# MULTI-EDGE INTRA-GROUP GRAPH CONSTRUCTION FOR CREDIT CARD FRAUD DETECTION

Roya Amiri
School of Computer Science and Engineering
University of Sunderland
Sunderland, SR6 0DDUnited Kingdom
roya.amiri@students.sunderland.ac.uk

Sardar Jaf
School of Computer Science and Engineering
University of Sunderland
Sunderland, SR6 0DD, United Kingdom
sardar.jaf@sunderland.ac.uk

***Abstract*—As financial systems grow more complex and interconnected, traditional fraud detection methods struggle to keep pace with increasingly sophisticated attacks. Graph-based approaches have been explored with a focus on cross-user interactions. In this paper, we propose a graph-based approach that focuses on individual cardholder behaviour. Each cardholder is modelled as an isolated graph capturing personal spending patterns. By leveraging advances in Graph Neural Networks (GNNs), we adopt an intra-group graph formulation where edges are restricted within a single cardholder's transaction history. We construct three complementary edge types: temporal edges linking sequential transactions, similarity edges connecting behaviourally similar transactions, and merchant-based edges capturing repeated interactions with the same merchant. To isolate the effect of graph construction, we use a controlled experimental setup with fixed model architecture, training procedure, and evaluation protocol. We test the proposed model on two public datasets: the Sparkov and the IBM credit card datasets. We find strong model performance when transaction histories are dense. On Sparkov dataset, the model achieves 0.909 (F1-score) and 0.992 (AUC), substantially outperforming prior published results on the same dataset. On IBM, where transaction histories are sparse, the model achieves 0.763 (F1-score) and 0.962 (AUC), which highlights the importance of graph connectivity.**

***Keywords: Graph Neural Networks, Credit Card Fraud Detection, Graph Attention Networks, Intra-Group Graph Construction, Transaction-Level Classification, GATv2, Class Imbalance***

## *I.* INTRODUCTION

As digital payment volumes increase, so does the sophistication of fraudulent schemes, placing pressure on detection systems to keep pace. Traditional approaches, including rule-based systems and machine learning models, exhibit significant limitations when confronted with evolving fraud tactics [1]. Recent developments in deep learning have improved fraud detection model performance [2]. However, many approaches still treat transactions in relative isolation, struggling to fully capture the complex relational structures underlying fraudulent behaviour [3].

In credit card fraud detection, GNN-based approaches typically construct graphs by linking transactions from different cardholders using shared attributes. While these cross-entity edge connections enable the model to detect fraud patterns that span multiple transactions, they also mix the behavioural contexts across cardholders, potentially diluting the per-account anomaly signals that are central to detecting compromised accounts. To address this limitation, this paper proposes an alternative approach based on intra-group graph construction, where all edges are restricted within individual cardholder histories. Under this intra-group formulation, each cardholder's activities represent an isolated graph capturing their own behavioural patterns. We construct multiple complementary edge types to model different aspects of transaction behaviour, including temporal dependencies, similarity relationships, and repeated merchant interactions. This design preserves individual behavioural context while enabling rich relational modelling within each account.

We evaluate the proposed model on dense and sparse transaction history from Sparkov and IBM datasets, respectively. The model achieves 0.909 F1-score on the Sparkov dataset and a 0.763 F1-score on the IBM dataset, thus revealing a critical dependency on per-cardholder transaction density.

In this paper, we make four contributions:

1) We introduce a multi-edge intra-group graph construction method for fraud detection using Graph Attention Network v2 (GATv2).
2) We identify graph density as a key factor governing GNN effectiveness, demonstrating that performance degrades when per-group connectivity becomes too sparse.
3) We show through empirical comparison that intra-group graphs can match cross-entity approaches under favourable data conditions, as demonstrated on the Sparkov dataset, while we identify reduced performance on the IBM dataset due to sparsity.
4) We design a leakage-safe experimental pipeline that enforces three safeguards: group-stratified splitting prevents cardholder overlap between training and evaluation sets, temporal ordering ensures no future transactions inform past predictions, and feature normalisation is fitted on training data only.

The rest of this paper is organised as follows. Section II reviews related work on graph-based approaches to fraud detection, covering graph construction strategies, attention mechanisms, and evaluation practices. Section III describes our proposed methodology, including the intra-group graph construction, edge type formulation, and GATv2 model architecture. Section IV outlines the experimental design, detailing the datasets, evaluation metrics, and training protocol. Section V presents the results, including graph topology analysis, test set performance, confusion matrix analysis, and comparison with prior work. Section VI discusses the key findings, error analysis, and limitations of the approach. Section VII concludes the paper and outlines directions for future work.

