Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Explainable Financial Fraud Detection Using Graph Neural Networks: A Comparative Study

Nakul Kumar Agarwal

nakulagarwal9042004@gmail.com

Dr. A.P.J. Abdul Kalam Technical University

Mohd. Faizan Malik

Dr. A.P.J. Abdul Kalam Technical University

Sahil Patwal

Dr. A.P.J. Abdul Kalam Technical University

Prakhar Sachan

Dr. A.P.J. Abdul Kalam Technical University

Research Article

Keywords: Financial Fraud Detection, Graph Neural Networks, Graph Convolutional Networks, GraphSAGE, Graph Attention Networks, GNNExplainer, Explainable Artificial Intelligence, Elliptic Bitcoin Dataset.

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11563193/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Explainable Financial Fraud Detection Using Graph Neural Networks: A Comparative Study

Nakul Kumar Agarwal
Computer Science & Engineering (Data Science)
ABES Engineering College
Ghaziabad, India
nakul.25b15430005@abes.ac.in

Sahil Patwal
Computer Science & Engineering (Data Science)
ABES Engineering College
Ghaziabad, India
sahil.24b1541028@abes.ac.in

Mohd. Faizan Malik
Computer Science & Engineering (Data Science)
ABES Engineering College
Ghaziabad, India
faizan.25b15430002@abes.ac.in

Prakhar Sachan
Computer Science & Engineering (Data Science)
ABES Engineering College
Ghaziabad, India
Prakhar.24b1541121 @abes.ac.in

Md. Suhail
Computer Science & Engineering (Data Science)
ABES Engineering College
Ghaziabad, India

Dr. Prabhat Singh
Computer Science & Engineering (Data Science)
ABES Engineering College
Ghaziabad, India
Prabhat.singh@abes.ac.in

Abstract - The financial fraud detection problem is extremely complicated and difficult because of the relationships between transactions in financial systems and the extreme imbalance of fraudulent and legitimate transactions. Traditional machine learning methods might not be suitable for modeling the relational nature of transaction networks, which is the reason why graph neural networks (GNNs) are employed in graph-based fraud detection. This research evaluates the performance of three different GNN architectures: Graph Convolutional Networks (GCN), GraphSAGE, and Graph Attention Networks (GAT) in the field of financial fraud detection using Elliptic's Bitcoin transaction dataset. The dataset is a temporal graph of bitcoin transactions, allowing to analyze transaction attributes and relationships. The study creates a comparative experimental setup for testing the three architectures according to classification metrics, focusing on the precision, recall and F1 score of fraud-class accuracy together with overall accuracy. The interpretability problem of graph-based predictions is overcome by plugging GNNExplainer into a unified explanation pipeline to locate the structure of subgraphs and the features of nodes that are relevant to each prediction from the model. This framework brings together prediction, explanation, and visualization at the instance level, to facilitate a discussion of model performance and interpretability. The goals of the study are to describe the strengths and weaknesses of the architectures evaluated and to evaluate the usefulness of graph-based explanations for financial fraud detection. Experimental conclusions will be verified based on the results of the model and explanation analyses.

Keywords - Financial Fraud Detection, Graph Neural Networks, Graph Convolutional Networks, GraphSAGE, Graph Attention Networks, GNNExplainer, Explainable Artificial Intelligence, Elliptic Bitcoin Dataset.

I. INTRODUCTION

1.1 Background and Motivation

Financial fraud detection is an important challenge due to the increasing volume and complexity of digital financial transactions. Traditional machine learning methods primarily focus on individual transaction characteristics, and can fail to detect relationships between transactions that can indicate suspicious behavior. To overcome this, Graph Neural Networks (GNNs) learn from both the attributes of the transactions and the relationships between these transactions, thus suitable for graph-based fraud detection. But, fraudulent transactions are difficult to detect due to class imbalance, complex transaction relationships, and low predictability of predictive models. While GNNs can enhance the classification of transactions, it is also crucial to explain the rationale behind a fraudulent classification for transparency and investigative purposes. This study thus evaluates GCN, GraphSAGE, GAT, and the ensemble model of GraphSAGE–GAT on the Elliptic Bitcoin transaction data set, and analyzes the influential features and relationships behind the predictions made by the models using GNNExplainer.

1.2 Dataset and Problem Characteristics

In this study, the Bitcoin transaction data are taken from the Elliptic dataset, which is a
Publicly accessible dataset created for research on detecting illicit activity in Bitcoin transaction networks. The graph is a directed graph where the nodes are Bitcoin transactions and the edges are the connections between transactions. It also supports a set of transactionlevel features and temporal information, which can be used to study a graph-based classification financial transaction system. The data set differentiates between legal transactions, illegal transactions, and transactions with unknown labels. This distinction is crucial because not all transactions that arrive in an unknown manner can be trusted. The data additionally comprises temporal data, which offers context concerning what activity is taking place in transactions over time.The experimental analysis is primarily directed at establishing a uniformity in the analysis of the models to be compared, and a clearness in the analysis procedures.

Table 1: Elliptic-Dataset Temporal Split

Dataset split	Time steps	Transactions with known labels
Training	1–34	29,894
Validation	35–41	7,829
Testing	42–49	8,841

1.3 Research Questions

The study is guided by the following research questions:

RQ1: What is the difference between GCN, GraphSAGE and GAT in differentiating illicit transactions from licit transactions in the Elliptic Bitcoin transaction dataset?

RQ2: What are the differences in the precision, recall and F1-score of the evaluated architectures in regards to fraud-class precision, recall and F1-score, especially in the case of class imbalance?

RQ3: How can the features of the nodes and the relationships within the graph be determined based on the individual prediction of the transaction classification using GNNExplainer?

RQ4: How do the limitations of the evaluation of the predictive performance and the interpretation of the explanation in graph-based financial fraud detection arise?

The questions are used to help guide the experimental comparison and then subsequent discussion. Answers will be determined based on the verified results and not based on theory of the architectures.

