The attraction is direct in transaction systems: two locally-ordinary transactions can become suspicious when they share a merchant, device, account cluster, or multi-hop neighborhood. However, building graphs may also lead to leakage. A static graph created over the entire window of observation, and thus allowing future relations to affect past scores, is incompatible with prospective fraud screening [17].

This temporal boundary has been made explicit in industrial research [14]. BRIGHT isolates the historical and real-time graph components so that the future payment topology cannot flow back in message passing [5]. Dynamic GNNs work similarly to model evolving collaborative networks rather than assuming that a single graph snapshot

*Corresponding author
Email addresses: yeswanthdintakurthy@my.unt.edu, yeswanthd.96@gmail.com, (Yeswanth Dintakurthy).

is stationary [6]. To distinguish behavior patterns from camouflage, context-aware aggregation has been proposed [7]. Relation-aware heterogeneous GNNs aim to preserve diverse semantic interactions between node types [8]. Recent time heterogeneous systems integrate user-merchant structure, time-aware representations, and imbalance handling in a common detector [9]. Large-scale graph-powered fraud systems further motivate the need to evaluate graph algorithms on classification metrics, but also on scalability and operational architecture [10].

These advances justify graph modeling, but do not prove that a GNN must beat a well-tuned transaction model. Three practical conditions can reverse the expected advantage. First, fraud labels may be too rare to train stable neighborhood aggregation. Second, a large fraction of future accounts may be unseen during training, reducing the utility of memorized relational state. Third, merchant and account history may already be summarized effectively by causal tabular aggregates, leaving a GNN with only incremental information. A credible experiment should therefore report a strong transaction-only baseline, a causal multisource baseline, a standalone GNN, and a fusion model rather than comparing a proposed graph architecture only with weak classifiers.

The empirical benchmark is a traceable 10,000-row BankSim extract, not the complete 594,643-row BankSim release. BankSim is a simulator calibrated from aggregated Spanish-bank payment behavior; consequently, the experiment is a controlled methodological study and not a claim about real customer or merchant risk. The contribution is fivefold. First, account, merchant, pair, and transaction attributes are separated and updated causally. Second, a chronological train/validation/test protocol prevents future topology from entering training. Third, an inductive GraphSAGE edge classifier is evaluated directly, including three-seed instability and unseen-account performance. Fourth, a validation-selected fusion combines graph and causal tabular evidence without using locked-test labels. Fifth, transaction risk is aggregated to merchant and account risk, allowing the paper to test whether relational scoring adds value at all three operational levels.

2. Related work and research gap

We use GraphSAGE in the current work as the benchmark is focused on inductive inference. *et al.* Hamilton introduced GraphSAGE as a neighborhood-aggregation framework that learns a function to generate representations, rather than a separate embedding for each training node [11]. This distinction is important in the case of no future customer accounts in the training graph. Attention based graph models generalize this idea by learning differential neighbor importance [12], which can be helpful in fraud applications where only a minority of adjacent entities are informative. The current experiment intentionally uses a simpler mean-aggregation GraphSAGE to attribute any gain to graph context and not a large attention architecture tuned on a very small fraud sample.

In fraud-specific GNN research, heterogeneity and time are paid more attention. BRIGHT is designed for online payment scoring, with strict historical information flow, in a two-stage topology [5]. CORE-DGNN (CORE) is designed to detect fast-evolving collaborative fraud structures. CAFD encodes the temporal context and adapts aggregation to fraud camouflage [7]. Efficient relation-aware heterogeneous GNNs explicitly model the differences among financial relations [8] and THG-OAFN combines temporal slicing, user–merchant heterogeneity and attention-based detection [9]. Such models demonstrate the representational power of graph learning, but their complexity may mask the utility of graph structure in a small highly imbalanced benchmark.

Traditional boosted trees still provide a strong baseline. XGBoost is a combination of additive decision tree models with regularized optimization and works well on nonlinear tabular interactions [13] For a payment transaction, account frequency, merchant frequency, amount deviation, recency, and category can all be represented as ordinary columns if they are constructed causally. This motivates the Causal Multi-XGB comparator below: if such features already summarize the useful graph information, a GNN should not receive credit merely for using entity identifiers.