## II. Related Work

GNN-based fraud detection has advanced rapidly, but existing approaches differ substantially in how they construct graphs and model relational structure. We organise prior work along three axes: graph construction strategy, attention and temporal modelling, and the treatment of class imbalance and evaluation rigour.

The dominant paradigm connects transactions from different cardholders through shared attributes such as merchants, geographic proximity, or device identifiers. In [4], a weighted multi relational transaction graph in which edges are defined. by shared IP addresses, MAC addresses, and temporal proximity, with each edge assigned a weight reflecting the strength of the relationship. Building on GraphSAGE, [4] introduces similarity-based neighbour sampling and a temporal attention mechanism, achieving improved F1-score and AUC on a real-world dataset. However, their approach tightly couples graph construction with model design and relies on manually defined logical rules for edge creation, making it difficult to isolate the contribution of graph topology from that of the model architecture.

In [5], NeuraShield, which models transactions as relationships between cardholders and merchants using an Edge Graph Convolutional Network (Edge-GCN). Evaluated on the IBM dataset under varying class-imbalance settings, Edge-GCN achieves an F1-score of 0.71 and AUC of 0.87, outperforming GCN, GraphSAGE, R-GCN, and MLP baselines. Similarly, [6] construct a heterogeneous cardholder–merchant graph using Graph Attention Networks, reporting an F1-score of 0.854 on a large, simulated dataset, though the study acknowledges limited incorporation of temporal dynamics. Both [5] and [6] demonstrate the value of heterogeneous cross-entity structure but do not explore whether comparable performance can be achieved without cross-entity edges.

Attention-based architectures have been increasingly adopted to weight the importance of neighbouring nodes during message passing. In [7], applies Graph Attention Networks to healthcare fraud detection, showing that adaptive attention weights improve identification of collusive behaviour in highly interconnected provider–patient–claim networks. While effective, the added computational cost and limited interpretability of the attention mechanism remain practical concerns. In [8], address temporal modelling more directly with their Augmented Temporal-Aware Graph Attention Network (ATGAT) for cryptocurrency fraud detection. ATGAT integrates multi-scale temporal embeddings with a triple attention mechanism spanning structural, temporal, and global context, achieving an AUC of 0.913 on the Elliptic++ dataset. The work demonstrates the importance of temporal dynamics in transaction graphs but introduces considerable model complexity.

In [9], integrates R-GCN into a verification pipeline, reporting 99.88% accuracy on IBM, though the metric is uninformative at 0.12% fraud rate. In [10], uses TigerGraph with heuristic thresholds on Sparkov, achieving 0.773 F1-score.

Across these studies, graph construction consistently links transactions across different cardholders through shared entities. This cross-entity connectivity enables the detection of patterns spanning multiple accounts but also mixes the behavioural contexts of different cardholders, potentially diluting per-account anomaly signals. To the best of our knowledge, the use of multi-edge intra-group graphs where all edges are scoped entirely within individual cardholder histories has not been systematically evaluated. In particular, the role of per-group graph density in governing GNN effectiveness under intra-group construction has not been studied.

## III. Proposed Method

### A. Problem Formulation

We formulate credit card fraud detection as a node-level binary classification problem on a homogeneous graph. Each transaction is considered as a node $v_i$ with feature vector $x_i$ consisting of categorical embeddings and scaled numerical features. The label $y_i$ in {0, 1} indicates whether the transaction is fraudulent. The graph $G = (V, E)$ is constructed such that all edges are scoped within individual cardholder groups, producing a set of disconnected subgraphs $\{G_1, G_2, \ldots, G_K\}$ where K is the number of unique cardholders.

### B. Graph Construction

Given a cardholder group with transactions ordered by time, we construct three edge types:

**Temporal sequential chains:** Each transaction $v_i$ is connected to its $k^t$ previous transactions $\{v_{i\_1}, v_{i\_2}, \ldots, v_{i\_kt}\}$ within the same cardholder, where $k_t = 3$ (temporal lookback depth). These edges encode the chronological spending sequence. Formally, $E_t = \{(v_i, v_j)$ : same_group $(v_i, v_j)$ AND $|rank(v_i) - rank(v_j)| <= k_t\}$, where rank denotes the temporal position within the group. The temporal lookback $k_t = 3$ is selected to capture short-term behavioural dependencies while avoiding overly dense connections that may dilute sequential patterns.

**Cosine similarity edges:** Within each cardholder group, pairwise cosine similarity is computed between all transaction feature vectors. Each transaction connects to its $k_s=5$ most similar neighbours above a threshold $tau=0.5$. Formally, $E_s = \{(v_i, v_j)$: same_group $(v_i, v_j)$ AND $cos(x_i, x_j) > tau$ AND $v_j$ in $top\_k_s(v_i)\}$. These edges cluster behaviourally similar transactions, so a fraudulent transaction with atypical features will have fewer and weaker similarity connections. Cosine similarity is chosen due to its scale-invariant property, making it well-suited for comparing transaction feature vectors with mixed numerical ranges. The choice of $k_s = 5$ balances connectivity and noise, ensuring that each transaction is linked to a small set of highly relevant neighbours without introducing excessive spurious edges.