1.4 Research Contributions

Experimental framework: The study provides a framework for comparing the performance of GCN, GraphSAGE and GAT for the same financial transaction classification task.
Class sensitive performance analysis: Evaluation includes class sensitive metrics as well as overall accuracy, providing a more informative evaluation of the detection of illicit transactions when the classes are imbalanced.
Shared GNNExplainer implementation: The study features an integrated implementation of GNNExplainer, which can be used across the various node-classification architectures evaluated, with a common mechanism for generating local explanations.
Joint examination of prediction and explanation: The study takes a look at predictive performance and the instance-level explanations, emphasizing the need to study both classification results and the information identified as relevant to individual predictions.
Comparative evaluation and generated explanations; identification of choices of explanatory evidence, and causal claims related to transaction behaviour, with regard to limitations of the dataset.
These contributions are related to the application, comparison and evaluation of current methods. The study is not an attempt to propose a new graph neural network architecture or an explanation algorithm.

2. LITERATURE REVIEW

2.1. Traditional and Deep Learning Approaches to Financial Fraud Detection

The traditional fraud detection system relies on patterns and characteristics that are preprogrammed and identified by machine learning algorithms. These techniques can have difficulties, however, to capture complex relationships between transactions and to adjust to new trends in fraud. Graph-based deep learning has thus attracted attention, as it can learn from transactions' attributes and network connectivity. Motie and Raahemi [6] surveyed the financial fraud detection using graph neural networks (GNNs) and pointed out its potential in various financial fields.

2.2. Graph-Based Financial Fraud Detection and the Elliptic Bitcoin Dataset

Graph-based financial fraud detection and. Elliptical bitcoin dataset Graph-based total fraud detection fashions economic hobbies as interconnected entities, allowing dubious styles to be recognized through relationships between transactions Weber 12et al. [1] verified the use of graph convolutional networks for bitcoin anti-cash laundering analysis. Similarly, Paurahabibi and others. [8] reviewed fully graph-based anomaly detection strategies for fraud detection, emphasizing the importance of analyzing relationships within a network. The Elliptic Bitcoin dataset supports this study by supplying transaction capabilities, graph connectivity, temporal statistics, valid, invalid, and unknown labels These characteristics make it suitable for evaluating the GNN architecture for illegal behavior class.

2.3. Graph Neural Network Architectures

In graph neural networks, nodes are represented by learning them from other nodes in the graph and their relationships. While there is a common theme across the various architectures of the GNN, the way in which neighbourhood information is combined and fed into node representations varies.

a) Graph Convolutional Network (GCN): Kipf and Welling introduced Graph Convolutional Networks for semisupervised classification of graph-structured data [2]. GCN uses node features with information from neighbouring nodes to propagate information through the graph using normalized graph connectivity to do so. This means information from a node's neighbourhood can be added to the representation of the node, as can its own attributes. This pattern-matching approach works well in financial fraud detection, where it can be used to detect suspicious patterns that may not be visible based on the attributes of the transactions themselves. From this point of view, GCN can serve as a valid baseline to assess graph-based transaction classification.

b) GraphSAGE: Hamilton et al. proposed the inductive method of learning node representations with neighbourhood sampling and aggregation, named GraphSAGE [3]. Instead of learning an independent embedding for each node, GraphSAGE learns to aggregate node features and neighbourhood information to produce node representations. This design can be used for representation learning of nodes that have not been encountered before, but their features and neighbourhood information are known. In general, GraphSAGE provides another approach to financial transaction analysis compared to the traditional graph convolution, which learns how to use information from neighbouring transactions to represent a transaction node.

c) Graph Attention Network (GAT): Veličković et al. [4] proposed graph attention networks, which use an attention mechanism to assign exclusive importance weights to neighboring nodes during statistics collection Unlike procedures that use fixed generalization schemes and aggregate adjacent information, GAT learns coefficients of interest that decide the relative contributions of related nodes to behavioral graphs, this mechanism can also help to convey a more informative relationship for a version class. However, discovered interest weights should not routinely be interpreted as the entire or causal motivation for forecasting. GCN, GraphSAGE, and GAT consequently provide 3 distinct strategies for graph-based node classification. Their architectural differences encourage a managed comparative examination in which all fashions are evaluated against consistent information practice and ranking measures of use. The relative effectiveness of architectures from their theoretical homes has to be established empirically rather than assumed.

2.4. Ensemble Learning and the GraphSAGE–GAT Ensemble

Ensemble learning merges multiple models' predictions to take advantage of their complementary strengths. GraphSAGE derives node representations through information aggregation of neighbor nodes [3], while Graph Attention Networks (GAT) apply different importance weights to neighbor nodes [4]. The two architectures provide different ways of learning transaction relationships. In this study, their predicted probabilities are merged to form a GraphSAGE-GAT ensemble using validation data for the selection of the decision threshold. The comparison of its performance with the GCN, GraphSAGE, and GAT models will reveal if the combination of neighborhood aggregation and attention learning enhances fraud classification.

2.5. Explainable Artificial Intelligence and GNNExplainer

Although graph neural networks are capable predictors, they may not be interpretable, reducing the transparency of individual decisions. XAI overcomes this problem by determining the factors that influence model predictions. Ying et al. [5] suggested GNNExplainer that identifies the important node features and graph structures related to the model predictions. The more recent research has also considered the relationship between graph learning and explainability in the area of financial AI [12]. GNNExplainer will be employed in this study to analyze the influential features and neighboring edges in the transaction classification, adding explainability

2.6. Comparative Evaluation and Research Considerations

Evaluating financial fraud detection requires multiple metrics because class imbalance can make accuracy alone misleading. Tong and Shen [9] investigated graph neural networks alongside imbalance-learning considerations, highlighting the relevance of addressing imbalanced transaction classes. This study therefore evaluates models using accuracy, fraud precision, fraud recall, and fraud F1-score. These metrics provide complementary information about overall classification performance and the identification of fraudulent transactions. Comparing GCN, GraphSAGE, GAT, and the GraphSAGE–GAT ensemble enables a consistent assessment of their relative performance, while explanations provide additional insight into model decisions.

2.7. Research Gap and Motivation for the Present Study

Existing studies and reviews demonstrate the growing importance of graph neural networks in financial fraud detection [6], [11]. Research has also explored temporal graph approaches for detecting fraudulent transactions [10], while explainability has emerged as an important consideration for transparent financial AI [12]. However, model selection still requires empirical evaluation on a common dataset, alongside investigation of the factors influencing individual predictions. This study addresses these objectives by comparing GCN, GraphSAGE, GAT, and a GraphSAGE–GAT ensemble on the Elliptic Bitcoin dataset and applying GNNExplainer for transaction-level interpretation. The resulting framework examines predictive performance and interpretability within a unified experimental study.

