Received: 20 July 2026 | Revised: 10 September 2026 | Accepted: 9 October 2026 | Published online: 26 November 2026

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2026, Vol. 00(00) 1–9
DOI: 10.47852/bonviewJDSIS620211464

BON VIEW PUBLISHING

Graph Context as a Residual Correction: Leakage-Controlled Graph-Tabular Fusion for Illicit Bitcoin Transaction Detection

Mohamed M. Abbassy[1,*], Amr E. Elshora[2] and Ayman Abo-Alnadr[3]

[1] *Faculty of Computers and Artificial Intelligence, Beni-Suef University, Egypt*

[2] *Department of Computer Science and Information Systems, Higher Institute of Management and Information Technology, Egypt*

[3] *Department of Information Systems, Higher Institute of Management and Information Technology, Egypt*

Abstract: Cryptocurrency fraud detection requires models that use both transaction attributes and relational context while preventing temporal leakage. This study proposes a leakage-controlled temporal graph-tabular residual fusion framework for Bitcoin fraud detection using the Elliptic++ benchmark. The native branch models 182 non-time transaction attributes with ExtraTrees, whereas a four-head role-aware Temporal Star Graph Attention Network (Temporal Star-GAT) aggregates same-time input- and output-wallet representations derived from 55 non-time actor attributes. The two risk scores are fused in logit space, with the fusion weight and decision threshold selected exclusively on validation data. Elliptic++ contains 203,769 transactions across 49 time steps; 46,564 are labeled (4,545 illicit and 42,019 licit), and 157,205 are unknown. We use time steps 1–20 for training, 21–35 for validation, and 36–49 for a locked test. Across five matched seeds, the proposed fusion achieved 91.864 ± 0.130% receiver operating characteristic area under the curve (ROC-AUC), 75.636 ± 0.081% precision-recall area under the curve (PR-AUC), 76.497 ± 0.697% F1, 89.782 ± 2.378% precision, 66.659 ± 0.373% recall, 0.477 ± 0.127% false-positive rate, and 0.7619 ± 0.0094 Matthews correlation coefficient. Compared with the ExtraTrees branch, paired bootstrap differences were + 0.134 percentage points in ROC-AUC and + 0.110 points in PR-AUC (both $p < 0.001$). The top 5% risk-ranked transactions captured 67.9% of illicit transactions at 79.9% precision. The results show that relational graph information provides a small but consistent residual correction while temporal and calibration diagnostics expose important deployment limitations.

Keywords: cryptocurrency fraud detection, Elliptic++, graph attention networks, graph-tabular fusion, temporal evaluation

1. Introduction

Cryptocurrency ecosystems combine transparent ledgers with pseudonymous identities, creating a security setting in which large volumes of observable transactions can coexist with difficult attribution and rapidly evolving fraud strategies [1]. Financial-fraud research increasingly treats such data as relational rather than purely tabular because laundering chains, phishing campaigns, scam clusters, and coordinated address activity are expressed through interactions among entities [2, 3].

Recent blockchain studies have therefore explored graph neural networks (GNNs) for phishing and fraudulent-account detection [4–6]. Temporal graph attention has been used to model evolving Ethereum behavior [7], transaction-network embeddings have been used to represent accounts [8], and dynamic or multi-relational graph models have been proposed for financial-safety applications [9, 10]. Anti-money-laundering comparisons and temporally evolving phishing detectors further demonstrate the value of graph context [11, 12], while early-warning work on scam tokens shows the importance of modeling how illicit patterns change over time [13].

Three methodological issues nonetheless remain unresolved. The first is chronological leakage: results are easily inflated when future transactions, later actor states, or randomly mixed neighborhoods find their way into training. The second is the strength of the tabular baseline. Native transaction attributes are often highly predictive on their own, and a graph model should therefore be asked to show what its relational branch adds rather than be assumed to dominate. The third concerns what a single AUC value can support. Class imbalance, false-positive burden, ranked triage, temporal degradation, calibration, and uncertainty all bear on whether a score can drive preventive controls, and none of them is visible in an aggregate discrimination figure.

***Corresponding author:** Mohamed M. Abbassy, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Egypt. Email: mmabbassy@fcis.bsu.edu.eg

© The Author(s) 2026. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

This study addresses these issues through a leakage-controlled temporal graph-tabular residual fusion framework. Each labeled Bitcoin transaction is represented by its native Elliptic++ transaction attributes and by same-time input/output wallet context. A role-aware Temporal Star Graph Attention Network (Temporal Star-GAT) learns the relational component, while an ExtraTrees branch models the native transaction attributes. The two scores are combined only after validation-based model selection, so the graph branch acts as a residual relational correction rather than an assumed dominant predictor.

The main contributions are:

1) A strict chronological evaluation protocol using time steps 1–20 for training, 21–35 for validation, and 36–49 for a locked test, with exact transaction time read from Elliptic++ and unknown labels excluded from supervised learning.
2) A role-aware Temporal Star-GAT that separately attends to input- and output-wallet mentions at the same time step and combines them with native transaction representations.
3) A validation-selected graph-tabular residual fusion in logit space, accompanied by a transparent ablation showing the standalone ExtraTrees and Star-GAT branches.
4) Five-seed evaluation with receiver operating characteristic area under the curve (ROC-AUC), precision-recall area under the curve (PR-AUC), F1, precision, recall, false-positive rate (FPR), Matthews correlation coefficient (MCC), paired bootstrap confidence intervals, and matched-seed Wilcoxon tests.
5) Operational diagnostics including Top-K fraud capture, per-time-step temporal robustness, calibration analysis, and a preventive risk-control interpretation for analyst review and transaction intervention.

