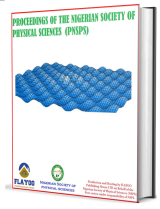
Published by Nigerian Society of Physical Sciences. Hosted by FLAYOO Publishing House LTD

Proceedings of the Nigerian Society of Physical Sciences

Journal Homepage: https://flayoophl.com/journals/index.php/pnspsc



Fraud detection in Nigerian financial card transactions using a graph attention network

Oladayo Tosin Akinwande[a,*], Sulaimon Adebayo Bashir[b], Opeyemi Aderiike Abisoye[b], Solomon Adelowo Adepoju[b]

[a]*Department of Software Engineering, Veritas University, Bwari-Abuja, Nigeria*
[b]*Department of Computer Science, Federal University of Technology, P.M.B. 65, Minna, Nigeria*

ABSTRACT

The growth of the digital payment ecosystem presents significant challenges to economic stability and consumer trust because of financial fraud. Traditional rule-based and machine-learning approaches often fail to capture the complex relational patterns present in fraudulent transaction networks. This study develops and evaluates a Graph Attention Network (GAT) model for detecting fraudulent card transactions in the Nigerian financial sector by using graph-based representations to capture relationships among entities. We constructed a heterogeneous graph representation of transaction data from a Nigerian bank, in which nodes represent cards, merchants, account holders, and transactions, while edges represent transaction relationships. A GAT architecture was implemented to learn node embeddings, and an ablation study was conducted to evaluate the contribution of graph structure by comparing the proposed GAT with a no-edge GAT and a randomized-edge GAT. The model was also benchmarked against GraphSAGE, a Gated Graph Recurrent Network (GRNN), a Graph Convolutional Network (GCN), and non-graph baselines including XGBoost, a feedforward neural network, and LightGBM. All models were evaluated on a real-world Nigerian banking dataset using stratified train–test splits, normalized numerical features, and median imputation for missing age values. The results show that graph structure substantially improves fraud detection: the proposed GAT achieved an F1-score of 0.9612, outperforming the no-edge GAT (0.2650) and the randomized-edge GAT (0.8769). Although the GAT achieved a higher F1-score than the GCN (0.9612 versus 0.8998), GraphSAGE achieved a higher AUC than the GAT (0.9961 versus 0.9929), indicating a trade-off between threshold-dependent performance and ranking performance.

Keywords: Fraud detection, Graph neural networks, Financial fraud, Graph attention network, Node classification.

DOI:10.61298/pnspsc.2026.3.295

1. INTRODUCTION

Financial fraud detection is an important task in the financial sector because it has far-reaching implications for security, trust, and economic stability. In the financial sector, fraud includes money laundering, debit- and credit-card fraud [1], and corruption-related activities such as embezzlement, theft of public funds, and abuse of power through extortion [2]. These forms of fraud can cause substantial financial losses; therefore, accurate and timely identification is necessary to minimize risk, even when complete elimination is not possible.

Traditional machine-learning methods often fail to detect sophisticated fraud schemes [3] because they do not adequately capture complex relationships and patterns in financial data.

*Corresponding Author Tel. No.: +234-703-9146-008.
e-mail: akinwandeo@veritas.edu.ng (Oladayo Tosin Akinwande)

Table 1. Bank transaction dataset by transaction type.

Type	Count	Total amount
Credit	16,657	9,155,148,448
Debit	20,440	11,257,883,785

Graph Neural Networks (GNNs), by contrast, exploit graph structures to model interactions among entities such as transactions, users, and merchants [4, 5]. By representing financial data as graphs, where nodes denote entities and edges denote interactions, GNNs can capture both local and global patterns in financial data [5].

The technical implementation of GNN-based fraud detection systems involves constructing graph representations of financial data and applying GNN models to identify fraudulent activities. Financial transactions can be represented as graphs in which nodes represent account holders, payment merchants, or transactions, and edges represent interactions such as payments or transfers. These graphs may be homogeneous or heterogeneous, with heterogeneous graphs capturing diverse relationships among different entity types [6].

Recent studies have shown that GNNs can predict card fraud by modeling transactions as relational graphs, such as card–merchant–device–IP graphs, where higher-order dependencies and coordinated fraud patterns can be learned more effectively than with tabular models [6, 7]. However, most existing GNN-based card-fraud models have been evaluated on non-African benchmark datasets, such as IEEE-CIS, that contain limited contextual features and often rely on fixed or generic graph-construction patterns. This paper addresses that gap by proposing a context-sensitive transaction graph for the Nigerian financial setting and by comparing graph-based and non-graph approaches with local behavioral and contextual signals.

This study demonstrates the use of graph neural networks for detecting debit- and credit-card fraud in financial transactions. The remainder of this paper is organized as follows. Section 2 reviews related work on fraud-detection systems in financial applications using graph neural networks. Section 3 presents the methodology. Section 4 reports and discusses the results. Section 5 concludes the paper and outlines future work.