3. THEORETICAL BACKGROUND

3.1. Graph Representation of Financial Transactions

Graph representation of financial transactions. A monetary transaction network is represented as a graph in which nodes form transactions and edges represent relationships between them. Each node is associated with an attribute vector and a class label.

$$X \in \mathbb{R}^{N \times F} \qquad \ldots(1)$$

3.2. Graph Neural Networks and Message Passing

Graph neural networks and message passing from its neighboring nodes Messaging enables each node to aggregate facts for account-aware modification Representation. These representations are used to classify behaviors based on their features and graph combination.

$$m_v^l = AGG^{(l)}\left(\left\{\varphi^{(l)}\left(h_v^l, h_u^l\right) : u \in N(v)\right\}\right) \qquad \ldots(2.1)$$

$$h_v^{l+1} = UPDATE^{(l)}\left(h_v^l, m_v^l\right) \qquad \ldots(2.2)$$

3.3. Graph Convolutional Network

GCN aggregates and transforms information from neighboring nodes using normalized graph connectivity. It learns node representations by combining transaction features with information from the surrounding graph structure.

$$H^{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l\right) \qquad \ldots(3.1)$$

3.4. GraphSAGE

GraphSAGE generates node embeddings by sampling and aggregating features from neighboring nodes and allowing it to learn representations using local neighborhood information.
A simplified GraphSAGE update can be expressed as

$$h_{N(v)}^l = AGG^{(l)}\left(\{h_u^l : u \in N(v)\}\right) \qquad \ldots(4.1)$$

$$h_v^{l+1} = \sigma\left(W^l\left[h_v^l \| h_{N(v)}^l\right]\right) \qquad \ldots(4.2)$$

3.5. Graph Attention Network

GAT uses attention mechanisms to assign different importance weights to neighboring nodes during feature aggregation. This enables the model to focus on more informative neighbors when learning transaction representations.

$$e_{ij} = LeakyReLU\left(a^T\left[Wh_i \| Wh_j\right]\right) \qquad \ldots(5.1)$$

The scores are normalized over the neighbourhood of node (i):

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})} \qquad \ldots(5.2)$$

The updated representation is then calculated as

$$h_i' = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W h_j\right) \quad \ldots(5.3)$$

where (W) is a trainable transformation matrix, Implementations may include self-loops and multi-head attention, depending on their configuration.
Unlike uniform or fixed neighborhood weighting, attention enables GAT to learn different contributions for different neighbouring nodes. However, attention coefficients are internal model weights and should not, by themselves, be considered complete explanations of a prediction.

3.6. GraphSAGE–GAT Probability Ensemble

The ensemble combines GraphSAGE and GAT predictions using a weighted probability-based strategy, leveraging neighborhood aggregation and attention mechanisms. The combined probabilities are used to classify transactions, with the decision threshold selected using validation data.

$$p_{ens(v)} = w_S p_{SAGE(v)} + w_A p_{GAT(v)} \quad \ldots(6.1)$$

This equation represents a two-model weighted ensemble; the actual weights must match the project's saved or validation-selected configuration.

The final class prediction is determined using a threshold

$$y_{hat_v} = 1 \; if \; p_{ens(v)} \geq \tau; \; 0 \; otherwise \quad \ldots(6.2)$$

3.7. GNNExplainer for Model Interpretability

GNNExplainer explains GNN predictions by identifying important node features and edges within the local computation graph that contribute to a transaction's classification. In this study, it is used to interpret predictions from GCN, GraphSAGE, and GAT models, providing insights into the features and relationships influencing fraud detection.

$$E_v = (M_X, M_E) \quad \ldots(7)$$

The masks are optimized to identify information that preserves the model's prediction while encouraging a concise explanation. Their values depend on the trained model, input graph, explainer settings, and optimization process.
In the present study, a shared GNNExplainer workflow is used to support compatible node-classification models that return raw class logits. The workflow extracts important node features and edges for an individual transaction and is designed to accommodate the GCN, GraphSAGE, and GAT architectures. It can also be applied to the GraphSAGE–GAT ensemble through an adapter that preserves the ensemble's configured probability-combination and threshold decision rule.

3.8. Evaluation Metrics

Let (TP) denote true positives, (TN) true negatives, (FP) false positives, and (FN) false negatives, with the positive class representing illicit transactions under the verified label mapping.

1) Accuracy: Accuracy measures the proportion of correctly classified evaluated transactions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots(8.1)$$

2) Precision: Precision measures the proportion of transactions predicted as fraudulent that are actually labelled fraudulent.

$$Precision = \frac{TP}{TP + FP} \quad \ldots(8.2)$$

3) Recall: Recall measures the proportion of actual fraudulent transactions correctly identified by the model.

$$Recall = \frac{TP}{TP + FN} \quad \ldots(8.3)$$

.

4) F1-score: The F1-score is the harmonic mean of precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \ldots(8.4)$$

4. METHODOLOGY

4.1 Overview of the Proposed Framework

The proposed framework searches for a suitable monetary to explain Fraud detection using graph-neural networks and a Probability-based grouping. The method includes data set guidance, graph construction, model development, version school learning, overall performance evaluation, cluster forecasting, and clarification technology Three individual graph neural network architectures—Graph Convolutional Network (GCN), GraphSAGE, and Graph Attention Network (GAT)—are considered for comparatiive evaluation In addition, the GraphSAGE–GAT group has been integrated to research the joint predictor behavior of unique graphs identifying architectures.