Evaluation under extreme imbalance requires the same care. Davis and Goadrich showed that ROC and precision–recall spaces emphasize different characteristics of binary ranking [15]; Saito and Rehmsmeier demonstrated why precision–recall analysis is particularly informative when the positive class is rare [16]. Accordingly, is the primary ranking metric in this work. is still reported, but a high is not interpreted as strong fraud concentration when only 0.8% of locked-test transactions are fraudulent. Thresholded precision, recall, F_1, F_2, and analyst-budget retrieval are reported separately.

The research gap is therefore narrower than "GNNs detect fraud." The question is whether a graph representation provides incremental information after causal account and merchant histories are already available, and whether that increment persists for unseen accounts and at entity-level risk aggregation. The proposed graph-fusion model answers this question with a negative control: standalone GraphSAGE is retained even when it performs poorly. A fusion improvement is considered credible only if the graph score is selected on validation data and the locked test remains untouched.

3. Methodology

3.1. Dataset, construct boundary, and chronology

The packaged dataset is a public 10,000-row extract of BankSim with columns `step`, `customer_id`, age, gender, `merchant_id`, category, amount, and fraud.[1] Its SHA-256 digest is recorded in the project. The extract contains 6,021 customer identifiers, 899 merchants, 16 commercial categories, and 78 fraud transactions (0.78%). The canonical BankSim benchmark is much larger; this study does not present the 10,000-row extract as the complete simulator output. Nor are `customer_id` and `merchant_id` real persons or businesses. In the paper, "account" denotes the simulated customer account represented by `customer_id`.

[1]Exact extract: public CSV; simulator description: BankSim paper.

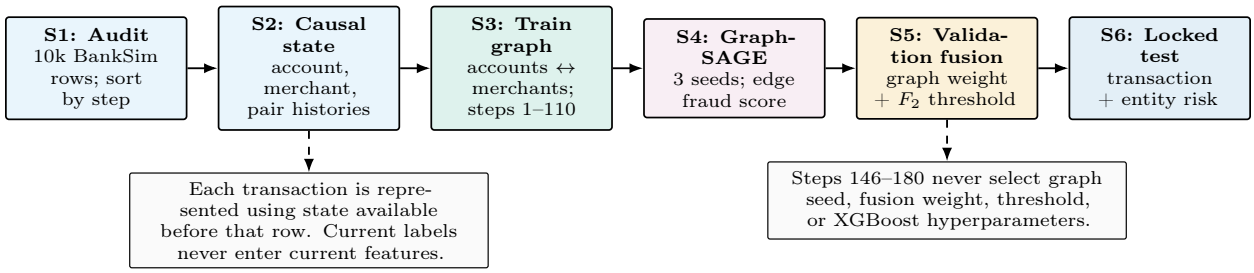

Figure 1: Leakage-controlled multisource fraud workflow.

Table 1: Dataset and chronological evaluation protocol.

Item	Value
Transactions / frauds	10,000 / 78
Fraud prevalence	0.780%
Accounts / merchants	6,021 / 899
Categories	16
Train, steps 1–110	6,039 / 47 frauds
Validation, steps 111–145	1,961 / 15 frauds
Locked test, steps 146–180	2,000 / 16 frauds
Test transactions with seen account	46.90%
Test transactions with seen merchant	99.95%
GraphSAGE seeds	11, 42, 73

Rows are sorted by simulation `step` and original row order. Steps 1–110 form the training period, steps 111–145 form validation, and steps 146–180 are locked for test. Table 1 shows that the split preserves a severe minority class. The test period contains 16 frauds among 2,000 transactions. Only 46.9% of test transactions belong to accounts already seen in training, while 99.95% use merchants seen during training. This asymmetry is valuable: merchant context is mostly recurrent, whereas account inference is frequently inductive.