2. LITERATURE REVIEW

The ability to model complex relationships in transaction graphs enables GNNs to detect fraudulent patterns that are not apparent from isolated transaction records. The Residual-Layered-Camouflage Graph Neural Network (RLC-GNN) is an example of a GNN model that outperforms traditional methods in accuracy, recall, and F1-score, and has achieved strong performance in financial fraud-detection tasks [4]. In the Nigerian financial sector, fraud detection has also used methods such as autoencoders and random forest classifiers with synthetic minority oversampling technique (SMOTE) to address class imbalance; in such settings, autoencoders have outperformed principal component analysis [8]. The use of SMOTE in the Nigerian context has improved accuracy from 98.02% to 99.19%, demonstrating the effectiveness of oversampling techniques for severe class imbalance [9].

Operational challenges remain in Nigerian financial systems, particularly high transaction volumes and the computational cost of graph-based models [8]. Fraud-detection performance is also influenced by graph-construction choices. Transaction-level graphs can capture temporal ordering and fund movement, while account-centric graphs support behavioral profiling based on historical activity [10]. Wei *et al.* [11] reported an accuracy of 93.82% and a recall of 89.5% by combining GAT layers with gradient-boosted decision trees and cost-sensitive learning, thereby leveraging both engineered features and relational learning. Encoder–decoder GNN models have also achieved superior precision, recall, and F1-scores compared with traditional methods in real-world financial datasets [11]. The BRIGHT framework enables real-time fraud detection and reduces latency by more than 75% while maintaining high accuracy, making it suitable for production environments [12].

Heterogeneous graph models have also been applied to credit-card and financial-fraud detection. For example, heterogeneous graph autoencoder models represent credit-card transactions as relationships among cardholders, merchants, and transactions, and use autoencoders to identify deviations from normal patterns [13]. GNNs have also been applied to general financial fraud, such as consumer-loan fraud, crypto-fraud detection, sampled graph convolution, and bitcoin-transaction fraud [14–17]. One example is a GNN with an imbalance-aware mechanism for fraudulent-loan detection, which models the relationships among borrowers, lenders, and transactions [16]. Despite these advances, there is insufficient evidence that GAT architectures have been evaluated on Nigerian card-transaction datasets, creating a research gap in the application of graph-based fraud detection to the Nigerian financial context.

3. METHODOLOGY

The proposed architecture is illustrated in Figure 1.

3.1. PROBLEM DEFINITION

Let the financial network be a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and each node $v_i \in \mathcal{V}$ has node features represented by $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. The objective is to learn a GNN-based classifier

$$f : (\mathcal{G}, \mathbf{X}) \rightarrow \mathbf{y} \in \{0, 1\}^{|\mathcal{V}|},$$

where $y_i = 1$ indicates that node v_i is fraudulent.

3.2. DATASET DESCRIPTION

The dataset is available on request, has undergone organizational ethics-approval processes, and is available for research purposes. It consists of 37,097 anonymized credit- and debit-card transaction observations with 16 attributes. The ratios of credit and debit transactions are presented in Table 1.

Each record represents a card-based transaction with variables related to customer profile, card characteristics, and transaction behavior. The features include numerical and categorical variables such as transaction amount, customer age, card type, transaction type, domain, and outcome. The target feature, *Outcome*, indicates whether a transaction is fraudulent (0) or legitimate (1). A description of the 16 dataset attributes is provided in Table 2.

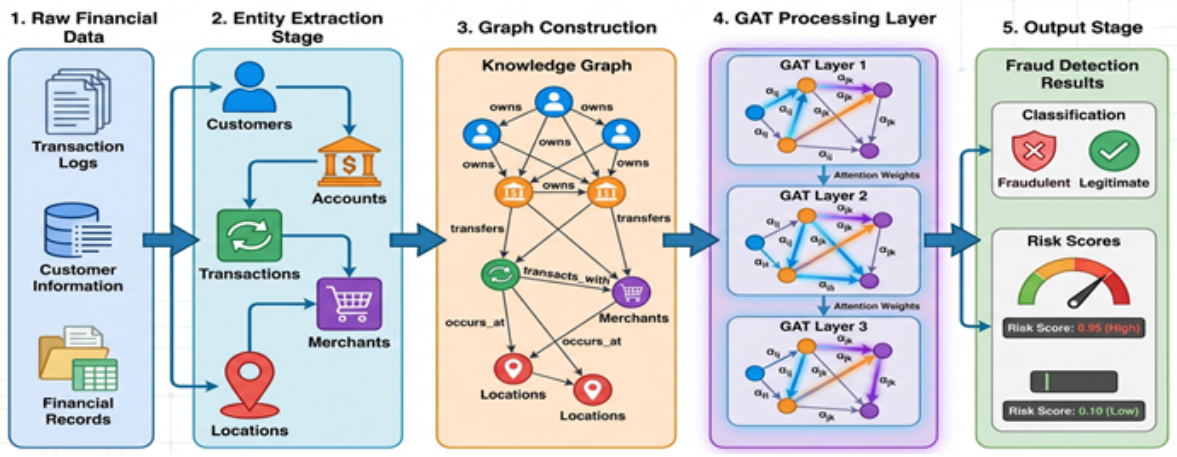


Figure 1. GAT architecture for financial-fraud detection.