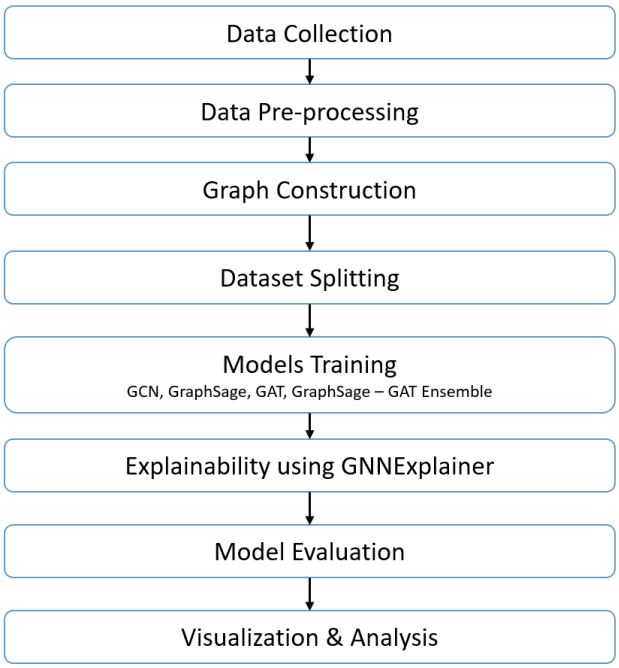


Figure 1 : Proposed methodology for explainable financial fraud detection using graph neural networks.

4.2 Dataset Description

The data comes from the Elliptic Bitcoin transaction graph, which was introduced for research into identifying malicious activity in the Bitcoin transaction network [1]. The dataset

consists of all the Bitcoin transactions as nodes and the relationship between each transaction as an edge. The graph structure represents the relationship between transactions and each node has a set of features related to the transactions. The dataset also includes some temporal information in the form of time steps. This allows transactions to be reviewed in the time dimension of the data set. The classification labels help differentiate the available transaction types such as licensed transactions, illicit transactions, and unknown transactions from the original dataset.

4.3 System Architecture

The proposed system consists of five main layers: data input, preprocessing, graph construction, model processing, and application visualization. The Elliptic Bitcoin dataset is processed to prepare transaction features, labels, and graph connectivity. The resulting graph is used by GCN, GraphSAGE, GAT, and the GraphSAGE–GAT ensemble for transaction classification. Model performance is evaluated using accuracy, fraud precision, fraud recall, and F1-score. GNNExplainer generates transaction-level explanations by identifying important features and edges. Finally, the Streamlit application presents predictions, model comparisons, graph visualizations, and explanations through an interactive interface.

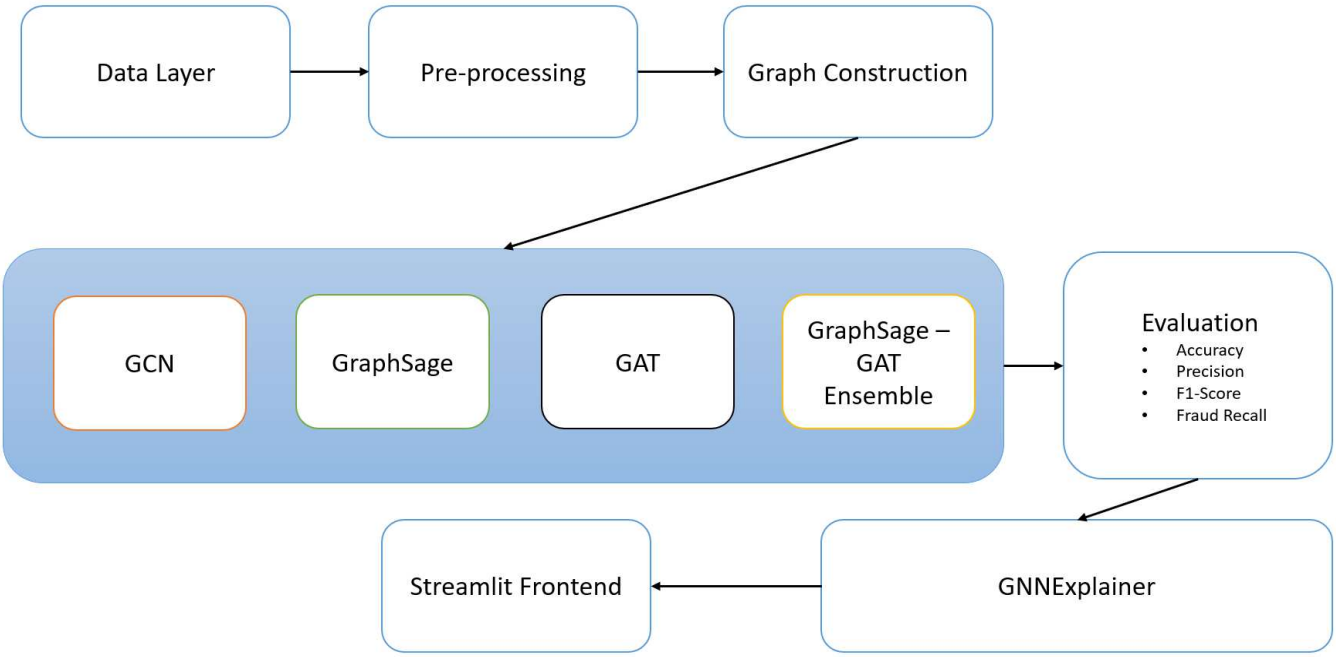


Figure 2 : System Architecture

4.4 Data Preprocessing and Graph Preparation

The preprocessing stage prepares the transaction attributes, labels, and graph connectivity for model training. The transaction feature data are organized into a node feature matrix, while the edge information is converted into the graph representation required by the graph neural network implementation.

First, the transaction features and corresponding identifiers are aligned to ensure that each feature vector refers to the correct graph node. The graph connectivity information is then represented using an edge index or an equivalent structure supported by the implementation. Maintaining consistent node indexing is essential because a mismatch between transaction identifiers and graph indices can lead to incorrect predictions and explanations.

Second, the original class labels are mapped into the format required by the classification pipeline. The mapping must distinguish the classes used for supervised learning from transactions whose labels are unknown. Unknown-label handling is particularly important when calculating performance metrics, since unlabeled transactions must not be counted as confirmed legitimate or fraudulent examples without justification.

Third, the data are prepared for training, validation, and testing according to the project's implemented split strategy. Any preprocessing operations that estimate parameters from data must be handled in a manner that prevents information from the test set from influencing model development. The same evaluation protocol should be maintained across the candidate models to support a fair comparison.