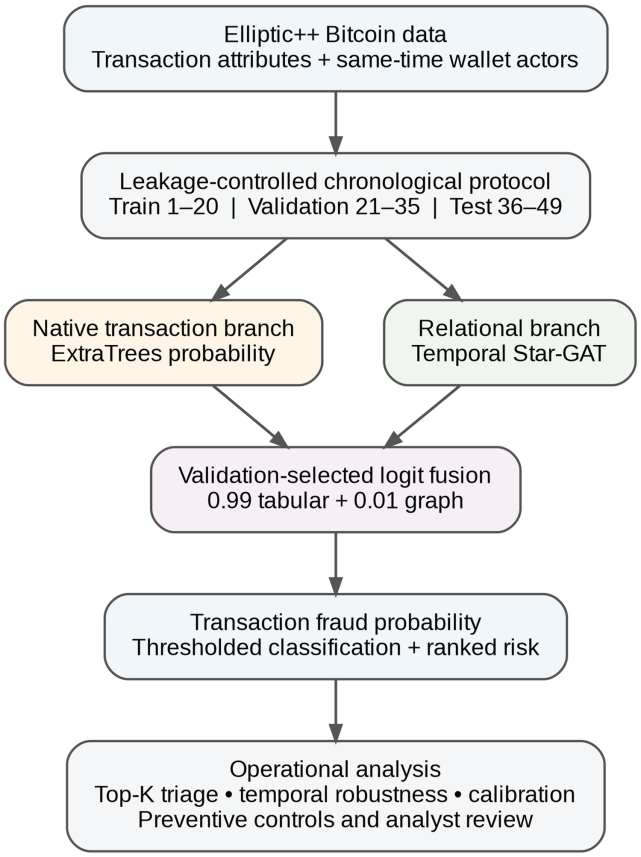
The remainder of the paper is organized as follows. Section 2 reviews graph-based fraud detection and methodological foundations. Section 3 defines the research problem and evaluation questions. Section 4 presents the proposed architecture and training procedure. Section 5 describes the dataset, temporal protocol, baselines, and reproducibility controls. Section 6 reports the results, statistical analysis, operational diagnostics, preventive-security implications, and limitations. Section 7 concludes the study and identifies actionable future work.

2. Related Research

Graph learning has become a central framework for representing non-Euclidean relational data [14, 15]. Recent relational deep-learning research emphasizes that predictive learning can benefit from jointly representing entities, relations, and contextual attributes [16], while heterogeneous GNNs have been used in fraud-detection settings that require different entity or relation semantics [17]. These ideas are relevant to blockchain analytics because transaction ledgers naturally contain linked entities, directionality, repeated actors, and time.

Earlier graph approaches such as Node2Vec learn structural embeddings through random walks [18], whereas Graph Convolutional Networks propagate and aggregate neighboring information end to end [19]. Graph Attention Networks extend message passing by learning data-dependent neighbor weights [20]. Attention is useful for selective aggregation, but it is not by itself an explanation; explainable-GNN research distinguishes predictive attention from faithful post hoc or counterfactual explanation criteria [21]. Figure 1 provides an overview of the leakage-controlled temporal graph-tabular residual fusion framework.

Figure 1
Overview of the leakage-controlled temporal graph-tabular residual fusion framework



The Elliptic++ benchmark was introduced to support financial-forensics research on both Bitcoin transactions and wallet actors [22]. Its combination of transaction attributes, wallet attributes, directed relations, time steps, and unknown labels makes it especially useful for leakage-aware evaluation. At the same time, large graph learning raises practical concerns regarding sampling [23] and distributed training [24]. These concerns motivate a local same-time star construction in the present study rather than unrestricted message passing over the entire future-expanded graph.

Robustness and explainability are also security requirements. Graph models can be attacked through structural or feature perturbations [25], and inductive neighborhood aggregation such as GraphSAGE is often used when generalization to evolving graph regions is required [26]. Recent GNN explainability research provides established taxonomies and evaluation frameworks for graph explanations [27], while more recent surveys stress adversarial robustness [28] and the privacy/security risks of graph-learning systems themselves [29]. Explainable financial-fraud systems such as xFraud further show the value of connecting model scores to investigator-facing evidence [30].

What distinguishes the present study from work that evaluates a graph classifier in isolation is the question it asks: whether relational attention contributes predictive information beyond a strong native-attribute ensemble, measured under a locked chronological split. This design also separates three questions

that are often conflated: discrimination, temporal robustness, and operational usefulness.

3. Problem Statement and Research Questions

Let a labeled Bitcoin transaction *i* occur at time step t_i. The available information includes a native transaction-feature vector x_*i* and a set of input- and output-wallet mentions observed at the same time step. The objective is to estimate $P(y_i = 1 \mid x_i, N_i, t_i)$, where $y_i = 1$ denotes an illicit transaction and *N_i*, t_*i* denotes the role-aware wallet context available at that time. The central constraint is temporal: information from later time steps must not influence model fitting, normalization, hyperparameter selection, threshold selection, or graph-context construction for earlier observations.

The study evaluates four research questions: (RQ1) how strong are native transaction attributes under a leakage-controlled temporal split; (RQ2) how much information is captured by the relational Star-GAT branch alone; (RQ3) whether a validation-selected graph residual improves a strong tabular baseline consistently across matched seeds; and (RQ4) whether the resulting scores are useful under realistic class imbalance when evaluated through false positives, Top-*K* triage, temporal drift, and calibration.

4. Proposed Methodology

4.1. Leakage-controlled transaction-centered graph representation

For each transaction *i*, we construct a local directed star centered on the transaction. Input-wallet mentions and output-wallet mentions are connected to the transaction with distinct role identifiers. Only wallet attributes recorded at the same Elliptic++ time step as the transaction are joined. This same-time constraint prevents actor states from future steps from being used as context for earlier transactions. Figure 2 illustrates the detailed architecture of the native transaction branch, role-aware Temporal Star-GAT, and validation-selected fusion.