Table 2. Dataset attributes.

No.	Feature	Description	Type
1	CustomerAge	Customer age	Numerical
2	ATM	ATM transaction limit	Numerical
3	POSWEBLimit	POS/web transaction limit	Numerical
4	CreditLimit	Credit-card limit	Numerical
5	Amount	Transaction amount	Numerical
6	AverageIncomeExpenditure	Average income/expenditure	Numerical
7	NewBalance	Post-transaction balance	Numerical
8	OldBalance	Pre-transaction balance	Numerical
9	Gender	Customer gender	Categorical
10	Marital Status	Customer marital status	Categorical
11	Cards	Card type	Categorical
12	CardColour	Card colour	Categorical
13	CardType	Card brand/type	Categorical
14	TransactionType	Type of transaction	Categorical
15	Domain	Transaction domain	Categorical
16	Outcome	Target label	Binary

Table 3. Exploratory analysis of numerical features.

Feature	Missing	Count	Min	Max	Mean	SD	25th	50th	75th
CustomerAge	8,851	28,246	18	85	39.2	20.1	23	29	55
ATMLimit	0	37,097	120,000	150,000	140,870.7	13,803.6	120,000	150,000	150,000
POSWEBLimit	0	37,097	1,200,000	4,000,000	2,452,101.2	1,066,813.9	2,000,000	2,000,000	4,000,000
CreditLimit	0	37,097	150,000	600,000	335,101.2	169,488.7	200,000	200,000	500,000
Amount	0	37,097	100,003	999,956	550,261.0	260,629.8	324,480	550,293	775,075
AverageIncomeExpenditure	0	37,097	100,017	399,971	227,387.0	78,977.1	161,178	222,530	283,447
NewBalance	0	37,097	-897,378	1,591,355	294,806.6	623,961.4	-239,814	215,432	850,992
OldBalance	0	37,097	100,001	599,990	350,202.8	144,781.9	224,577	348,916	477,291

Table 4. Exploratory analysis of categorical features.

Feature	Missing	Unique categories	Most common	Distribution
Gender	0	2	Male	Male: 23,186; Female: 13,911
Marital Status	0	4	Married	Married: 17,257; Single: 14,393; Divorced: 2,743; Unknown: 2,704
Cards	0	3	Debit	Debit: 18,550; Credit: 11,289; Prepaid: 7,258
CardColour	0	2	Gold	Gold: 18,550; White: 18,547
CardType	0	3	Verve	Verve: 18,550; Visa: 11,289; MasterCard: 7,258
TransactionType	0	2	Debit	Debit: 20,440; Credit: 16,657
Domain	0	2	International	International: 26,497; Local: 10,600
Outcome	0	2	1	1: 27,370; 0: 9,727

Table 5. Graph Attention Network algorithm.

Input: $G = (V, E)$, $X \in \mathbb{R}^{N \times F}$, neighborhoods $\mathcal{N}_i$, attention heads K.
Output: $H' \in \mathbb{R}^{N \times (KF')}$ for intermediate layers, or $H' \in \mathbb{R}^{N \times F'}$ for final averaging.
1. For each head $k = 1, \ldots, K$, compute $Z^k = X(W^k)^T$.
2. For each edge (i, j) with $j \in \mathcal{N}_i$, compute $e_{ij}^k = \text{LeakyReLU}((a^k)^T [z_i^k \| z_j^k])$.
3. For each node i, compute $\alpha_{ij}^k = \text{softmax}_{j \in \mathcal{N}_i}(e_{ij}^k)$.
4. Update $h_i^k = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k z_j^k\right)$.
5. Concatenate head outputs in intermediate layers, or average them in the final layer.

Table 6. Ablation results.

Model	F1-score	AUC
GAT + random edges	0.8769	0.9319
GAT + no edges	0.2650	0.5000
GAT + true graph (proposed)	0.9612	0.9929

Table 7. Confusion matrix for the GAT model.

	Predicted positive	Predicted negative
Actual positive	1,958 (TP)	146 (FN)
Actual negative	16 (FP)	1,590 (TN)

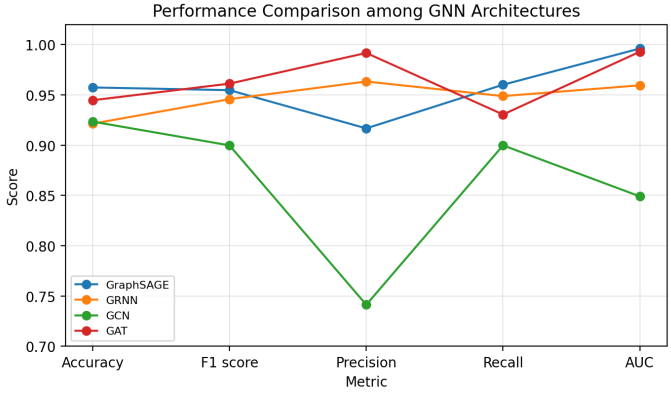


Figure 2. Performance comparison among GNN architectures.