The resulting graph representation provides the features, connectivity information, and applicable labels required for the subsequent model training stage.

4.5 Graph Neural Network Model Development

Four graph neural network architectures are evaluated in the proposed framework.

a)Graph Convolutional Network: GCN learns node representations by aggregating and transforming information from neighboring nodes through normalized graph convolution operations. It serves as a baseline graph learning architecture for evaluating the contribution of transaction relationships to classification performance.

b)GraphSAGE: GraphSAGE generates node representations by aggregating information from a node's neighborhood. Its aggregation-based design provides an alternative to the graph convolution operation used by GCN and enables the model to learn representations from local graph structure.

c)Graph Attention Network: GAT applies attention coefficients to neighborhood information, allowing the model

to assign different learned importance weights to neighboring nodes. This architecture provides a different approach to incorporating relational information into transaction classification.

d)GraphSAGE–GAT Ensemble: The GraphSAGE–GAT ensemble combines the predicted fraud probabilities of GraphSAGE and GAT using a weighted probability-based strategy. The ensemble weights and classification threshold are selected using validation data, and the resulting predictions are evaluated on the test set alongside the individual GNN models.

The four models are trained for the same general node-classification task. Their predictions are evaluated using a consistent set of metrics, with particular attention to the performance of the illicit or fraud class. Model-specific settings and training configurations should be documented from the actual implementation to ensure reproducibility.

4.6 Model Training and Prediction

In the training phase, each model is given the prepared node features and connectivity of the graph, and is trained to predict the class that corresponds to each eligible transaction. The trainable parameters are optimized by applying the training data and loss function adopted in the project. Validation data (where available) are used for model selection and tuning of the relevant model configuration parameters. Each model yields class scores or probabilities for the assessed nodes after training. These outputs are mapped to predicted labels based on the decision procedure implemented in the model. When there is binary fraud classification, the decision threshold is the value that is used to convert predicted probabilities of fraud into class predictions. The training/evaluation procedures need to maintain the separation between training, validation and test data. A particular advantage of this is that the test set shouldn't to be used to select model parameters or optimize the decision threshold. This helps to ensure that the reported results are valid models' generalization over the data used for model development.

4.7 GraphSAGE–GAT Probability Ensemble

The combination is done based on the ensemble weights provided in the implementation. The resulting probability is then passed through a decision threshold to give the final class prediction. When weights and threshold are selected by experiment, they should be selected from validation data instead of the final test set.If selected by experiment, then the weights and threshold should be selected from validation data and not the final test set. The ensemble performance is compared with GCN, GraphSAGE, and GAT for the same metrics that are applicable to the three models. This comparison is to determine if there is a benefit to using the combined predictions from GraphSAGE and GAT, as opposed to using them independently. All improvements or limitations are derived by way of experiment and not in actuality.

4.8 Model Evaluation

The models are evaluated using accuracy, fraud-class precision, recall for fraud class and fraud class F1 score. Precision represents the percentage of all transactions that are predicted as fraudulent that are actually fraudulent, while accuracy refers to the overall percent correct. The recall is the percentage of correct fraudulent transactions classified as such, and the F1-score is a summary of the precision and recall. Financial Fraud datasets may be heavily imbalanced, so accuracy is not enough to characterize model performance. Thus, fraud-class precision, recall, and F1-score are important for comparing the models' capabilities to identify fraudulent transactions that include the consideration of false-positive predictions. To evaluate all candidate models, the same test data set shall be used and the interpretation of the positive class shall be the same. The comparative is to be based on the values which have been measured from the experimental record. No model should be presented as a better model without the necessary results to back it up.

4.9 Explainability Using GNNExplainer

The framework has integrated GNNExplainer, a tool to study the interpretability of the trained graph neural networks. The explainer produces instance-specific explanations for a set of node predictions, by looking for relevant input features and local graph interactions corresponding to model output. The shared explanation workflow is intended for models, which are compatible with different node-classification models returning class logits. When choosing a specific transaction, important features and edges within the scope of the explainer's local explanation are identified in this explanation process. These outputs can then be displayed as feature rankings and visualizations of the local graph. The explanation stage will help complete the prediction evaluation, by adding more details about specific model decisions. It might enable the exploration of neighboring relationships and transaction attributes emphasized for a chosen prediction, for instance. Comparisons can be made across architectures that are supported to examine if different models use the same or different information to achieve the same results. The explanations are assumed to be evidence for the prediction, rather than evidence that the transaction is fraudulent or that a highlighted feature leads to fraud. The reliability of the explanations depends on the model, the node chosen, the node configuration of the explanation and the stability of the resulting masks. If applicable, the quality of the explanations should be explored by using fidelity or sensitivity analyses.

5. EXPERIMENTAL SETUP AND IMPLEMENTATION DETAILS

5.1 Experimental Environment

The experiments use the Elliptic Bitcoin transaction dataset, with preprocessing and graph construction performed as described in Section IV. Four model configurations are evaluated: GCN, GraphSAGE, GAT, and the GraphSAGE–GAT ensemble. Implementation-specific settings, including model hyperparameters and training configurations, follow the project implementation.

5.2 Training and Validation Procedure

The models are trained using the designated training data, while validation data is used for model selection and tuning. For the GraphSAGE–GAT ensemble, the combination weights and classification threshold are selected using validation data. The test set is reserved for final performance evaluation to avoid using test results during model selection.

5.3 Evaluation Protocol

Model performance is assessed using accuracy, fraud precision, fraud recall, and fraud F1-score. The same evaluation protocol is applied to GCN, GraphSAGE, GAT,

and the GraphSAGE–GAT ensemble. GNNExplainer is used to examine relevant node features and graph edges contributing to individual predictions, complementing quantitative evaluation with local explanations.

6. RESULTS

6.1. Individual Model Results

GCN: Fig. 3 presents the GCN confusion matrix, precision-recall curve, ROC curve and Training Loss Graph.