**Merchant sub-relation chains:** Within each cardholder, transactions at the same merchant are sorted by time and connected as sequential chains. Formally, $E_m = \{(v_i, v_{i+1})$: same_group $(v_i, v_{i+1})$ AND same_merchant $(v_i, v_{i+1})$ AND consecutive_at_merchant $(v_i, v_{i+1})\}$. These edges capture repeat-visit loyalty patterns. A fraudulent transaction at a never-visited merchant will have zero merchant connections.

The final edge set is $E = E_t \cup E_s \cup E_m \cup E_{self}$, where $E_{self}$ adds self-loops for every node. All edges are bidirectional. No edges cross cardholder boundaries. Groups larger than 200 transactions are randomly subsampled to prevent $O(n^2)$ similarity computation on very active accounts.

**Temporal edges** encode spending rhythm sudden bursts of transactions within a short window produce structural signals that the attention mechanism can learn to weight.

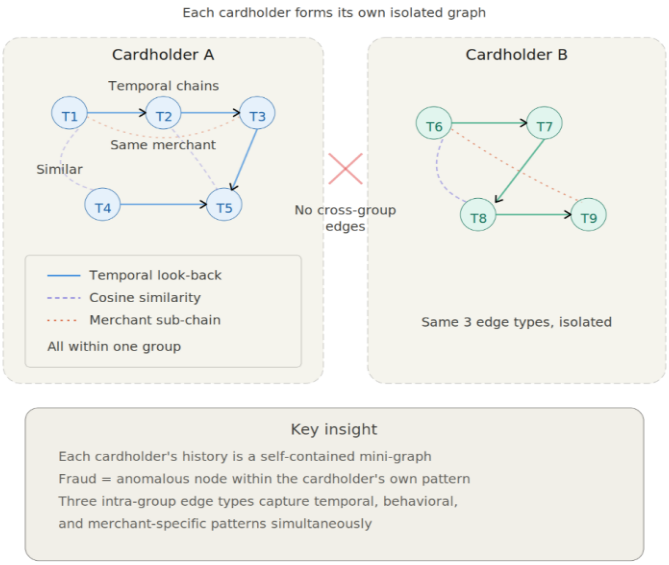


Figure 1: Intra-group graph construction. Each cardholder forms an isolated graph where nodes represent transactions. Temporal edges (blue) connect sequential transactions, similarity edges (purple) link behaviourally similar transactions, and merchant edges (coral) connect repeated visits. No edges exist between different cardholders.

**Similarity edges** capture behavioural profiling a fraudulent transaction with an unusual amount or unfamiliar category will have few or weak connections to the cardholder's typical spending cluster, even if it occurs at a normal time.

**Merchant sub-chains** encode location loyalty a fraudulent transaction at a never-visited merchant will have zero merchant connections, while legitimate repeat visits provide normalising context that reduces false positive risk. The three types are complementary: The full model $E = E_t \cup E_s \cup E_m$ combines all three dimensions to provide the richest possible neighbourhood structure within each cardholder's subgraph. Figure 1 illustrates this construction, showing two cardholders as isolated subgraphs with all three edge types operating within each group and no cross-group connections.

### C. Computational Complexity

Temporal and merchant edges are constructed in linear time per cardholder group, while similarity edges require $O(n^2)$ pairwise comparisons. To ensure scalability, we restrict similarity edges to top-$\boldsymbol{k_s}$ neighbours and subsample groups exceeding 200 transactions. This keeps graph construction tractable while preserving the most informative relationships.

### D. GATv2 Model Architecture

We use a two-layer GATv2 [11]. which computes dynamic attention coefficients that depend on both source and target node features, allowing the attention function to assign different neighbour importance depending on the query node. This expressiveness is particularly relevant for our formulation, where temporal neighbours, similarity neighbours, and merchant-chain neighbours coexist within the same graph and require different attention weights depending on their relevance to fraud detection. Each layer applies GATv2Conv with 32 hidden units and 2 attention heads, producing a 64-dimensional node representation. LayerNorm and LeakyReLU activation are applied between layers, followed by dropout ($p = 0.30$). A two-layer MLP classifier (Linear 64→128, LeakyReLU, Dropout, Linear 128→1) maps node embeddings to a single logit per transaction. Categorical features are encoded via learned embedding tables (dimension 8 per column) and concatenated with StandardScaler-normalised numerical features. The model is trained using BCEWithLogitsLoss with automatically computed positive class weight.