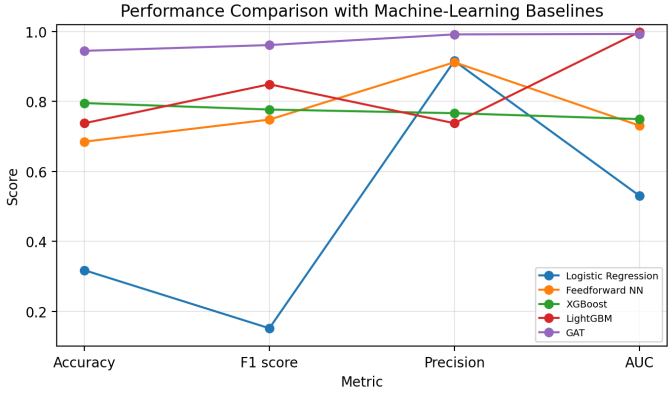


Figure 3. Performance comparison with machine-learning baseline models.

3.3. EXPLORATORY DATA ANALYSIS

Exploratory analysis was performed to summarize the numerical and categorical features. The numerical EDA is presented in Table 3, while the categorical EDA is presented in Table 4.

The CustomerAge feature had 8,851 missing values, representing approximately 23.8% of the records. These missing values were replaced during preprocessing with the median value of 35 years. Customer ages ranged from 18 to 85 years, with a mean of 39.2 years and a standard deviation of 20.1 years. The ATM, POSWEBLimit, and CreditLimit variables represent customer transaction limits. ATM limits were concentrated near the maximum threshold of 150,000. POSWEBLimit varied more widely, ranging from 1.2 million to 4 million, while CreditLimit had a mean of 335,101, indicating heterogeneous customer credit capacities.

Transaction amounts ranged from 100,003 to 999,956, with a mean of 550,261 and a standard deviation of 260,629.81. AverageIncomeExpenditure had a mean of 227,386, with less dispersion than transaction amount. NewBalance and OldBalance represent account balances after and before the transaction, respectively. NewBalance showed substantial variability, including negative balances that may reflect overdrafts or liabilities across debit, credit, and prepaid cards. Transaction amount and balance changes are important for fraud detection because unusual spending or balance patterns may indicate fraudulent activity. Numerical features were normalized using StandardScaler.

The categorical variables had no missing values. Male customers represented 62.5% of the dataset, while female customers represented 37.5%. Married customers formed the largest marital-status category (46.5%), followed by single customers (38.8%). Debit cards accounted for 50% of card usage, while credit and prepaid cards represented 30.4% and 19.6%, respectively. Gold and white cards were nearly equally represented. Verve was the predominant card type (50%), followed by Visa (30.4%) and MasterCard (19.6%). Debit transactions accounted for 55% of transaction types, while credit transactions accounted for 45%. Most transactions were international (71.4%), while local transactions accounted for 28.6%. The target variable was mildly imbalanced, with 73.8% of transactions classified as legitimate and 26.2% classified as fraudulent.

3.4. DATA PREPROCESSING

The data were grouped into transactional data, user data, merchant data, and labels. Transactional variables included transaction type, amount, new balance, old balance, CVC, card information, ATM limit, POS/web limit, and credit limit. User variables included account number, customer age, gender, marital

Table 8. Wilcoxon signed-rank test comparing the GAT with leading baseline models.

Comparison	Metric	Mean (GAT / other)	p-value	Significant at $\alpha = 0.05$
GAT vs. GraphSAGE	F1	0.9612 / 0.9547	0.006	Yes
GAT vs. GraphSAGE	AUC	0.9929 / 0.9961	0.006	Yes, in favor of GraphSAGE
GAT vs. LightGBM	F1	0.9612 / 0.8491	0.005	Yes
GAT vs. LightGBM	AUC	0.9929 / 0.9987	0.008	Yes, in favor of LightGBM

Table 9. Performance comparison among GNN architectures.

No.	Algorithm	Accuracy	F1-score	Precision	Recall	AUC
1	GraphSAGE	0.9574	0.9547	0.9167	0.9601	0.9961
2	GRNN	0.9214	0.9458	0.9633	0.9488	0.9595
3	GCN	0.9234	0.8998	0.7413	0.8998	0.8491
4	GAT model	0.9447	0.9612	0.9917	0.9305	0.9929

Table 10. Machine-learning baseline comparison.