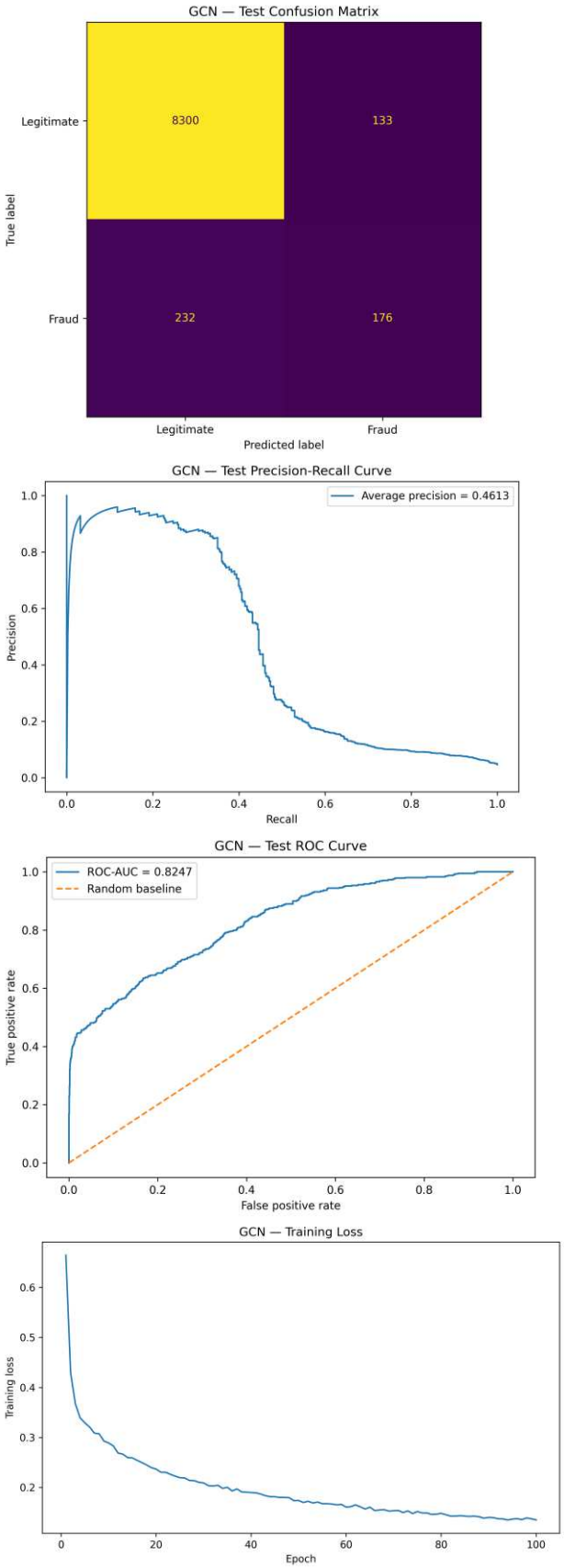


Figure 3: Performance evaluation of the GCN model

GraphSAGE: Fig. 4 shows the GraphSAGE performance results, providing insight into its fraud detection performance and its ability to identify fraudulent transactions.

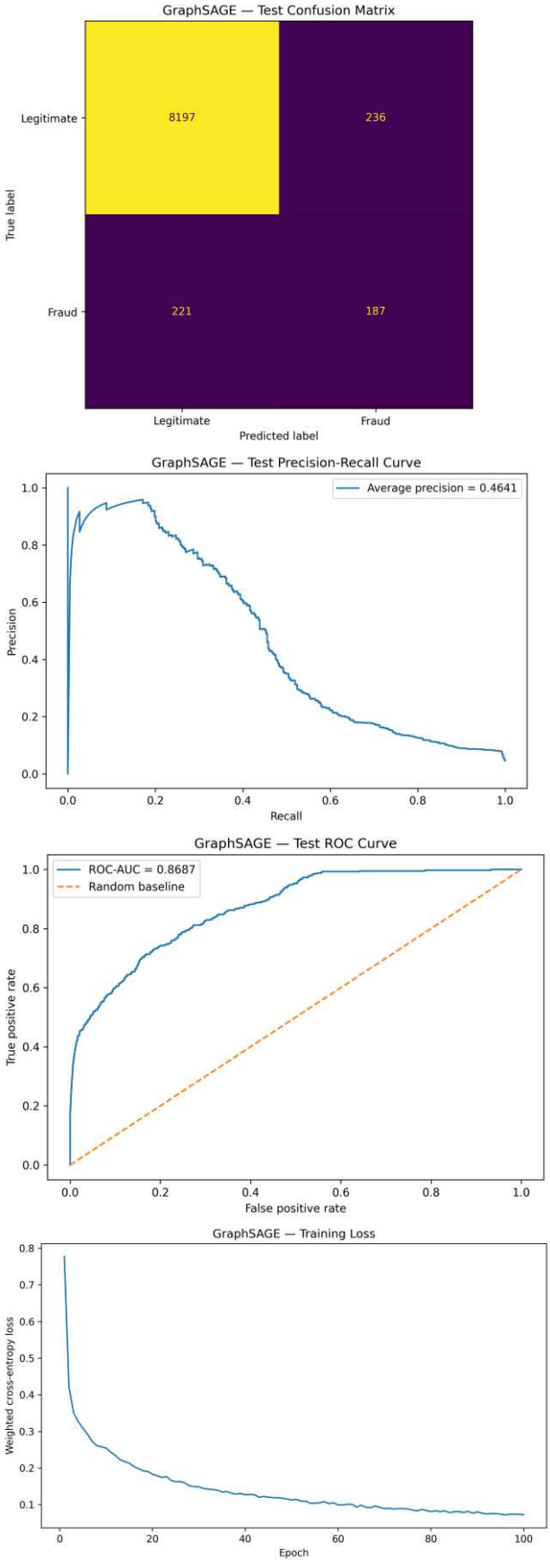


Figure 4 : Performance evaluation of the GraphSAGE model

GAT: Fig. 5 presents the GAT confusion matrix. Its results highlight the model's classification behavior when attention-based neighborhood aggregation is used.