### E. Training and Evaluation Protocol

**Data splitting:** Each dataset used in this study is partitioned into a training set (80%) and a held-out test set (20%) using a group-aware holdout split based on user identity, ensuring that all transactions from a given user appear exclusively in either the training or test set, but never both. To maintain class distribution, users are approximately stratified according to their group-level fraud prevalence prior to splitting. Within the training set, group-stratified 5-fold cross-validation is applied to generate train/validation splits. The held-out test set remains completely unseen and is not used at any stage of model training, validation, or threshold tuning.

**Training:** Models are trained with AdamW and CosineAnnealingWarmRestarts. Gradient clipping is applied at norm 1.0. Early stopping monitors validation Average Precision (AP) every 5 epochs with patience of 12 checks.

**Threshold selection:** After training, the classification threshold is tuned to maximise the F1-score for the positive (fraudulent) class on pooled validation predictions across all 5 folds.

**Test evaluation:** Predicted probabilities from all 5-fold models are averaged (ensemble) and evaluated on the held-out TEST set using the pooled validation threshold. All reported performance metrics are from this held-out test set only.

## IV. Experimental Setup

### A. Datasets

Table 1 summarises both datasets. The Sparkov dataset [13] contains 1.85M transactions at ~0.1% fraud rate. After down sampling non-fraud to achieve an 18.4% fraud rate, 52,394 transactions remain with 27 engineered features including cyclic hour encodings, customer-merchant distance, amount relative to cardholder average, and inter-transaction time differences. The IBM dataset [14] contains 24.4M synthetic transactions at 0.09% fraud rate. Smart down sampling retains all 29,757 fraud cases with a 1:10 non-fraud ratio, producing 327,327 transactions at 9.09% fraud rate with 23 features including user-spending normalisation (fitted on training data only to prevent leakage).

### B. Evaluation Metrics

Even after downsampling, fraud remains a minority class (18.42% in Sparkov, 9.09% in IBM), making standard accuracy uninformative --a naive all-legitimate classifier would achieve 81.58% and 90.91% accuracy while detecting zero fraud. We therefore prioritise metrics sensitive to minority-class performance. Binary F1-score, Average Precision (AP), Precision, Recall, AUC-ROC, and Brier Score. To prevent the model from converging to a majority-class solution, BCEWithLogitsLoss is trained with automatically computed positive class weight (negative/positive ratio), upweighting the fraud class.

## V. RESULTS

### A. Graph Topology

Table 2 highlights a clear contrast in graph structure between the two datasets. The intra-group method produces moderately dense graphs on Sparkov (average degree 12.0), enabling effective neighbourhood aggregation and information flow across transactions. In contrast, the IBM dataset results in extremely sparse graphs (average degree 1.1), where most nodes are connected only through self-loops. This structural difference significantly limits the ability of the GNN to propagate information, reducing its advantage over non-graph models.

### B. Test Set Performance

Table 3 reports the held-out test set performance on both datasets. On Sparkov, the model achieves an F1-score of 0.909 with high precision (0.943) and recall (0.877), indicating effective fraud detection with minimal false positives. The low Brier score (0.027) confirms well-calibrated probability estimates. On IBM, performance is lower (F1-score of 0.763, AUC of 0.962), consistent with the sparse graph structure identified in Section V-A. To assess result stability, Table 4 reports our cross-validation metrics with standard deviations across the 5-fold validation.

On Sparkov, the extremely low variance (F1 std = 0.001) confirms that the reported performance is stable and not an artefact of a favourable split. On IBM, higher variance (F1 std = 0.019) reflects sensitivity to which cardholders appear in each fold, consistent with the sparse graph structure where small changes in group composition can substantially affect connectivity (Table 4).

### C. Edge-Type Ablation

To assess the individual contribution of each edge type, we conduct an ablation in which the intra-group graph is built using only one edge family at a time temporal, similarity, or merchant holding architecture, training protocol, splits, and threshold tuning fixed. Table 5 reports held-out test performance for each variant alongside the full model. On Sparkov, temporal edges alone achieve the strongest performance (F1 = 0.920, AUC = 0.995), with similarity edges performing poorly in isolation (F1 = 0.522) and merchant edges providing intermediate performance (F1 = 0.794); the full model (F1 = 0.909, AUC = 0.992) remains competitive while offering more stable calibration across folds. On IBM, all four variants perform within a narrow band (F1 = 0.764–0.768 for single edge types, 0.763 for the full model), indicating that under extreme sparsity (average degree ≈ 1.1), edge-type selection has little effect on performance and graph connectivity itself is the binding constraint. These results suggest that the relative contribution of each edge type is density-dependent rather than uniformly complementary across datasets.