Algorithm	Accuracy	F1-score	Precision	AUC
Logistic regression	0.3176	0.1516	0.9158	0.5305
Feedforward NN	0.6850	0.7481	0.9120	0.7310
XGBoost	0.7954	0.7769	0.7664	0.7495
LightGBM	0.7377	0.8491	0.7377	0.9987
GAT	0.9447	0.9612	0.9917	0.9929

status, average income/expenditure, cards, card type, and card colour. Merchant information was represented by the transaction domain, while the label variable was the investigator-verified outcome.

3.5. GRAPH CONSTRUCTION, FEATURE ENGINEERING, AND REPRESENTATION

Let a heterogeneous graph be defined as $G = (V, E, T, X, A)$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, T is a node-type mapping with $T(v) \in \{\text{Transaction, Card, Merchant, IP}\}$, $X \in \mathbb{R}^{|V|\times d}$ is the node-feature matrix, and $A \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix. Each transaction node is associated with numerical features and one-hot-encoded categorical features, while card and merchant nodes serve as connectors that induce relational structure.

Three graph designs were considered. A homogeneous transaction-only graph was rejected because it discards entity-level relationships such as shared cards. A bipartite transaction–card graph was rejected because it omits merchant and IP information. The selected design was a complete heterogeneous graph because it preserves multi-hop relationships, such as the same card being used at different merchants or the same IP being used with multiple cards.

For a dataset with N transactions and M entities, the total number of nodes is $|V| = N + M$. The adjacency matrix is constructed as

$$A = \begin{bmatrix} 0_{N\times N} & R_{N\times M} \\ R^{\top}_{M\times N} & 0_{M\times M} \end{bmatrix},$$

where $R_{ij} = 1$ if transaction i is associated with entity j. Self-loops are added to stabilize training, yielding $A_{\text{final}} = A + I$. For transaction nodes, each row of X contains normalized numerical and encoded categorical features. Entity-node rows are initialized as zero vectors so that the GAT can learn their embeddings during training. The resulting graph contains approximately 37,097 nodes and 296,776 edges.

3.6. NODE TYPES AND FEATURE ENGINEERING

Customer nodes represent unique customers derived from account identifiers. Customer-level features include CustomerAge, AverageIncomeExpenditure, and transaction count. Card nodes represent distinct cards and include ATMLimit, POSWEBLimit, CreditLimit, and CardType. Merchant nodes are identified by unique MerchantIDs and serve as connection points for merchant-level transaction patterns. Transaction nodes form the core of the graph and include amount, new balance, old balance, and the binary fraud label.

3.7. EDGE TYPES AND SEMANTIC RELATIONSHIPS

Four directed edge types were used to capture the ownership and transaction structure:

1. *Owns* (Customer → Card), representing the ownership relationship between an account holder and a payment card.
2. *Transfers* (Customer → Transaction), representing transaction origination by a customer.
3. *Transaction_with* (Card → Transaction), enabling the model to learn card-specific fraud patterns.
4. *Occur_At* (Transaction → Merchant), linking each transaction to the relevant merchant.

The graph was constructed by transforming raw transactional data into an entity–relationship representation comprising customers, cards, transactions, and merchants as node types, and Owns, Transfers, Transaction_with, and Occur_At as edge types.

3.8. GRAPH CONSTRUCTION PROCESS

The graph-construction process involved three steps. First, unique identifiers were extracted for customers (AccountNumber), cards (CardInformation), merchants (MerchantID), and transactions (TransactionID). Second, source–target pairs were extracted from the original transaction records for each relationship type, with duplicates removed to avoid redundant connections. Third, a graph schema was defined in which transaction nodes connect to the relevant entity nodes, while customer–card ownership forms a hierarchical relationship. The resulting graph

was stored as CSV files: `customers.csv`, `cards.csv`, `transactions.csv`, `merchants.csv`, and `edges.csv`, with the latter including an explicit Edge_Type column.

3.9. GRAPH ATTENTION NETWORK ARCHITECTURE

Let $G = (V, E)$ be a directed or undirected graph with $|V| = N$ nodes. Each node $i \in V$ has an input feature vector $x_i \in \mathbb{R}^F$. Let $\mathcal{N}_i$ denote the one-hop neighborhood of node i, including i when self-loops are used. A single GAT layer maps input features to output features $h'_i \in \mathbb{R}^{F'}$.

GAT first applies a shared linear projection to each node:

$$z_i = Wx_i, \qquad W \in \mathbb{R}^{F' \times F}. \tag{1}$$

For every edge (i, j) with $j \in \mathcal{N}_i$, an unnormalized attention score is computed by

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^\top [z_i \| z_j]\right). \tag{2}$$

The scores are normalized over the neighborhood using a masked softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}. \tag{3}$$

The single-head node update is

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} z_j\right). \tag{4}$$

The GAT model was implemented using PyTorch Geometric. Input features were embedded using learnable linear transformations. Card nodes used 12-dimensional feature vectors, merchant nodes used 8-dimensional vectors, and account-holder nodes used 10-dimensional representations. Two GAT layers with eight attention heads each were employed, and a final linear layer with sigmoid activation produced fraud-probability scores for transaction nodes.