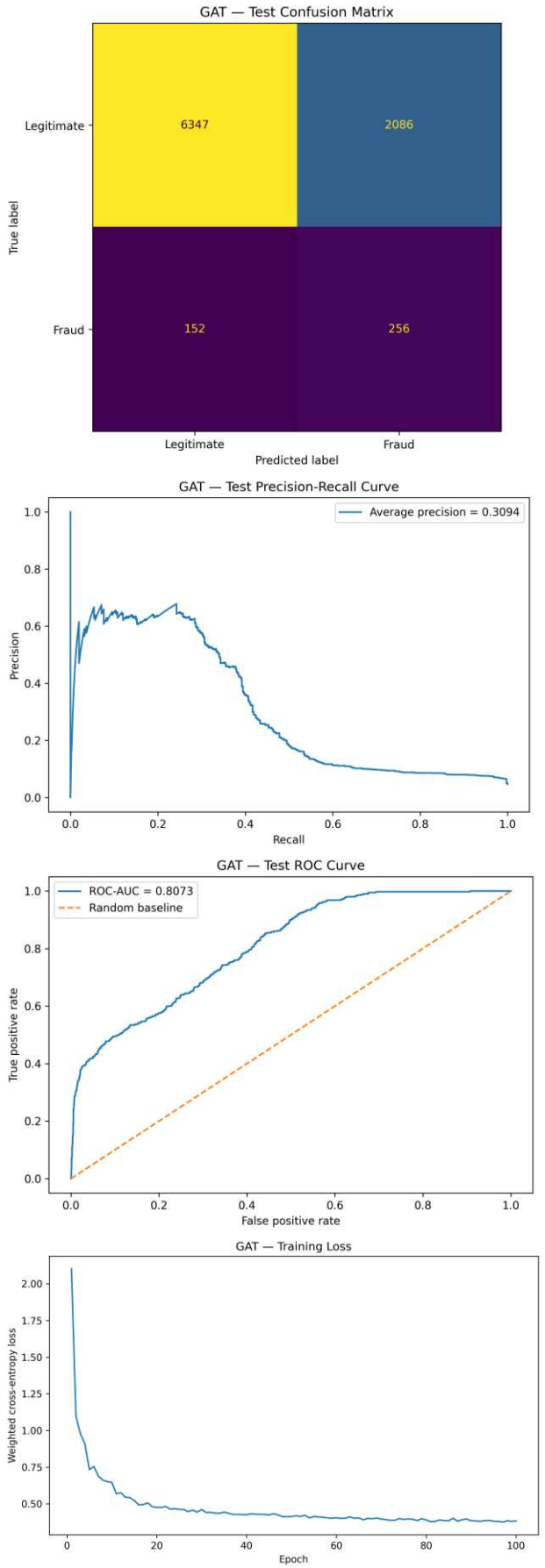


Figure 5 : Performance evaluation of the GAT model

6.2. Analysis of the GraphSAGE–GAT Ensemble

Fig. 6 presents the confusion matrix of the GraphSAGE–GAT ensemble. By combining the predicted probabilities of GraphSAGE and GAT, the ensemble is evaluated to determine whether it improves fraud detection performance compared with the individual models. Its effectiveness is assessed using the same evaluation metrics and test set.

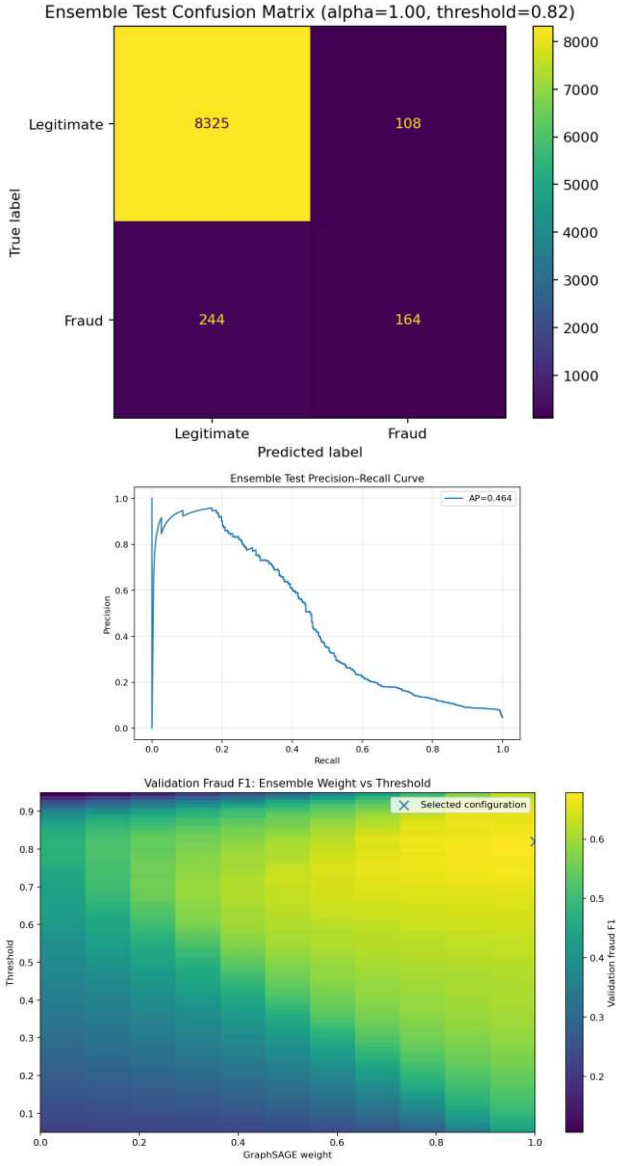


Figure 6 : Performance evaluation of the GraphSAGE–GAT ensemble

6.3. Comparative Performance of the Models

Table presents the framework for comparing the four prediction configurations. The numerical entries must be populated using the verified results produced by the final experiments.

Table 2 : Comparative Performance of the Evaluated Models

Model	Accuracy (%)	Fraud Precision (%)	Fraud Recall (%)	Fraud F1 (%)
GCN	95.87	56.96	43.14	49.09
GraphSAGE	94.83	44.21	45.83	45.01

GAT	74.69	10.93	62.75	18.62
GraphSAGE–GAT ensemble	96.02	60.29	40.20	48.24

The table enables a direct comparison of overall classification performance and fraud-class detection. The model with the highest accuracy is not necessarily the model with the highest fraud recall or F1-score. Therefore, conclusions should consider the full set of metrics rather than relying on a single performance measure.