### D. Confusion Matrix Analysis

Table 6 presents the row-normalised confusion matrices, complemented by ROC, Precision–Recall, and threshold tuning curves. Both datasets achieve high accuracy for legitimate transactions (Class 0), with 98.81% on Sparkov and 97.80% on IBM, indicating effective modelling of normal behaviour. However, performance diverges on the fraud class (Class 1). Sparkov detects 87.73% of fraud cases with a 1.19% false positive rate (FPR), while IBM detects 75.61% with a higher FPR of 2.20%. This disparity is attributed to graph sparsity in IBM (average degree ≈ 1.1), which limits neighbourhood information and forces greater reliance on node features, negatively impacting minority-class detection.

Diagnostic curves support these findings. Figures 2 and 3 present the evaluation results of the intra-group model across two datasets. On the IBM dataset, the model achieves an AUC of 0.962 at a decision threshold of 0.90, correctly identifying 4,249 fraudulent transactions while maintaining a low false positive rate of 1,086. On the Sparkov dataset, the model performs notably stronger, achieving an AUC of 0.992 at a threshold of 0.86, with only 102 false positives and 236 missed fraud cases out of 10,485 total test samples. The precision-recall curves confirm that high precision is sustained across a broad recall range on both datasets, with performance degrading only at very high recall values.

TABLE 1: DATASET STATISTICS AFTER PREPROCESSING AND GROUP-STRATIFIED SPLIT.

| Property | Sparkov | IBM |
|---|---|---|
| **Raw transactions** | 1,852,394 | 24,386,834 |
| **After down sampling** | 52,394 | 327,327 |
| **Fraud rate (post)** | 18.42% | 9.09% |
| **Engineered features** | 27 | 23 |
| **Cardholder group key** | cc_num | card |
| **Temporal resolution** | Full datetime | Hour: Minute only |
| **Geolocation available** | Yes (lat/long) | No |
| **DEV / TEST split** | 41,909 / 10,485 | 263,478 / 63,849 |
| **DEV fraud rate** | 18.44% | 9.14% |
| **TEST fraud rate** | 18.34% | 8.90% |

TABLE 2: GRAPH TOPOLOGY STATISTICS (FOLD 1 TRAINING SET).

| Dataset | Nodes | Edges | Avg Deg | Deg (min-max) |
|---|---|---|---|---|
| Sparkov | 33,815 | 407,381 | 12.0 | 6 - 18 |
| IBM | 211,649 | 223,583 | 1.1 | 1 - 20 |

TABLE3: HELD-OUT TEST SET RESULTS (5-FOLD ENSEMBLE).

| Metric | Sparkov | IBM |
|---|---|---|
| **F1** | 0.909 | 0.763 |
| **AUC** | 0.992 | 0.962 |
| **Brier** | 0.027 | 0.069 |

TABLE4: CROSS VALIDATION PERFORMANCE SUMMARY. (MEAN ± STANDARD DEVIATION ACROSS 5 FOLDS).

| Metric | Sparkov (mean ± std) | IBM (mean ± std) |
|---|---|---|
| **F1** | 0.913 ± 0.001 | 0.743 ± 0.019 |
| **Precision** | 0.926 ± 0.013 | 0.731 ± 0.032 |
| **Recall** | 0.900 ± 0.015 | 0.755 ± 0.013 |
| **AUC** | 0.988 ± 0.002 | 0.953 ± 0.005 |
| **AP** | 0.968 ± 0.003 | 0.793 ± 0.019 |

TABLE 5: EDGE-TYPE ABLATION

| Variant | Sparkov F1 | Sparkov AUC | IBM F1 | IBM AUC |
|---|---|---|---|---|
| **Temporal only** | 0.920 | 0.995 | 0.764 | 0.962 |
| **Similarity only** | 0.522 | 0.818 | 0.767 | 0.962 |
| **Merchant only** | 0.794 | 0.961 | 0.768 | 0.963 |
| **Full model (ours)** | 0.909 | 0.992 | 0.763 | 0.962 |

TABLE 6: CONFUSION MATRICES (ROW-NORMALISED, % OF TRUE CLASS)

| | Sparkov | | IBM | |
|---|---|---|---|---|
| | **Pred Legit** | **Pred Fraud** | **Pred Legit** | **Pred Fraud** |
| **True Legit** | 98.81% | 1.19% | 97.80% | 2.20% |
| **True Fraud** | 12.27% | 87.73% | 24.39% | 75.61% |

The threshold tuning plots indicate that the selected thresholds represent a well-balanced operating point between precision and recall, as reflected by the peak F1 scores. The superior performance on Sparkov compared to IBM is consistent with the denser and more structured graph topology of the Sparkov dataset, which provides richer neighbourhood information for the GNN to exploit.