Final fraud prediction is defined as

$$\hat{y}_v = \sigma(w^T h_v + b). \tag{5}$$

The training loss is binary cross-entropy:

$$\mathcal{L} = -\frac{1}{|\mathcal{V}^L|} \sum_{v \in \mathcal{V}^L} \left[y_v \log(\hat{y}_v) + (1 - y_v) \log(1 - \hat{y}_v) \right]. \tag{6}$$

3.10. EXPERIMENTAL SETUP

The dataset was randomly split into training (80%), validation (10%), and test (10%) sets. To preserve the original fraud ratio of approximately 26.2%, stratified sampling was applied using the transaction label. The validation set was used for early stopping during model training over 20 epochs.

Because of the computational cost of training GNNs on a dataset with 37,097 transactions and many edges, five-fold cross-validation was performed. The dataset was mildly imbalanced, so weighted loss was applied during training, giving fraudulent transactions higher weight in the loss function. No oversampling or undersampling was used, to avoid distorting the graph structure. Focal loss with $\alpha = 0.25$ and $\gamma = 2.0$ was also used to address class imbalance. The Adam optimizer was employed with an initial learning rate of 0.001 and exponential decay. Training was conducted on an Intel(R) Core(TM) i7-8650U CPU at 1.90 GHz with four cores, eight logical processors, and 8 GB memory.

3.11. BASELINE COMPARISONS AND EVALUATION METRICS

Model performance was compared against logistic regression, XGBoost, a feedforward neural network, random forest, and LightGBM, as well as graph-based baselines including GraphSAGE, GRNN, and GCN. The evaluation metrics were AUC-ROC, precision, recall, and F1-score.

3.12. ABLATION STUDY

To isolate the contribution of graph structure from node features, three configurations were evaluated:

1. GAT with randomized edges, in which the adjacency matrix was randomly permuted while preserving degree distribution.
2. GAT with no edges, in which the architecture used only node features.
3. GAT with the true graph, corresponding to the proposed transaction–entity graph.

All configurations used the same 80/10/10 split, normalized features, and hyperparameters. The results are summarized in Table 6.

The true graph improved the F1-score by 69.6% over the no-edge GAT and by 8.4% over the randomized-edge GAT, demonstrating that graph structure contributes substantially beyond node features alone.

3.13. ERROR ANALYSIS AND CONFUSION MATRIX

The confusion matrix for the GAT on the held-out test set is shown in Table 7. The matrix contains 3,710 test transactions.

From these counts, the total number of actual positives is 2,104, and the total number of actual negatives is 1,606. Recall is $1958/2104 = 0.9305$, and precision is $1958/1974 = 0.9917$. The 16 false positives represent legitimate transactions flagged as fraud, corresponding to approximately 1% of actual negatives. The 146 false negatives represent missed fraudulent transactions, or 6.95% of actual positives. These errors may occur when transaction-level features appear normal. Future work should incorporate temporal edge aggregation or amount-weighted attention to reduce missed fraud cases without substantially increasing false alarms.

3.14. STATISTICAL SIGNIFICANCE TESTING

To ensure that observed differences were not due to random initialization or data partitioning, all models were trained and evaluated 10 times using different random seeds. For each run, F1-score and AUC were recorded. The paired Wilcoxon signed-rank test was used to compare GAT with GraphSAGE and LightGBM. The results are shown in Table 8.

The Wilcoxon tests show statistically significant differences between the GAT and the best-performing baseline models, but the preferred model depends on the metric. The GAT achieved a significantly higher F1-score than GraphSAGE, indicating a better balance between precision and recall for fraud detection. GraphSAGE achieved a significantly higher AUC, indicating stronger ranking ability across thresholds. The GAT also achieved a much higher F1-score than LightGBM, confirming that graph neural networks capture relational patterns that traditional tabular models may miss. Because the significance tests

were based on only 10 random seeds, future work should use 30 or more seeds to strengthen confidence.

4. RESULTS AND DISCUSSION

4.1. MODEL PERFORMANCE

The GAT achieved strong performance across the evaluation metrics. Table 9 compares the GAT with other GNN architectures.

GraphSAGE achieved the highest accuracy (0.9574) and recall (0.9601), with a strong AUC of 0.9961. Its F1-score (0.9547) was slightly below that of the GAT model, and its precision (0.9167) was lower than those of GRNN and GAT, suggesting a tendency to produce more false positives. GRNN showed balanced performance across all metrics, with high precision (0.9633), competitive recall (0.9488), and an AUC of 0.9595. GCN underperformed relative to the other graph models. Although its accuracy (0.9234) was comparable with GRNN, its precision (0.7413) and AUC (0.8491) were substantially lower.