3.2. Causal transaction, account, and merchant features

For transaction i at step t_i, state variables are calculated before updating the history with transaction i. Let $C_a(t^-)$ and $C_m(t^-)$ be counts of prior transactions for account a and merchant m. Let $\bar{A}_a(t^-)$ and $\bar{A}_m(t^-)$ denote prior mean amounts. Representative causal features are

$$\mathbf{h}_i = \Big[\log(1+C_a), \log(1+C_m), U_a, U_m, C_{am}, \Delta_a, \Delta_m, \frac{A_i}{1+\bar{A}_a}, \frac{A_i}{1+\bar{A}_m}\Big]. \quad (1)$$

where U_a and U_m are unique-counterparty counts, C_{am} is the prior account–merchant pair count, and Δ denotes step recency. No current or future fraud label is used. Category and gender are one-hot encoded; amount, age, and step are retained. The transaction-only models omit all historical terms in Eq. (1).

3.3. Historical heterogeneous graph and GraphSAGE

The graph contains two node types: account nodes and merchant nodes. A training-period transaction creates an undirected account–merchant relation; repeated transactions collapse to one structural edge while each transaction remains a separate supervised edge example. Node features

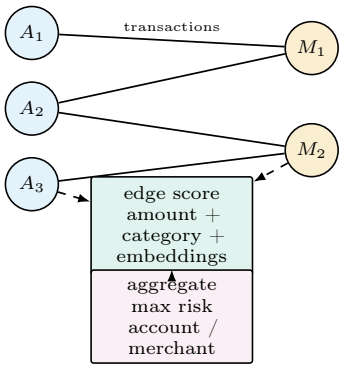

Figure 2: Heterogeneous risk construction.

combine a type indicator, account age/gender or merchant category, log interaction degree, log unique-counterparty count, and log mean amount from the permissible history.

For layer ℓ, mean-neighborhood GraphSAGE is implemented as given by (2).

$$\mathbf{z}_v^{(\ell+1)} = \sigma\left(W_s^{(\ell)}\mathbf{z}_v^{(\ell)} + W_n^{(\ell)} \sum_{u\in\mathcal{N}(v)} \frac{\mathbf{z}_u^{(\ell)}}{\sqrt{d_u d_v}}\right). \quad (2)$$

Two layers produce 32-dimensional entity embeddings. For transaction $i = (a, m)$, the edge classifier receives (3), where $\mathbf{x}_i$ contains transaction features.

$$\mathbf{q}_i = [\mathbf{z}_a, \mathbf{z}_m, |\mathbf{z}_a - \mathbf{z}_m|, \mathbf{z}_a \odot \mathbf{z}_m, \mathbf{x}_i], \quad (3)$$

Weighted binary cross-entropy compensates for the 47 positive training edges. Early stopping uses validation only. At locked-test time, node features and topology may use interactions through step 145 because that history precedes the test; model parameters are not retrained with validation labels. Three deterministic GraphSAGE seeds are averaged.

3.4. Baselines, fusion, and entity risk

Transaction LR is class-balanced logistic regression on transaction fields only. Transaction XGB uses the same fields with XGBoost. Causal Multi-XGB adds Eq. (1). The standalone GraphSAGE model tests whether relational representation is independently sufficient. The proposed fusion combines the causal XGBoost and three-seed GraphSAGE scores (4):

$$p_F(i) = (1-w)p_X(i) + wp_G(i), \qquad w \in \{0, .05, \ldots, 1\}. \quad (4)$$

Algorithm 1 Leakage-controlled graph-fusion fraud scoring

Require: Chronological transactions D, train end 110, validation end 145
Sort D by simulation step and original row order
for each transaction i in chronological order **do**
    Compute account, merchant, and pair histories from prior rows only
    Update histories only after storing features for i
end for
Fit transaction LR, transaction XGB, and causal Multi-XGB on steps 1–110
Build the account–merchant training graph from steps 1–110
for seed $s \in \{11, 42, 73\}$ **do**
    Train GraphSAGE edge classifier; early stop on validation
end for
Average GraphSAGE validation scores
Select fusion weight w and F_2 threshold on steps 111–145
Recompute graph state using permissible history through step 145
Score locked steps 146–180 once
Aggregate transaction scores to account and merchant risk with Eq. (6)

The validation set selects w by ; the selected value is $w = 0.65$. Each model then receives a validation-selected threshold maximizing

$$F_2 = \frac{5PR}{4P + R}, \tag{5}$$

The Eq. (5)emphasizes fraud recall while preserving precision. Locked-test labels do not influence either selection.