Precision–Recall curves further emphasise this gap: Sparkov maintains precision above 0.95 up to ~0.85 recall, whereas IBM drops below 0.80 by ~0.60 recall, indicating increased false positives. Threshold analysis reveals that Sparkov has a broad optimal range (F1 stable between ~0.70–0.90), while IBM exhibits a narrow optimum around 0.91, reflecting lower robustness.

### *E. Comparison with Prior Work*

Table 7 compares our method against published results on the same datasets. On Sparkov, it improves F1 by 17.6% over TigerGraph (0.909 vs 0.773). On IBM, it outperforms NeuraShield in F1 (0.763 vs 0.714, +6.9%) and precision (0.771 vs 0.642, +20.1%), though NeuraShield achieves higher recall (0.804 vs 0.756). The GAT+MLP approach outperforms our method on IBM (F1 = 0.854 vs 0.763) but uses a fundamentally different heterogeneous edge-level formulation with cross-entity edges. These comparisons should be interpreted with caution, as the methods differ not only in graph construction but also in preprocessing, evaluation protocol, and model formulation.

TigerGraph relies on heuristic thresholds and community detection rather than learned representations, making the comparison one of paradigm rather than architecture. NeuraShield and GAT+MLP use heterogeneous cross-entity graphs with different node and edge definitions, and their reported metrics may reflect different train/test splits and class imbalance handling. No prior work on either dataset enforces group-stratified splitting or controlled downsampling as applied here, which may affect reported performance in either direction. Despite these differences, a consistent pattern emerges: the proposed method achieves competitive or superior performance when per-cardholder graph density is sufficient (Sparkov). A fully controlled comparison would require re-implementing all methods under identical preprocessing and evaluation conditions, which we identify as an opportunity for future benchmarking work.

Downsampling substantially eases the task: at native fraud rates (~0.1% Sparkov, ~0.09% IBM), fraud is roughly 1,000x rarer than legitimate transactions, versus ~4:1 and ~10:1 after downsampling. This easier class balance likely inflates our F1 relative to studies evaluated at native imbalance and should be weighed when interpreting Table 7.

TABLE 7: COMPARISON WITH PRIOR WORK.

| Method | Dataset | F1 | P | R | AUC | Graph |
|---|---|---|---|---|---|---|
| TigerGraph [10] | Sparkov | 0.77 | 0.82 | 0.724 | - | Hetero |
| NeuraShield [5] | IBM | 0.71 | 0.64 | 0.804 | 0.87 | Hetero |
| GAT+MLP [9] | IBM | 0.85 | 0.86 | 0.84 | - | Hetero |
| **Ours (intra-grp GATv2)** | Sparkov | **0.91** | **0.94** | **0.88** | **0.99** | Homo |
| **Ours (intra-grp GATv2)** | IBM | 0.76 | 0.77 | 0.76 | **0.96** | Homo |

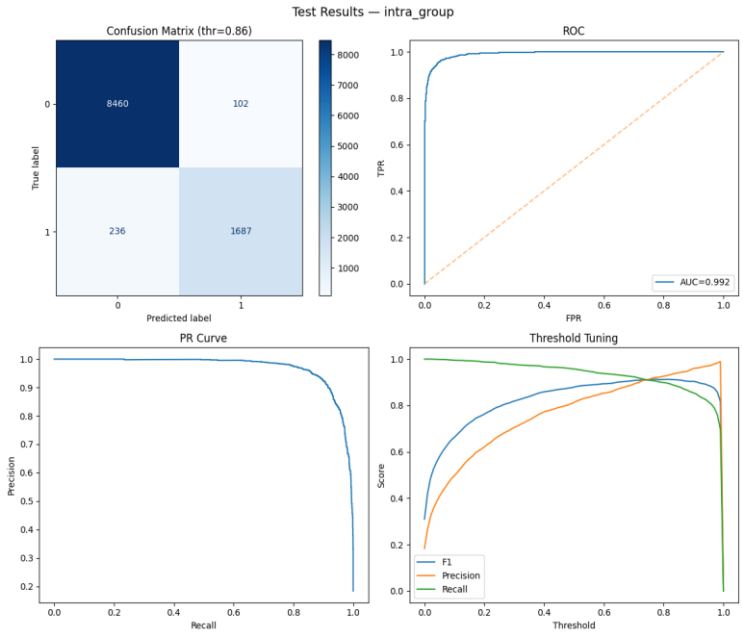


Fig. 2. Evaluation results of the intra-group GNN model on the Sparkov Credit Card Fraud dataset (threshold=0.86). The confusion matrix (TN=8,460, FP=102, FN=236, TP=1,687), ROC curve (AUC=0.992), precision-recall curve, and threshold tuning plot are shown.