The transaction branch uses 182 non-time attributes from txs_features.csv. The wallet branch uses 55 non-time actor attributes from wallets_features.csv; the dataset time-step variable is retained for temporal alignment but is not treated as a free predictive feature. Input and output roles are preserved separately rather than collapsing all neighbors into one unordered set.

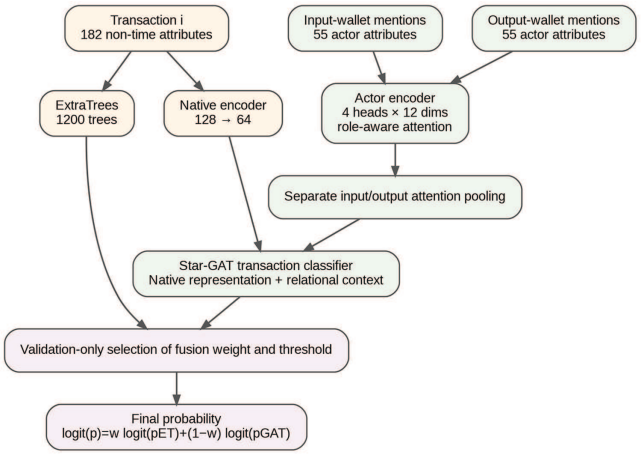
4.2. Native transaction branch

The native branch is an ExtraTrees ensemble trained only on labeled training transactions. ExtraTrees was selected as the strongest tabular reference during validation and uses 1,200 randomized trees, square-root feature subsampling, a minimum leaf size of one, and balanced class weights. This branch provides the tabular probability used in the fusion.

4.3. Temporal Star-GAT relational branch

The relational branch follows the attention principle of GAT [20] but uses a transaction-centered, role-aware star. A two-layer native encoder maps x_i through 128 and 64 hidden dimensions with LayerNorm and Gaussian Error Linear Unit (GELU). Wallet attributes are transformed into four attention heads of 12 dimensions each. A learned role embedding distinguishes input from output-wallet mentions, as set out in Equation (1).

Figure 2
Detailed architecture of the native transaction branch, role-aware Temporal Star-GAT, and validation-selected fusion



$$h_i = f_{\text{tx}}(x_i), \quad z_{ij}^{h} = f_a(a_j, t_i)^{h} + r_{\text{role}(j)}^{h} \tag{1}$$

For head *h*, each transaction–wallet pair is assigned a scaled dot-product score $s_{ij}^{h} = (q_i^{h})^{\mathrm{T}} z_{ij}^{h}/\sqrt{d}$. Softmax normalization is then performed separately for input and output roles, as expressed in Equation (2), so a large input neighborhood does not suppress output-wallet evidence or vice versa. Equation (3) forms a role-specific multi-head aggregate and then concatenates the input- and output-role representations.

$$\alpha_{ij}^{h,r} = \frac{\exp\left(s_{ij}^{h}\right)}{\sum_{k \in N_r(i)} \exp\left(s_{ik}^{h}\right)}, \quad r \in \{\text{in}, \text{out}\} \tag{2}$$

$$g_i^{r} = \text{concat}_{h=1}^{H}\left(\sum_{j \in N_r(i)} \alpha_{ij}^{h,r} z_{ij}^{h}\right), \quad r \in \{\text{in}, \text{out}\} \tag{3}$$

The input- and output-role vectors are then concatenated as $g_i = [g_i^{\text{in}}; g_i^{\text{out}}]$. The pooled graph representation is normalized and multiplied by a learnable sigmoid gate before concatenation with the 64-dimensional native neural representation. The classifier uses hidden dimensions 96 and 32 with GELU and dropout, producing the Star-GAT probability of the graph branch. This attention mechanism is treated as a learnable aggregation rule; we do not claim that raw attention weights constitute faithful explanations.

4.4. Residual logit fusion

The final fused logit score s_i combines the ExtraTrees and Star-GAT probabilities in logit space, as given in Equation (4).

$$s_i = w\text{logit}\left(p_{ET,i}\right) + (1 - w)\,\text{logit}\left(p_{GAT,i}\right) \tag{4}$$

Here, $\text{logit}(p) = \ln[p/(1 - p)]$ is the log-odds transform of a probability $p \in (0, 1)$; the inverse-logit (sigmoid) $\sigma(s) = 1/(1 + e^{-s})$ maps the fused logit score s_i back to probability space, so $p_i = \sigma(s_i)$. Thus, Equation (4) performs a weighted combination in log-odds space. The global fusion weight *w* is selected using validation

data only from a grid over probability, logit, and rank fusion. The validation objective is mean PR-AUC with a small ROC-AUC tie-breaking term across the matched seeds. The selected configuration was logit fusion with $w = 0.99$ for ExtraTrees and 0.01 for Star-GAT. This small graph weight is reported transparently and is interpreted as evidence that the graph branch adds a modest residual signal rather than dominating prediction.

4.5. Training objective and threshold selection

The Star-GAT is optimized with weighted binary cross-entropy. For the reported five-seed runs, the positive-class weight is $(N_{neg}/N_{pos})^{0.35}$, matching the --pw-exp 0.35 setting used in the released run script; this moderates class imbalance without applying the full negative-to-positive ratio. AdamW is used with learning rate 1.2×10^{-3}, weight decay 2×10^{-4}, batch size 1,024, gradient clipping at 5, and a maximum of 20 epochs. Early stopping uses validation PR-AUC plus 0.02 × ROC-AUC, with a patience of five and a minimum of 15 epochs. The relational gate logit is initialized at − 2.2 before sigmoid transformation. Five matched random seeds are evaluated: 42, 7, 21, 100, and 2025.