Transaction scores are converted into entity intelligence after inference. For entity e with test transactions $\mathcal{T}_e$, risk is (6):

$$R(e) = \max_{i \in \mathcal{T}_e} p(i). \tag{6}$$

An account or merchant is considered positive for evaluation if at least one of its locked-test transactions is fraudulent. This is an evaluation convention for risk concentration, not a statement that the entity itself is fraudulent.

4. Results and discussion

4.1. Transaction-level discrimination

Table 2 contains the central result. Transaction LR has the highest test at 0.8556 but only 0.2648 , illustrating the danger of interpreting ROC performance alone under 0.8% prevalence. Transaction XGB improves to 0.4251. Adding causal histories to XGBoost increases test recall at the validation-selected threshold from 0.4375 to 0.5000, but its falls to 0.4086.

Standalone GraphSAGE is the critical negative control. Its three-seed mean score reaches only 0.0746 and F1 0.1463. This result rejects a simplistic "graph is always better" narrative. With 47 training frauds and frequent cold-start accounts, graph supervision is too weak for the relational representation to act as a complete classifier. The fusion model nevertheless benefits from the graph score as a correction term. It obtains the highest (0.4500), F1 (0.4103), F_2 (0.4598), and MCC (0.4114). At the frozen threshold it detects 8/16 frauds with 15 false alerts, compared with 7/16 and 20 false alerts for transaction XGB.

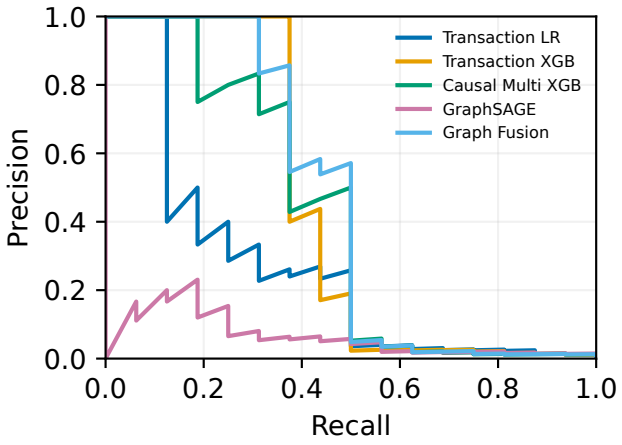

Figure 3: Locked-test precision–recall curves. (NOte: Graph fusion improves the high-precision portion of the ranking even though standalone GraphSAGE is weak.

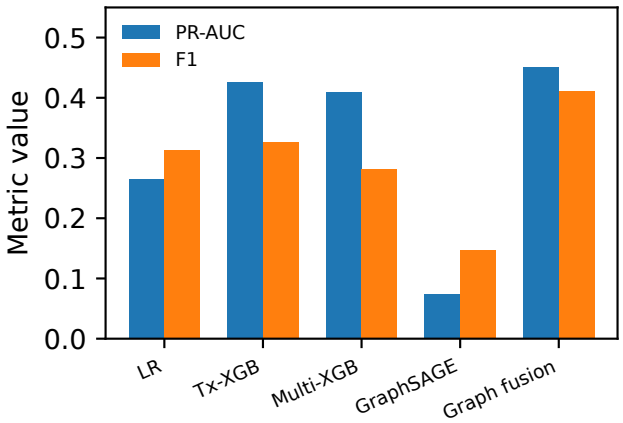

Figure 4: PR-AUC and frozen-threshold F1. (Note:The proposed model gains from fusion; the graph-only result is retained as a negative control.

4.2. Merchant and Account Risk Assessment

The paper's second question is whether a transaction detector concentrates risk at the entity level. Table 3 uses Eq. (6). Transaction XGB reaches account 0.4295 and merchant 0.4353. Graph fusion increases these values to 0.4502 and 0.4654, respectively. The merchant gain is consistent with the graph coverage audit: almost every test transaction points to a merchant already present in training, so merchant embeddings have historical structural support. Account improvement is smaller because more than half of test transactions involve accounts unseen in training.