The GAT model achieved the highest F1-score (0.9612) and precision (0.9917), with an AUC of 0.9929. Although its accuracy was lower than that of GraphSAGE, its superior precision and F1-score indicate that the attention mechanism reduces false positives while maintaining competitive recall. This trade-off makes GAT suitable for applications in which false alarms are costly. GAT outperformed GCN across all metrics, supporting the view that attention-based aggregation can capture more relevant neighborhood information than fixed-weight convolution. GraphSAGE remains competitive for recall-driven applications, while GAT provides the best overall balance in this study.

Table 10 compares the GAT with non-graph machine-learning baselines. The corresponding plot is shown in Figure 3.

In comparative evaluations against baseline models, the GAT substantially outperformed the non-graph models in F1-score and accuracy. The GAT achieved an F1-score of 0.961, compared with 0.777 for XGBoost, 0.849 for LightGBM, and 0.748 for the feedforward neural network. The GAT also achieved superior precision (0.992), minimizing costly false positives. The Wilcoxon tests in Table 8 revealed a trade-off with GraphSAGE: the GAT performed better on F1-score, while GraphSAGE performed better on AUC. Overall, the GAT with the proposed true graph structure is recommended for operational fraud detection when the goal is to balance precision, recall, and ranking performance.

4.2. COMPARISON WITH LITERATURE BENCHMARKS

The GCN accuracy of 0.9234 and F1-score of 0.8998 indicate reasonable performance, while the AUC-ROC of 0.8491 shows acceptable discriminative ability. This is consistent with studies showing that GCNs can capture local graph topology for fraud detection [18, 19]. GraphSAGE achieved an accuracy of 0.9574 and an AUC-ROC of 0.9961, outperforming the other tested architectures on those two metrics. This agrees with reports that GraphSAGE is a promising approach for online transaction security [20] and with studies reporting strong performance from improved GraphSAGE variants [21]. The GAT achieved an accuracy of 0.9447 and an AUC of 0.9929, demonstrating strong fraud-detection performance, although the reported performance remains below some recent state-of-the-art systems [22].

4.3. FUTURE DIRECTIONS

Dynamic graph modeling should be explored to detect evolving fraud schemes, because graph structure changes over time and may contain temporal fraud signals [23]. The generalization ability of GNN models could also be improved through transfer learning across domains, enabling fraud detection in diverse financial environments. Explainability and interpretability should be further investigated to support trust and adoption in real-world systems. Attention mechanisms and model-interpretability techniques can provide insight into model decisions and improve operational acceptance [24].

5. CONCLUSION

This study shows that Graph Attention Networks can effectively model transaction patterns in the Nigerian financial sector. The GAT was trained to classify fraudulent and legitimate transactions and was evaluated against multiple baselines, including GraphSAGE, GRNN, GCN, XGBoost, LightGBM, a no-edge GAT, and a randomized-edge GAT. The results show that graph structure substantially improves fraud-detection performance. The ablation study confirms that relational information provides predictive value beyond node-level features alone.

No single model dominated all metrics. GraphSAGE achieved the highest AUC, while the GAT achieved the best F1-score and precision, making it suitable for settings where false positives are costly. The study is limited by the use of a dataset from a single Nigerian bank and by statistical testing based on 10 random seeds. Future studies should evaluate the approach across multiple African financial institutions and use larger-scale replication. The main contribution of this work is the development and evaluation of a heterogeneous graph schema for Nigerian banking transactions, demonstrating that GNNs can learn complex relational patterns relevant to region-specific fraudulent behavior.

DATA AVAILABILITY

The anonymized card-transaction dataset is available from the corresponding author upon reasonable request, subject to institutional and ethical approvals.

DECLARATION OF COMPETING INTEREST

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

FUNDING

The authors declare that no funding was received during the preparation of this manuscript.

References

[1] W. Hilal, S. A. Gadsden & J. Yawney, "Financial fraud: a review of anomaly detection techniques and recent advances", Expert Systems with Applications **193** (2022) 116429. https://doi.org/10.1016/j.eswa.2021.116429

[2] A. A. S. Alsuwailem & A. K. J. Saudagar, "Anti-money laundering systems: a systematic literature review", Journal of Money Laundering Control **23** (2020) 833. https://doi.org/10.1108/JMLC-02-2020-0018

[3] O. I. Odufisan, O. V. Abhulimen & E. O. Ogunti, "Harnessing artificial intelligence and machine learning for fraud detection and prevention in Nigeria", Journal of Economic Criminology **7** (2025) 100127. https://doi.org/10.1016/j.jeconc.2025.100127

[4] A. Kesharwani & P. Shukla, ‘‘FFDM–GNN: A financial fraud detection model using graph neural network’’, International Conference on Computing and Communication Security (ICCSC), 2024, pp. 1–6. https://doi.org/10.1109/ICCSC62048.2024.10830438