The classification threshold is selected on validation data by maximizing validation F1 and is then frozen for the test set. No test labels are used for hyperparameter, fusion-weight, or threshold selection.

4.6. Evaluation metrics and statistical analysis

Because only 5.9% of the labeled test transactions are illicit, ROC-AUC is complemented by PR-AUC, F1, precision, recall, FPR, and MCC. PR-AUC is emphasized for ranking under class imbalance, while MCC summarizes all four confusion-matrix cells. Top-K precision and recall evaluate investigator-facing prioritization.

Uncertainty is evaluated in two ways. First, model variability is reported as mean ± sample standard deviation across five matched seeds. Second, a paired stratified bootstrap resamples illicit and licit held-out transactions 2,000 times and compares mean-of-five proposed probabilities with the corresponding ExtraTrees probabilities. We also report the exact paired Wilcoxon signed-rank result across five matched seeds. This combination distinguishes transaction-level uncertainty from seed-level variability.

4.7. Preventive and secure deployment interpretation

A risk score prevents nothing until it is tied to an operational policy. In a deployment pipeline, low-risk transactions could proceed normally, medium-risk transactions could trigger step-up authentication or multi-signature approval, and high-risk transactions could be delayed for analyst review or temporarily frozen when legally and operationally permitted. Transaction value, account history, and external threat intelligence can be combined with the model score so that a single statistical output does not automatically block legitimate users.

The same framework can expose investigator-facing evidence such as the transaction score, high-risk neighboring wallet mentions, input/output roles, and the transaction attributes associated with the alert. For secure collaborative use, future implementations should consider access control, secure aggregation, federated training, differential privacy, audit logging, and defenses against data poisoning, inference attacks, and adversarial graph manipulation [28, 29].

5. Experimental Setup

5.1. Elliptic++ dataset and labels

Experiments use the public Elliptic++ Bitcoin dataset [22]. The transaction component contains 203,769 transactions observed over 49 time steps. Of these, 4,545 are labeled illicit, 42,019 are labeled licit, and 157,205 have unknown labels. The transaction-feature file provides 183 features when the time step is counted, corresponding to 182 non-time transaction attributes used by the model. The actor component contains 822,942 unique wallet addresses represented by 1,268,260 temporal wallet rows; 56 actor features are provided when the time step is counted, corresponding to 55 non-time actor attributes used by the relational branch.

Unknown transaction labels are retained for dataset accounting but excluded from supervised training, validation, threshold selection, and test metrics. The study does not manufacture edge labels or infer illicit status from neighboring actors. The target is the native transaction label supplied in txs_classes.csv, so no edge-level ground truth is required at any point.

5.2. Exact temporal protocol

Exact transaction time is read from txs_features.csv. The chronological partition is fixed before model fitting: time steps 1–20 for training, 21–35 for validation, and 36–49 for testing. The test set is not reused for tuning. Training statistics are used for neural standardization, and same-time actor features are joined by the pair (address, time step). Figure 3 and Table 1 summarize the protocol.

The corresponding numbers of unknown transactions are 71,873, 38,664, and 46,668 in train, validation, and test, respectively. These observations are excluded from supervised metrics but remain part of the public dataset.

Figure 3
Exact chronological split and leakage-control protocol used in all revised experiments

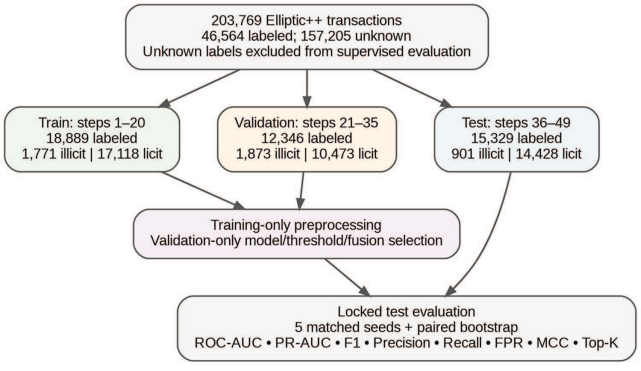


Table 1
Exact Elliptic++ transaction split used for supervised evaluation

Split	Time steps	Total tx	Labeled	Illicit	Licit
Train	1–20	90,762	18,889	1,771	17,118
Validation	21–35	51,010	12,346	1,873	10,473
Test	36–49	61,997	15,329	901	14,428

5.3. Baselines and ablation design

The comparison centers on ExtraTrees as the strongest native-attribute branch, the Temporal Star-GAT as a graph-only/relational ablation, and the proposed Graph-Tabular Fusion. All three models use the same chronological split and native transaction target and are evaluated over five matched seeds. This design directly tests whether the relational branch contributes predictive information beyond a strong native-attribute model.

The ablation is deliberately blunt. If the fusion improves on ExtraTrees while the Star-GAT alone is weaker, what has been demonstrated is a residual relational contribution, not the general superiority of graph learning. This is the interpretation the paper defends.

5.4. Reproducibility controls

The accompanying reproducibility package includes data-preparation scripts, Star-GAT training code, ExtraTrees training code, fusion-selection code, diagnostics, five-seed logs, validation-search outputs, bootstrap and Wilcoxon results, calibration bins, Top-K metrics, and high-resolution figures. Dataset files are referenced by their original Elliptic++ filenames. The code records the five random seeds and writes per-seed validation/test metrics to CSV, allowing the reported aggregate values to be independently checked.

Deployment throughput, inference latency, scaling behavior, and adversarial robustness were not measured, and no claim is made about any of them. What the experiments support is discrimination and ranking quality under the protocol described above; anything beyond that would require a separate study.