The unseen-account audit strengthens the interpretation. Among 1,062 test transactions from accounts absent in training, 11 are fraudulent. Transaction XGB reaches 0.4464, Causal Multi-XGB 0.4564, and Graph Fusion 0.4973. The gain therefore cannot be explained solely by memorizing training account IDs. For the 938 transactions from seen accounts, however, Graph Fusion is 0.4060, essentially equal to Transaction XGB at 0.4088.

Table 2: Locked-Test Transaction Performance. (Note: Thresholds Are Selected on Validation F_2 Only.

Model			Precision	Recall	F1	F_2	MCC	FP / FN
Transaction LR	**0.8556**	0.2648	0.3125	0.3125	0.3125	0.3125	0.3070	11 / 11
Transaction XGB	0.8305	0.4251	0.2593	0.4375	0.3256	0.3846	0.3299	20 / 9
Causal Multi-XGB	0.8134	0.4086	0.1951	**0.5000**	0.2807	0.3810	0.3039	33 / 8
GraphSAGE, 3-seed mean	0.8214	0.0746	0.1200	0.1875	0.1463	0.1685	0.1415	22 / 13
Graph Fusion	0.8189	**0.4500**	**0.3478**	**0.5000**	**0.4103**	**0.4598**	**0.4114**	15 / 8

Table 3: Entity-Level Locked-Test Risk Concentration

Model	Account	Merchant
Transaction LR	0.2651	0.2765
Transaction XGB	0.4295	0.4353
Causal Multi-XGB	0.4088	0.4241
GraphSAGE	0.0749	0.0858
Graph Fusion	**0.4502**	**0.4654**

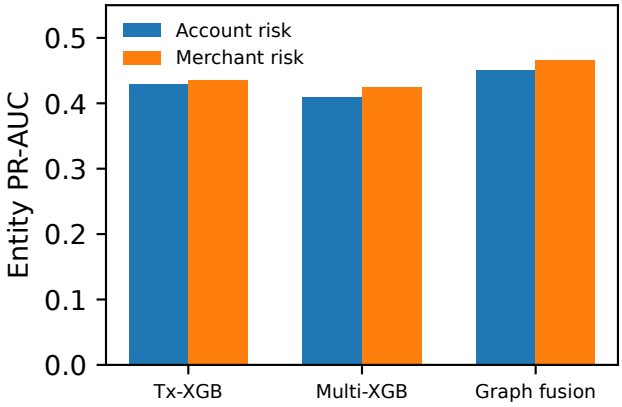

Figure 5: Entity-level PR-AUC after max aggregation of locked-test transaction scores.

The graph signal is thus most useful as a contextual regularizer in the inductive subgroup rather than as a universal improvement.

4.3. Analyst budgets and graph instability

Operationally, fraud teams review only a small score-ranked fraction. Figure 6 reports retrieval under fixed alert budgets. At the tightest 0.25% budget (five alerts), both Transaction XGB and Graph Fusion recover five frauds with no false alerts. At 1% (20 alerts), Graph Fusion and Causal Multi-XGB recover 8/16 frauds, while Transaction XGB recovers 7/16. The proposed model therefore gains one fraud at the same 20-alert capacity. Beyond 2%, all three leading models plateau at eight detected frauds in this small extract, showing that the remaining frauds are not simply recoverable by expanding the queue modestly.

GraphSAGE training is notably unstable (Figure 7). Seed-specific standalone values are 0.0558, 0.1324, and 0.0399. This dispersion is expected with only 47 positive training edges and provides a second reason not to deploy the graph score alone. The proposed protocol averages three graph seeds before validation fusion. Averaging does not make the graph model strong by itself; it reduces

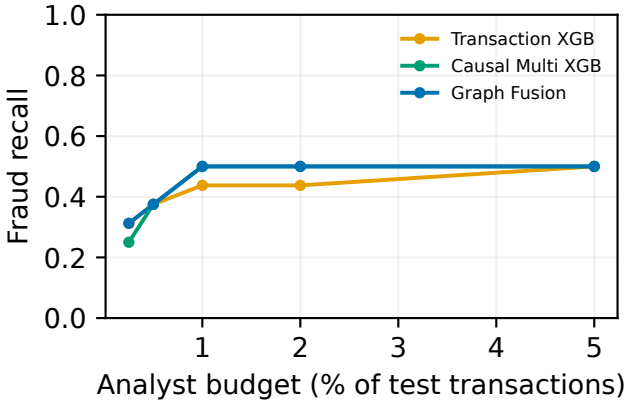

Figure 6: Fraud recall under fixed analyst-review budgets.