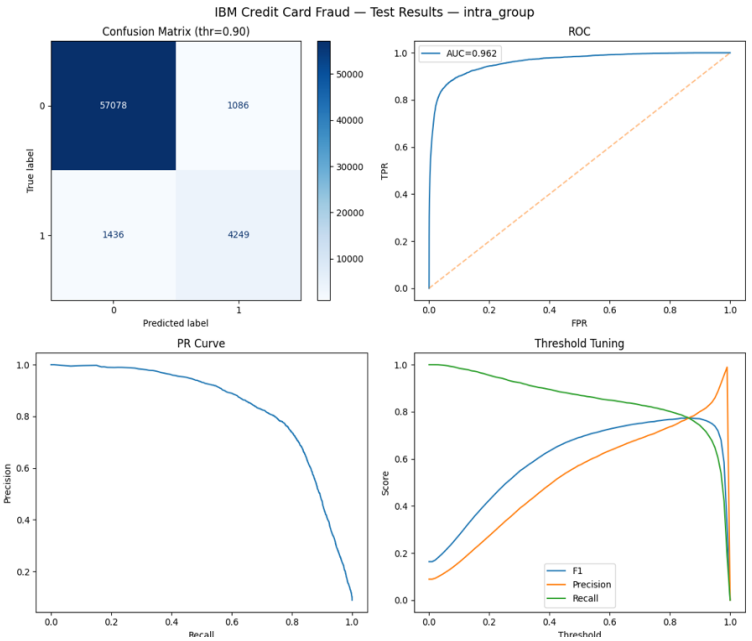


Fig. 3. Evaluation results of the intra-group GNN model on the IBM Credit Card Fraud dataset (threshold=0.90). The confusion matrix (TN=57,078, FP=1,086, FN=1,436, TP=4,249), ROC curve (AUC=0.962), precision-recall curve, and threshold tuning plot are shown.

## VI. Discussion and Error Analysis

### *A. The Density Threshold*

The central finding of this paper is the critical dependency of intra-group graph construction on per-cardholder transaction density. On Sparkov, cardholders average 12.0 edges per node (degree range 6–18), providing rich neighbourhood structure for all three edge types to contribute. On IBM, the average degree collapses to 1.1 (median 1.0), meaning most nodes have only self-loops. While this contrast suggests that graph density may be an

important factor in model performance, further validation across additional datasets with varying density levels would be needed to confirm this relationship. The model effectively degrades to an MLP with no neighbourhood aggregation. The downsampling required to make the 24.4M transaction IBM dataset computationally tractable retaining all fraud but only 1:10 non-fraud reduces most cardholders to one or two transactions, limiting the formation of temporal chains, similarity edges, and merchant connections. This presents a practical trade-off between computational feasibility and graph quality.

An edge-type ablation (Table 5, Section V-C) shows that the relative contribution of each edge type is density-dependent rather than uniformly complementary. On Sparkov, temporal edges alone achieve F1 = 0.920, narrowly exceeding the full model (F1 = 0.909), while similarity edges in isolation perform poorly (F1 = 0.522), suggesting they introduce some noise when combined with other edge types. The full model is retained as the primary configuration due to its higher AUC (0.992) and more stable calibration (Brier = 0.027) across folds. On IBM, all edge-type variants perform within a narrow band (F1 = 0.764–0.768), confirming that under extreme sparsity, graph connectivity rather than edge-type selection is the binding constraint on performance.

### *B. Error Analysis: False Negatives*

On Sparkov, the 236 false negatives represent transactions where the fraudulent purchase closely resembles the cardholder's legitimate behaviour. These are the hardest cases: a thief who mimics the victim's typical spending pattern (similar amounts, familiar merchants, normal hours) will produce a node with strong similarity edges and natural temporal context, making it indistinguishable from legitimate transactions within the intra-group graph alone. Detecting such well-camouflaged fraud may require additional signals beyond the cardholder's own transaction history, such as merchant-level risk scores or device fingerprinting, which fall outside the scope of the proposed graph construction.

### *C. Error Analysis: False Positives*

The 102 false positives on Sparkov (1.19% FPR) represent legitimate transactions that appear anomalous within their cardholder's history. Common patterns include first-time purchases at new merchants (no merchant sub-chain), unusually large purchases (low similarity to neighbours), and transactions during atypical hours (temporal discontinuity). These are cases where the cardholder's behaviour has legitimately changed, but the intra-group graph encodes this change as an anomaly. In practice, such false positives could be mitigated by incorporating contextual features (e.g., whether the merchant category is new for this cardholder) or by adjusting the classification threshold to favour precision.