6. Results and Discussion

6.1. Main fraud-detection performance

Table 2 reports the principal test metrics. The proposed fusion achieved 91.864 ± 0.130% ROC-AUC, 75.636 ± 0.081% PR-AUC, and 76.497 ± 0.697% F1 across five matched seeds. ExtraTrees achieved 91.731 ± 0.130% ROC-AUC, 75.492 ± 0.069% PR-AUC, and 76.058 ± 0.774% F1. The standalone Star-GAT was materially weaker in PR-AUC and F1, showing that the native transaction attributes carry most of the predictive signal on this split. Nevertheless, adding the small validation-selected graph residual improved all five paired fusion runs.

Figures 4 and 5 show the mean-of-five ROC and precision-recall curves against the mean ExtraTrees reference. The curves nearly coincide, which follows directly from the 0.99/0.01 fusion weight. The graph branch is best read as a small residual correction, not as an architectural replacement.

Figure 4
ROC curves for ExtraTrees and the proposed mean-of-five graph-tabular fusion on the locked test set

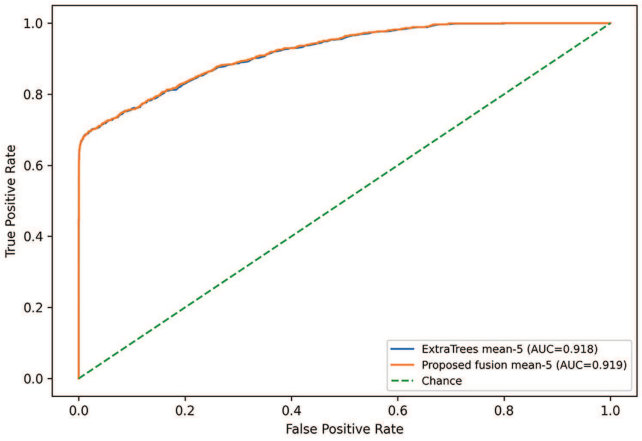
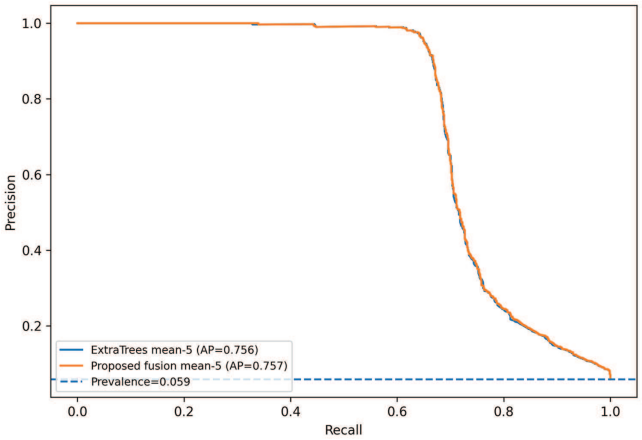


Figure 5
Precision-recall curves for ExtraTrees and the proposed mean-of-five graph-tabular fusion on the locked test set



6.2. Precision, recall, false positives, and operating point

At seed-specific validation-selected thresholds (Table 3), the proposed model obtained 89.782 ± 2.378% precision, 66.659 ± 0.373% recall, and 0.477 ± 0.127% FPR. ExtraTrees obtained 88.426 ± 2.462% precision, 66.748 ± 0.338% recall, and 0.549 ± 0.135% FPR. The gain therefore appears as fewer false positives and higher precision at essentially unchanged recall.

For the mean-of-five probability ensemble, the validation-selected threshold was 0.2944. On the test set, this operating point produced 603 true positives, 76 false positives, 14,352 true

Table 2
Held-out test performance under the exact temporal protocol. All three models are evaluated over five matched seeds and reported as mean ± SD; values are percentages except MCC

Model	ROC-AUC	PR-AUC	F1	MCC
ExtraTrees	91.731 ± 0.130	75.492 ± 0.069	76.058 ± 0.774	0.7561 ± 0.0102
Temporal Star-GAT	90.214 ± 0.569	58.086 ± 3.110	54.193 ± 3.833	0.539 ± 0.038
Proposed fusion	91.864 ± 0.130	75.636 ± 0.081	76.497 ± 0.697	0.7619 ± 0.0094

Table 3
Threshold-dependent metrics across five matched seeds (mean ± SD)

Model	Precision (%)	Recall (%)	FPR (%)	MCC
ExtraTrees	88.426 ± 2.462	66.748 ± 0.338	0.549 ± 0.135	0.7561 ± 0.0102
Proposed fusion	89.782 ± 2.378	66.659 ± 0.373	0.477 ± 0.127	0.7619 ± 0.0094

negatives, and 298 false negatives, corresponding to 88.807% precision, 66.926% recall, 0.527% FPR, and 76.329% F1. This operating point is examined further in Section 6.7.

6.3. Statistical comparison with the strongest tabular baseline

A paired stratified bootstrap over the held-out transactions compared the mean-of-five proposed probabilities with the ExtraTrees probabilities. The proposed-minus-ExtraTrees ROC-AUC difference was + 0.134 percentage points with a 95% bootstrap confidence interval of + 0.108 to + 0.161 points. The PR-AUC difference was + 0.110 points with a 95% interval of + 0.074 to + 0.149 points. None of the 2,000 bootstrap replicates crossed zero for either metric; accordingly, we report $p < 0.001$ rather than $p = 0$.

Across the five matched seeds, the proposed model improved ROC-AUC, PR-AUC, F1, FPR, and MCC in the same direction in every run. With only five pairs, however, the exact two-sided Wilcoxon p-value is 0.0625; the directional improvement test is 0.03125. The revision therefore avoids presenting the seed-level result as definitive two-sided significance and instead reports both transaction-level bootstrap uncertainty and seed-level evidence.