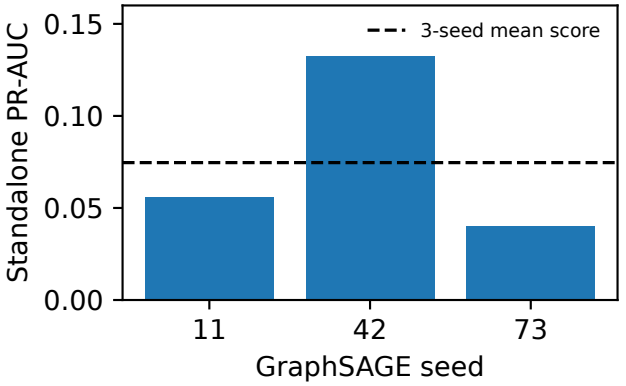

Figure 7: Standalone GraphSAGE PR-AUC across deterministic seeds. (Note: Dashed line is the PR-AUC obtained from the mean of the three score vectors.

variance sufficiently for the graph score to serve as a complementary signal. Future work on a larger BankSim release or real payment graph should test more stable objectives, class-balanced neighborhood sampling, temporal message passing, and calibration.

4.4. Feature diagnostics and limitations

The causal XGBoost diagnostics in Figure 8 rank log amount and raw amount first, followed by gender indicators, category fields, merchant recency, and amount deviation from account history. These are predictive associations only. BankSim's simulator can encode relationships between fraud scenarios, amount, and merchant category that need not transfer to a production bank. Importantly,

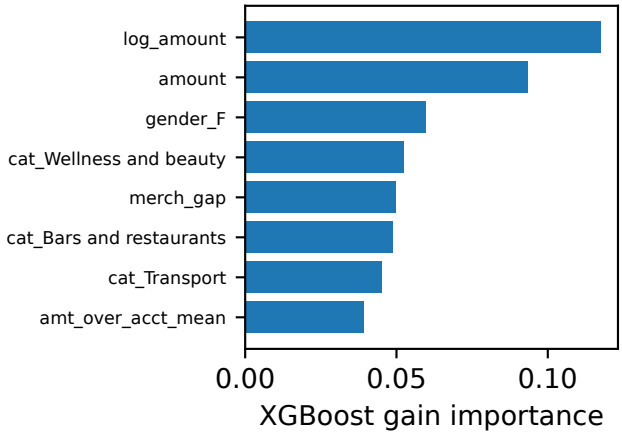

Figure 8: Top causal Multi-XGB gain importances.

no feature name is interpreted as a causal mechanism of fraud.

Four limitations bound the claims. First, the data are synthetic and the packaged file is only a 10,000-row public extract; absolute metrics should not be generalized to real payment systems. Second, the locked test contains only 16 frauds, so one recovered event changes recall by 6.25 percentage points. Third, the graph has only account–merchant relations; real payment intelligence also includes devices, IP addresses, cards, beneficiaries, authentication events, and merchant ownership. Fourth, graph state is refreshed at partition boundaries rather than after every locked-test transaction. This is intentionally conservative because no future test edge can influence another score, but it omits legitimate online graph adaptation.

5. Conclusion

This work evaluated multisource fraud intelligence in transaction, account, merchant and graph representations under a chronological BankSim protocol. The main result is intentionally qualified. Transaction-only XGBoost is a strong ranker by itself with test 0.4251. Causal account and merchant histories improve recall, but not ranking. Standalone GraphSAGE is unstable and weak with only 0.0746 after three-seed score averaging. However, a validation selected fusion of causal XGBoost and GraphSAGE achieves of 0.4500, F1 of 0.4103 and MCC of 0.4114, detecting 8/16 locked frauds and 15 false alarms. The fusion at entity level achieves of 0.4502 for account risk and 0.4654 for merchant risk. It particularly performs well on transactions from unseen accounts (0.4973).