## VI. Conclusion

We presented a cardholder-isolated graph construction method for transaction-level credit card fraud detection using GATv2. By scoping all edges within individual cardholder histories and combining temporal, similarity, and merchant sub-chain edges, the method captures per-account spending behaviour in a self-contained graph structure that the GATv2 attention mechanism can learn to classify at the transaction level. The method achieves F1 = 0.909 and AUC = 0.992 on the Sparkov dataset, substantially outperforming prior published results on the same data. The critical dependency on per-cardholder transaction density (F1 degrades to 0.763 on sparse IBM data) provides actionable deployment guidance: cardholder-isolated construction requires sufficient per-account history to form meaningful graph structure.

Our edge-type ablation further shows that this dependency extends to edge construction itself: on dense graphs (Sparkov), temporal edges drive most of the discriminative signal, while similarity and merchant edges contribute secondary, density-dependent value; on sparse graphs (IBM), edge-type choice has little effect, as connectivity itself becomes the binding constraint. This indicates that the value of multi-edge construction is not fixed, but conditional on sufficient per-cardholder transaction density.

Future work includes investigating adaptive group key selection (automatically choosing grouping columns based on density analysis), augmenting sparse cardholder graphs with density-aware edge generation techniques, and developing edge-type-aware attention mechanisms that could allow the model to learn the relative importance of temporal, similarity, and merchant edges directly, rather than treating them as a fixed union.

## REFERENCES


[1] A. Abdallah, M. A. Maarof, and A. Zainal, “Fraud detection system: A survey,” *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016, doi: 10.1016/j.jnca.2016.04.007.

[2] C. I. Amuche, “Multi-relation graph transformer and edge-aware learning: A scalable approach for detecting fraud in heterogeneous graphs,” in *Proc. Int. Conf. Softw. Eng. Comput. Sci. (CSECS)*, 2025.

[3] Y. Cui, X. Han, J. Chen, X. Zhang, J. Yang, and X. Zhang, “FraudGNN-RL: A graph neural network with reinforcement learning for adaptive financial fraud detection,” *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3281234.

[4] G. Liu, J. Tang, Y. Tian, and J. Wang, “Graph neural network for credit card fraud detection,” in *Proc. IEEE Int. Conf. Big Data*, 2019, doi: 10.1109/BigData47090.2019.9005999.

[5] P. Gunawardana, W. J. Samaraweera, and R. G. U. I. Meththananda, “NeuraShield: A GNN-based approach to credit card fraud detection,” in *Proc. Int. Res. Conf. Smart Comput. Syst. Eng. (SCSE)*, 2025.

[6] C. A. Kabwama , “Graph attention networks for credit card fraud detection: A relational learning approach,” *World J. Adv. Res. Rev.*, vol. 26, no. 3, pp. 2580–2585, 2025, doi: 10.30574/wjarr.2025.26.3.2400.

[7] S. Mardani and H. Moradi, “Using graph attention networks in healthcare provider fraud detection,” *IEEE Access*, vol. 12, pp. 132786–132800, Jul. 2024, doi: 10.1109/ACCESS.2024.3425892.

[8] Z. Zheng, B. Zhou, and Y. Song, “Temporal-aware graph attention network for cryptocurrency transaction fraud detection,” *arXiv preprint arXiv:2506.21382*, 2025, doi: 10.48550/arXiv.2506.21382.

[9] M. Huo, K. Lu, Q. Zhu, and Z. Chen, “Enhancing customer contact efficiency with graph neural networks in credit card fraud detection workflow,” in *Proc. IEEE 7th Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, 2025, pp. 320–324, doi: 10.1109/CISCE65916.2025.11065245.

[10] N. Mauliddiah and Suharjito, “Implementation graph database framework for credit card fraud detection,” *Procedia Comput. Sci.*, vol. 227, pp. 326–335, 2023, doi: 10.1016/j.procs.2023.10.531.

[11] S. Brody, U. Alon, and E. Yahav, “How attentive are graph attention networks?” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[12] Sparkov, “Simulated credit card transactions,” 2020. [Online]. Available: https://www.kaggle.com/datasets/kartik2112/fraud-detection

[13] IBM, “IBM AML transaction dataset,” 2019. [Online]. Available: https://www.kaggle.com/datasets/ealtman2019/ibm-transactions-for-anti-money-laundering-aml

[14] S. Motie and M. Sadeghian, “Financial fraud detection using graph neural networks: A systematic review,” *Expert Syst. Appl.*, 2024.

[15] S. Priyadarshi, “Financial fraud detection using graph neural networks,” in *Proc. World Conf. Appl. Intell. Comput. (AIC)*, 2025.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.