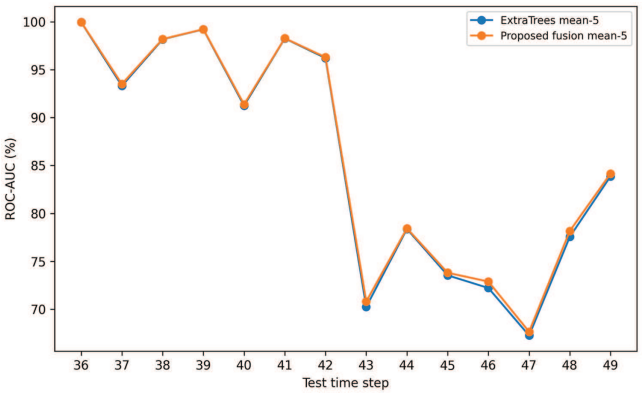
6.4. Operational Top-*K* risk ranking

In many fraud operations, analysts review only the highest-risk fraction of activity. Table 4 evaluates this setting. The top 1% of ranked test transactions contained 153 illicit transactions with 100% precision and 17.0% recall. Expanding review to the top 5% captured 612 illicit transactions, corresponding to 79.9% precision and 67.9% recall. This result provides an interpretable workload-performance trade-off that is more actionable than AUC alone.

6.5. Temporal robustness and concept-drift signal

Aggregate test metrics can hide time-dependent degradation. Figure 6 reports ROC-AUC separately for time steps 36 – 49. The proposed model ranges from approximately 99.96% ROC-AUC at step 36 to 67.65% at step 47. Performance drops are concentrated in several later low-prevalence periods, where only a small number of illicit transactions are available. The graph residual generally tracks or slightly improves the ExtraTrees curve, but it does not remove temporal variability.

Figure 6
Per-time-step ROC-AUC on test steps 36–49, illustrating temporal variability and potential concept drift



This finding is important for deployment: a fixed offline model should not be assumed stable under concept drift. Production systems should monitor prevalence, alert yield, calibration, and delayed labels, and should define retraining or recalibration triggers. Future evaluation should also model chain reorganizations, emerging addresses, streaming updates, and adversarial pattern changes rather than extrapolating from a single static test interval.

6.6. Calibration analysis

The raw proposed fusion obtained a Brier score of 0.0246 and a 10-bin expected calibration error (ECE) of 0.0407. A Platt scaling model fitted on validation data did not improve calibration on the temporally shifted test set; its Brier score increased to 0.0333 and ECE to 0.0943. Therefore, the raw fusion probabilities are retained. Figure 7 illustrates the reliability relationship and demonstrates why post hoc calibration should be validated temporally rather than assumed beneficial.

6.7. Confusion matrix and class-specific error burden

Figure 8 shows that the selected operating point favors precision and low false-positive burden: only 76 licit transactions are flagged among 14,428 licit test transactions. The upper-left

Table 4
Operational risk-ranking performance of the proposed mean-of-five fusion

Review cutoff	Transactions	Frauds captured	Precision@*K*	Recall@*K*
Top 1%	153	153	100.0%	17.0%
Top 5%	766	612	79.9%	67.9%
Top 10%	1,533	653	42.6%	72.5%
Top 100	100	100	100.0%	11.1%
Top 500	500	496	99.2%	55.0%

Figure 7
Reliability diagram for the raw proposed fusion and validation-fitted Platt calibration

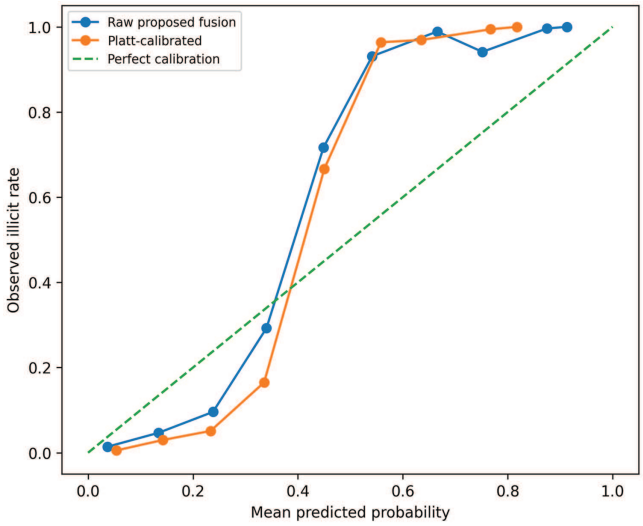
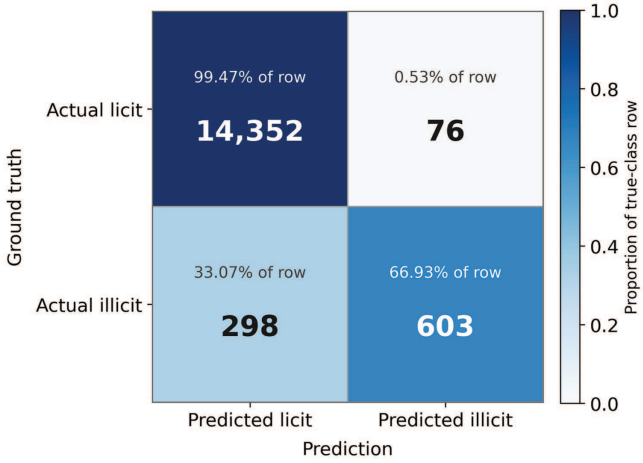


Figure 8
Confusion matrix for the proposed mean-of-five fusion at the validation-selected operating point. The upper-left cell is the true-negative count (14,352); TN (true negative), FP (false positive), FN (false negative), and TP (true positive). labels are shown explicitly in the revised figure



cell represents 14,352 true negatives (licit transactions correctly predicted as licit), while the other cells show 76 false positives, 298 false negatives, and 603 true positives. Cell shading reflects count magnitude only and does not indicate missing or masked data. This trade-off may be appropriate for analyst triage but not automatically for transaction rejection. A prevention layer should therefore combine the model probability with transaction value, customer risk, authentication status, and policy-specific consequences.