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

[6] D. Cheng, X. Wang, Y. Zhang & L. Zhang, ‘‘Graph neural network for fraud detection via spatial-temporal attention’’, IEEE Transactions on Knowledge and Data Engineering **34** (2022) 3800. https://doi.org/10.1109/TKDE.2020.3025588

[7] N. Jiang, F. Duan, H. Chen, W. Huang & X. Liu, ‘‘MAFI: GNN-based multiple aggregators and feature interactions network for fraud detection over heterogeneous graph’’, IEEE Transactions on Big Data **8** (2022) 905. https://doi.org/10.1109/TBDATA.2021.3132672

[8] O. Onyeama, ‘‘Credit card fraud detection in the Nigerian financial sector: A comparison of unsupervised TensorFlow-based anomaly detection techniques, autoencoders and PCA algorithm’’, arXiv (2024) arXiv:2407.08758. https://doi.org/10.48550/arXiv.2407.08758

[9] B. O. Malasowe, A. O. Adewumi & C. K. Ayo, ‘‘Enhancing the random forest model via synthetic minority oversampling technique for credit-card fraud detection’’, Journal of Computing Theories and Applications **2** (2024) 456. Available online: https://www.researchgate.net/publication/379324623_Enhancing_the_Random_Forest_Model_via_Synthetic_Minority_Oversampling_Technique_for_Credit-Card_Fraud_Detection

[10] M. Guang, Z. Li, C. Yan, Y. Xu, J. Wang, D. Cheng & C. Jiang, ‘‘Multi-temporal partitioned graph attention networks for financial fraud detection’’, IEEE Transactions on Information Forensics and Security **20** (2025) 9399. https://doi.org/10.1109/TIFS.2025.3607231

[11] L. Wei, Y. Li & J. Xu, ‘‘Financial anti-fraud based on dual-channel graph attention network’’, Journal of Theoretical and Applied Electronic Commerce Research **19** (2024) 297. https://doi.org/10.3390/jtaer19010016

[12] M. Lu, Z. Han, S. Rao, Z. Zhang, Y. Zhao, Y. Shan, R. Raghunathan, C. Zhang & J. Jiang, ‘‘BRIGHT: Graph neural networks in real-time fraud detection’’, Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), 2022, pp. 3342–3351. https://doi.org/10.1145/3511808.3557136

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

[14] B. Xu, H. Shen, B.-J. Sun, R. An, Q. Cao & X. Cheng, ‘‘Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field’’, Proceedings of the AAAI Conference on Artificial Intelligence **35** (2021) 4537. https://doi.org/10.1609/aaai.v35i5.16582

[15] A. Singh, A. Gupta, H. Wadhwa, S. Asthana & A. Arora, ‘‘Temporal debiasing using adversarial loss based GNN architecture for crypto fraud detection’’, 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 391–396. https://doi.org/10.1109/ICMLA52953.2021.00067

[16] J. Chen, T. Ma & C. Xiao, ‘‘FastGCN: Fast learning with graph convolutional networks via importance sampling’’, International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 2018, pp. 1–15. https://doi.org/10.48550/arXiv.1801.10247

[17] A. Asiri & K. Somasundaram, ‘‘Graph convolution network for fraud detection in bitcoin transactions’’, Scientific Reports **15** (2025) 11076. https://doi.org/10.1038/s41598-025-95672-w

[18] D. Wang, Y. Qi, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou & S. Yang, ‘‘A semi-supervised graph attentive network for financial fraud detection’’, Proceedings of the IEEE International Conference on Data Mining (ICDM), 2019, pp. 598–607. https://doi.org/10.1109/ICDM.2019.00070

[19] S. Li, J. Zhou, C. Mo, J. Li, G. K. F. Tso & Y. Tian, ‘‘Motif-aware temporal GCN for fraud detection in signed cryptocurrency trust networks’’, arXiv (2022) arXiv:2211.13123. https://doi.org/10.48550/arXiv.2211.13123

[20] A. Kumar & V. Kataria, ‘‘GraphSAGE vs. fraudsters: The future of online transaction security’’, International Journal of Innovative Research in Science, Engineering and Technology **13** (2024) 9422. https://doi.org.10.15680/IJIRSET.2024.1311213

[21] Z. Xiao, ‘‘Research on financial fraud detection method based on graph neural network’’, International Conference on Algorithms, Image Processing, and Deep Learning (AIPDL 2025), 2025, pp. 56–61. https://doi.org/10.1117/12.3078631

[22] Y. Wang & X. Wang, ‘‘Real-time transaction flow analysis with graph neural networks for financial fraud detection’’, Journal of Computational Methods in Sciences and Engineering (2025) 1. https://www.researchgate.net/publication/396605700_Real-time_transaction_flow_analysis_with_graph_neural_networks_for_financial_fraud_detection

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

[24] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu & D. Koutra, ‘‘Beyond homophily in graph neural networks: Current limitations and effective designs’’, Proceedings of Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 7793–7804. https://doi.org/10.48550/arXiv.2006.11468