The implication for deployment is architectural rather than promotional: relational embeddings should be viewed as one evidence channel in a larger fraud system, absent demonstrated standalone stability. In the case of a graph with few labels, a strong tabular model can still be the primary detector, with graph representations providing a contextual correction. Bigger benchmarks should evaluate temporal GNNs, richer node types, delayed adjudication, calibration and drift. All customer, merchant or financial-loss claims must be verified by a real bank.

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 Claude 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

Yeswanth Dintakurthy: Conceptualization, Methodology, Investigation, Visualization, Writing- Original draft & editing.

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] Abdallah A, Maarof MA, Zainal A. Fraud detection system: A survey. *Journal of Network and Computer Applications*. 2016;68:90–113. Available from: https://doi.org/10.1016/j.jnca.2016.04.007

[2] Hirnschall D. Semi-supervised Bayesian GANs with log-signatures for uncertainty-aware credit card fraud detection. *Mathematics*. 2025;13(19):3229. Available from: https://doi.org/10.3390/math13193229

[3] Motie S, Raahemi B. Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications*. 2024;240:122156. Available from: https://doi.org/10.1016/j.eswa.2023.122156

[4] Cheng D, Zou Y, Xiang S, et al. Graph neural networks for financial fraud detection: A review. *Frontiers of Computer Science*. 2025;19:199609. Available from: https://doi.org/10.1007/s11704-024-40474-y

[5] Lu M, Han Z, Rao SX, et al. BRIGHT: Graph neural networks in real-time fraud detection. In: *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. 2022. p. 3342–3351. Available from: https://doi.org/10.1145/3511808.3557136

[6] Ren L, Hu R, Li D, et al. Dynamic graph neural network-based fraud detectors against collaborative fraudsters. *Knowledge-Based Systems*. 2023;278:110888. Available from: https://doi.org/10.1016/j.knosys.2023.110888

[7] Lou C, Wang Y, Li J, Qian Y, Li X. Graph neural network for fraud detection via context encoding and adaptive aggregation. *Expert Systems with Applications*. 2025;261:125473. Available from: https://doi.org/10.1016/j.eswa.2024.125473

[8] Li E, Ouyang J, Xiang S, et al. Efficient relation-aware heterogeneous graph neural network for fraud detection. *World Wide Web*. 2025;28:55. Available from: https://doi.org/10.1007/s11280-025-01369-5

[9] Wei S, Lee S. Internet fraud transaction detection based on temporal-aware heterogeneous graph oversampling and attention fusion network. *PLoS ONE*. 2025;20:e0337208. Available from: https://doi.org/10.1371/journal.pone.0337208

[10] Chen C, Liang C, Lin J, et al. A graph-powered large-scale fraud detection system. *International Journal of Machine Learning and Cybernetics*. 2024;15:115–128. Available from: https://doi.org/10.1007/s13042-023-01786-w

[11] Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems*. 2017;30. Available from: https://doi.org/10.48550/arXiv.1706.02216

[12] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: *Proceedings of the International Conference on Learning Representations*. 2018. Available from: https://doi.org/10.48550/arXiv.1710.10903

[13] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. p. 785–794. Available from: https://doi.org/10.1145/2939672.2939785

[14] Mogili VB. Empowering Human-AI Collaboration: Enterprise Technology Platforms and Human Expertise Synergy in Healthcare, Finance, and Scientific Research. *Journal of Computer Science and Technology Studies*. 2025 Jul 10;7(7):513–20. Available from: https://doi.org/10.32996/jcsts.2025.7.7.57

[15] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning*. 2006. p. 233–240. Available from: https://doi.org/10.1145/1143844.1143874

[16] Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*. 2015;10(3):e0118432. Available from: https://doi.org/10.1371/journal.pone.0118432

[17] Dintakurthy Y, Rama Krishna Innmuri, Ashutosh Vanteru, Arun Thotakuri. Emerging Applications of Artificial Intelligence in Edge Computing: A Comprehensive Review. *Journal of Modern Technology*. 2025 Jan. 26;1(2):175-8. Available from: https://doi.org/10.71426/jmt.v1.i2.pp175-185