6.8. Preventive cybersecurity integration, explainability, and privacy

The model can be integrated before settlement as a tiered decision service. A low-risk tier can continue without intervention; a medium-risk tier can trigger enhanced authentication, lower transaction limits, or multi-signature confirmation; and a high-risk tier can be delayed for manual investigation or temporarily suspended when policy permits. This design converts a detection score into a preventive control while retaining human oversight for high-impact decisions.

Actionability requires explanations that reference observable evidence. The Star-GAT branch can report influential input/output wallet mentions and their attention contributions, while the native branch can expose transaction-feature importance. These signals should be treated as investigative aids rather than proofs of causality. Future work can incorporate post hoc GNN explanation methods [27] and evaluate explanation fidelity and stability explicitly.

Secure deployment should also protect transaction intelligence and the detection infrastructure. Federated learning and secure aggregation could support cross-institution learning without centralizing all sensitive data, while differential privacy and access controls can reduce disclosure risks. Robustness testing should include poisoning, feature/edge manipulation, inference attacks, mixers, transaction splitting, synthetic addresses, and rapidly changing multi-hop laundering strategies [28, 29].

6.9. Limitations

Several limitations constrain interpretation. First, the final fusion weight assigns 99% of the logit contribution to ExtraTrees and 1% to Star-GAT, so the measured graph contribution is statistically consistent but small. Second, the Star-GAT is a local same-time relational model rather than a full continuous-time dynamic GNN. Third, no adversarial attack benchmark, prevented-loss simulation, end-to-end exchange latency, or streaming deployment was executed in this revision. Fourth, Elliptic++ labels are incomplete and historical; unknown transactions cannot be assumed licit, and test-time prevalence varies markedly across time steps. Finally, calibration did not remain stable after Platt scaling across temporal shifts. These limitations motivate continued work rather than claims of universal or state-of-the-art superiority.

7. Conclusion

This study developed and evaluated a leakage-controlled temporal graph-tabular residual fusion framework for Bitcoin transaction fraud detection. The framework combined a strong native ExtraTrees branch with a role-aware Temporal Star-GAT that aggregated same-time input/output wallet context, and it selected the fusion weight and decision threshold using validation data only. On the locked Elliptic++ test interval, the proposed model achieved 91.864 ± 0.130% ROC-AUC, 75.636 ± 0.081% PR-AUC, 76.497 ± 0.697% F1, and a 0.477 ± 0.127% FPR across five matched seeds. The graph branch did not outperform the native model on its own; instead, it provided a small, consistent residual improvement. This more modest result is supported by paired bootstrap analysis, Top-K ranking, temporal robustness, calibration diagnostics, and a confusion-matrix operating point.

Future work should move from offline detection toward evaluated prevention. Priority directions include continuous-time or heterogeneous temporal GNNs, continual learning and drift alarms, adversarial training, threat-intelligence integration, federated/privacy-preserving collaboration, and explanation-fidelity testing. Operational studies should measure prevented financial loss, detection lead time, false-blocking rate, analyst

workload, recovery effectiveness, and end-to-end decision latency under live or replayed streaming conditions. Such evaluation would establish whether small relational gains translate into meaningful security improvements in real cryptocurrency systems.

Acknowledgment

The authors thank the developers and maintainers of the publicly available Elliptic++ dataset for enabling reproducible research on cryptocurrency transaction and wallet graphs.

Ethical Statement

This study used a publicly available, de-identified blockchain benchmark and did not involve human participants, clinical data, or animal subjects.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The Elliptic++ dataset used in this study is publicly available through the dataset repository described by Elmougy and Liu [22]. This article is accompanied by a reproducibility package containing the analysis scripts, configuration, five-seed result files, statistical tests, diagnostic outputs, and figure-generation artifacts required to verify the reported results.

Author Contribution Statement

Mohamed M. Abbassy: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization, Supervision, Project administration. **Amr E. Elshora:** Methodology, Validation, Resources, Writing – review & editing. **Ayman Abo-Alnadr:** Validation, Formal analysis, Writing – review & editing.

References

[1] Abufardeh, S. O. (2025). Cryptocurrency market security challenges and solutions. In *The Cryptocurrency Market: Investigating Trust, Technology, and Regulations* (pp. 157–181). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-90542-1_8

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

[3] Cheng, D., Zou, Y., Xiang, S., & Jiang, C. (2025). Graph neural networks for financial fraud detection: A review. *Frontiers of Computer Science*, *19*(9), 199609. https://doi.org/10.1007/s11704-024-40474-y

[4] Xiong, A., Tong, Y., Jiang, C., Guo, S., Shao, S., Huang, J., . . . , & Qi, B. (2024). Ethereum phishing detection based on graph neural networks. *IET Blockchain*, *4*(3), 226–234. https://doi.org/10.1049/blc2.12031

[5] Liu, Y., Shen, J., Fang, D., & Xu, H. (2025). Fraud detection on multi-relational graphs via semantic extraction and topological enhancement. In *Advanced Intelligent Computing Technology and Applications* (pp. 433–444). Springer. https://doi.org/10.1007/978-981-96-9946-9_37