6.4. Explainability Results Using GNNExplainer

To explore the features and graph edges that are important to selected transaction predictions, GNNExplainer was employed. Fig. X provides some illustrative explanations, emphasizing the features that were influential and transaction relationships in the local area. These visualizations can be used to gain further understanding of the model decisions but should not be used to make inferences regarding causal relationships.

6.5. Limitations of the Experimental Findings

This study is limited to the Elliptic Bitcoin dataset and the experimental conditions. The results may be influenced by class imbalance, unknown transaction labels, limitations of local explanations. The comparative evaluation looks at trade-offs between the four model configurations in terms of their predictive performance and their interpretation.

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this study, two GNN architectures are compared for financial fraud detection, using the Elliptic Bitcoin transaction dataset. Four model configurations: Graph Convolutional Network (GCN), GraphSAGE, Graph Attention Network (GAT) and an ensemble of GraphSAGE and GAT (GraphSAGE–GAT Ensemble) are explored to study their performance in transaction network-based illicit transaction classification, based on the features of the transaction and graph relationship. Accuracies are not the only measures used for modeling performance, as these are measured using accuracy, as well as the fraud precision, the fraud recall and the fraud F1-score. The study also takes into account the interpretability of the individual GNN predictions by incorporating GNNExplainer. The explanation process reveals top features of the nodes as well as their local relationships in the graph, which helps to understand the factors of the transaction types. This provides a complement to quantitative evaluation, allowing users to explore the role of graph structure and transaction attributes in decisions in the model. In general, this research is an integrated approach to comparative model evaluation and local explainability in the context of graph-based financial fraud detection. The findings are proposed to be used to inform a more informed evaluation of the trade-offs between classification performance and interpretability. Instead of assuming that a specific architecture is best overall, the study compares the models within the same experimental context and emphasizes the necessity of tackling both predictive effectiveness and transparency in designing a fraud detection system based on a GNN.

7.2. Limitations

There are a number of limitations in this study. For one, the assessment is carried out on the Elliptic Bitcoin transaction data, representing a certain monetary area. Therefore, the findings might not be directly applicable to other transaction networks, payment platforms, or financial fraud situations. Second, the data is class imbalanced and it has transactions with unknown classes that can affect the training, evaluation of the data, and the understanding of fraud detection performance. Various evaluation criteria are employed but the outcomes are always subject to the selection of the data set, data processing choices, and experimental settings. In addition, the comparison in this work is done by choosing only some specific GNN architectures and the ensemble of the GraphSAGE–GAT is only implemented. The results may vary depending on the model configuration, hyperparameters, ensemble method, or classification threshold. GNNExplainer also does not offer a definitive causal proof, but rather offers local, model-specific explanations; the features and edges identified should therefore be treated as being influential for a prediction, and not as evidence of a transaction being fraudulent. When extrapolating these results and/or using the proposed framework outside of the experimental context, the limitations need to be taken into account.

7.3 Future Work

Several directions can extend this study.

1) Evaluation on additional datasets: This framework can be evaluated on other datasets, such experiments would help assess the generalizability of the models beyond the Elliptic Bitcoin transaction network.

2) Temporal graph learning: Elliptic dataset is divided into time steps which provides an opportunity to investigate models that explicitly represent changes in transaction relationships and behavior over time.

3) Improved fraud-class detection: Further experiments can investigate class-imbalance handling, loss functions, sampling strategies, and decision-threshold selection.

4) Ensemble optimization: The GraphSAGE–GAT ensemble can be studied further by evaluating alternative probability-combination strategies and validation-based weight selection.

5) Explanation quality and stability: Future work can examine stability of GNNExplainer outputs across different target transactions, model architectures, and explanation configurations.

6) Broader application integration: Additional functionality could include clearer transaction-level reporting, explanation comparisons, and structured summaries that help users examine model outputs without treating them as definitely fraud.

7.4 Final Remarks

Explainable financial fraud detection requires attention to both predictive performance and the interpretation of model decisions. This study establishes a framework for comparing GCN, GraphSAGE, GAT, and a GraphSAGE–GAT probability ensemble while incorporating GNNExplainer to examine individual predictions.

The framework provides a foundation for investigating how graph neural networks can support transaction classification and how their predictions can be examined through local explanations. Its practical significance ultimately depends on verified experimental findings, appropriate evaluation, and careful interpretation of explanation outputs. Further work on temporal modeling, cross-dataset evaluation, explanation quality, and application integration can help develop the framework into a more comprehensive approach to explainable financial fraud detection.

8. References

[1] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," in *Proc. ACM SIGKDD Workshop on Anomaly Detection in Finance*, 2019.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2017.

[3] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[5] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[6] S. Motie and B. Raahemi, "Financial fraud detection using graph neural networks: A systematic review," *Expert Systems with Applications*, vol. 240, Art. no. 122156, 2024, doi: 10.1016/j.eswa.2023.122156.

[7] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining and Knowledge Discovery*, vol. 29, pp. 626–688, 2015, doi: 10.1007/s10618-014-0365-y.

[8] "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decision Support Systems*, 2020, doi: 10.1016/j.dss.2020.113303.

[9] G. Tong and J. Shen, "Financial transaction fraud detector based on imbalance learning and graph neural network," *Applied Soft Computing*, vol. 149, Art. no. 110984, 2023, doi: 10.1016/j.asoc.2023.110984.

[10] D. Saldaña-Ulloa, G. De Ita Luna, and J. R. Marcial-Romero, "A temporal graph network algorithm for detecting fraudulent transactions on online payment platforms," *Algorithms*, vol. 17, no. 12, Art. no. 552, 2024, doi: 10.3390/a17120552.

[11] D. Cheng, Y. Zou, S. Xiang, and C. Jiang, "Graph neural networks for financial fraud detection: A review," *Frontiers of Computer Science*, vol. 19, no. 9, Art. no. 199609, 2025, doi: 10.1007/s11704-024-40474-y.

[12] Z. Shams Khoozani, M. Seera, and C. P. Lim, "Towards transparent financial AI: A systematic review of graph learning and explainable methods for credit risk and fraud detection," *Artificial Intelligence Review*, vol. 59, Art. no. 246, 2026, doi: 10.1007/s10462-026-11645-z.