[6] Chen, Z., Liu, S.-Z., Huang, J., Xiu, Y.-H., Zhang, H., & Long, H.-X. (2024). Ethereum phishing scam detection based on data augmentation method and hybrid graph neural network model. *Sensors*, *24*(12), 4022. https://doi.org/10.3390/s24124022

[7] Wang, L., Xu, M., & Cheng, H. (2023). Phishing scams detection via temporal graph attention network in Ethereum. *Information Processing & Management*, *60*(4), 103412. https://doi.org/10.1016/j.ipm.2023.103412

[8] Lin, Z., Xiao, X., Hu, G., Li, Q., Zhang, B., & Luo, X. (2023). Tracking phishing on Ethereum: Transaction network embedding approach for accounts representation learning. *Computers & Security*, *135*, 103479. https://doi.org/10.1016/j.cose.2023.103479

[9] Wang, X., Guo, J., Luo, X., & Yu, H. (2024). DyHDGE: Dynamic heterogeneous transaction graph embedding for safety-centric fraud detection in financial scenarios. *Journal of Safety Science and Resilience*, *5*(4), 486–497. https://doi.org/10.1016/j.jnlssr.2024.05.005

[10] Tong, G., & Shen, J. (2023). Financial transaction fraud detector based on imbalance learning and graph neural network. *Applied Soft Computing*, *149*, 110984. https://doi.org/10.1016/j.asoc.2023.110984

[11] Marasi, S., & Ferretti, S. (2024). Anti-money laundering in cryptocurrencies through graph neural networks: A comparative study. In *2024 IEEE 21st Consumer Communications & Networking Conference*, 272–277. https://doi.org/10.1109/CCNC51664.2024.10454631

[12] Zhang, J., Sui, H., Sun, X., Ge, C., Zhou, L., & Susilo, W. (2024). GrabPhisher: Phishing scams detection in Ethereum via temporally evolving GNNs. *IEEE Transactions on Services Computing*, *17*(6), 3727–3741. https://doi.org/10.1109/TSC.2024.3411449

[13] Wu, C., Chen, J., Zhao, Z., He, K., Xu, G., Wu, Y., . . . , & Xiang, Y. (2024). TokenScout: Early detection of Ethereum scam tokens via temporal graph learning. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 956–970. https://doi.org/10.1145/3658644.3690234

[14] Wu, L., Cui, P., Pei, J., Zhao, L., & Guo, X. (2022). Graph neural networks: Foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4840–4841. https://doi.org/10.1145/3534678.3542609

[15] Waikhom, L., & Patgiri, R. (2023). A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. *Artificial Intelligence Review*, *56*(7), 6295–6364. https://doi.org/10.1007/s10462-022-10321-2

[16] Dwivedi, V. P., Kanatsoulis, C., Huang, S., & Leskovec, J. (2025). Relational deep learning: Challenges, foundations and next-generation architectures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5999–6009. https://doi.org/10.1145/3711896.3736558

[17] Wu, B., Chao, K.-M., & Li, Y. (2024). Heterogeneous graph neural networks for fraud detection and explanation in supply chain finance. *Information Systems*, *121*, 102335. https://doi.org/10.1016/j.is.2023.102335

[18] Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery*

and Data Mining, 855–864. https://doi.org/10.1145/2939672.2939754

[19] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

[20] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

[21] Li, Y., Zhou, J., Verma, S., & Chen, F. (2022). *A survey of explainable graph neural networks: Taxonomy and evaluation metrics*, arXiv, https://doi.org/10.48550/arXiv.2207.12599

[22] Elmougy, Y., & Liu, L. (2023). Demystifying fraudulent transactions and illicit nodes in the Bitcoin network for financial forensics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3979–3990. https://doi.org/10.1145/3580305.3599803

[23] Liu, X., Yan, M., Song, S., Lv, Z., Li, W., Sun, G., . . . , & Fan, D. (2023). GNNSampler: Bridging the gap between sampling algorithms of GNN and hardware. In *Machine Learning and Knowledge Discovery in Databases* (pp. 498–514). Springer. https://doi.org/10.1007/978-3-031-26419-1_30

[24] Shao, Y., Li, H., Gu, X., Yin, H., Li, Y., Miao, X., . . . , & Chen, L. (2024). Distributed graph neural network training: A survey. *ACM Computing Surveys*, *56*(8), 1–39. https://doi.org/10.1145/3648358

[25] Sun, L., Dou, Y., Yang, C., Zhang, K., Wang, J., Yu, P. S., . . . , & Li, B. (2023). Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *35*(8), 7693–7711. https://doi.org/10.1109/TKDE.2022.3201243

[26] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

[27] Yuan, H., Yu, H., Gui, S., & Ji, S. (2023). Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(5), 5782–5799. https://doi.org/10.1109/TPAMI.2022.3204236

[28] Hou, Z., Lin, M., Torkamani, M. A., Wang, S., & Liu, X. (2024). Adversarial robustness in graph neural networks: Recent advances and new frontier. In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*. https://doi.org/10.1109/DSAA61799.2024.10722771

[29] Guan, F., Zhu, T., Zhou, W., & Choo, K.-K. R. (2024). Graph neural networks: A survey on the links between privacy and security. *Artificial Intelligence Review*, *57*, 40. https://doi.org/10.1007/s10462-023-10656-4

[30] Rao, S. X., Zhang, S., Han, Z., Zhang, Z., Min, W., Chen, Z., . . . , & Zhang, C. (2021). xFraud: Explainable fraud transaction detection. *Proceedings of the VLDB Endowment*, *15*(3), 427–436. https://doi.org/10.14778/3494124.3494128

How to Cite: Abbassy, M. M., Elshora, A. E., & Abo-Alnadr, A. (2026). Graph Context as a Residual Correction: Leakage-Controlled Graph-Tabular Fusion for Illicit Bitcoin Transaction Detection. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS620211464