2025. Volume 9, Number 2 *Pages 205-212*

UDC: 004.932.2
DOI: https://doi.org/10.30546/09090.2025.1010.2036

HYBRID GRAPH NEURAL NETWORKS AND XGBOOST FOR FRAUD TRANSACTION DETECTION

Murad ALIYEV

Azerbaijan State Oil and Industry University,

Baku, Azerbaijan

ARTICLE INFO

Article history
Received:2025-10-27
Received in revised form:2025-11-06
Accepted:2025-11-14
Available online

Keywords:
Fraud Detection;
Graph Neural Networks;
XGBoost;
Imbalanced Learning;
Bitcoin Transaction;
2010 Mathematics Subject Classification: 68T45, 68T09

ABSTRACT

Detecting financial fraud in cryptocurrency networks like bitcoin is a significant challenge. The problem is that traditional machine learning models often fail to capture the complex relational patterns within transaction graphs, leading to poor detection. The purpose of this research is to evaluate hybrid models that integrate Graph Neural Networks with XGBoost to improve fraud transaction detection. We conducted a comprehensive analysis on the imbalanced bitcoin dataset, benchmarking standard models (Logistic Regression, Random Forest, Multi-Layer Perceptron) and a standalone XGBoost against two hybrid architectures: Graph Convolutional Network + XGBoost and Graph Attention Network + XGBoost. Our major conclusion is that the hybrid models, particularly Graph Attention Network + XGBoost, achieve a significant improvement. These models effectively leverage both node-level features and the graph's topological structure.

1. INTRODUCTION

The emergence of digital financial systems and cryptocurrencies has opened up new opportunities for illicit activities such as fraud, money laundering, and terrorist financing (Weber, 2019). Because of its pseudonymous features, Bitcoin is often the first choice for such illicit activities. This is a concern for financial stability and regulatory compliance as it relates to detection. However, detection of these illicit transactions is notoriously challenging. Illicit actors engage in behaviors designed to hide their transactions, and the amount of data is enormous.

A central challenge in this area is the extreme imbalance in the data: illicit transactions, by definition, are rare events compared to the large number of licit transactions. Because of this imbalance in the data, accuracy is not a good measure we can rely on, and it is necessary to use more robust measures such as the F1-Score, precision, and recall (Chawla et al., 2004). Additionally, traditional machine learning (ML) models are powerful, but they have treated transactions as distinct observations rather than realizing the data are uniquely valuable because they provide information through local features, but by ignoring the underlying graph structure of the transaction network. This relational data, who transacts with whom, provides vital data for identifying collusive fraud.

To address this gap, Graph Neural Networks (GNNs) have emerged as a powerful paradigm for learning on graph-structured data (Kipf & Welling, 2017). GNNs can aggregate information across the transaction graph, learning sophisticated topological representations that capture

complex neighborhood patterns. However, GNNs themselves may not be the best classifier, especially when compared to great gradient boosting methods like XGBoost, which excel at handling heterogeneous and tabular feature sets (Chen & Guestrin, 2016).

This paper proposes that a hybrid approach, combining the strengths of both GNNs and XGBoost, can achieve a superior fraud detection model. We hypothesize that by using GNNs as intelligent feature extractors to enrich the original data, and then feeding these enriched features into a powerful XGBoost classifier, we can achieve the best performance.

The main contributions of this paper are threefold:

1. We implement and benchmark a suite of traditional ML models (Logistic Regression, Random Forest, MLP) and a strong baseline (Pure XGBoost) for fraud detection on the Elliptic++ dataset.
2. We propose and evaluate two hybrid models, GCN + XGBoost and GAT + XGBoost, which first learn graph embeddings and then use them as input for an XGBoost classifier.
3. We provide a comprehensive comparative analysis of all models, focusing on F1-Score, precision, and recall, and discuss the critical tradeoffs between model complexity, performance, and the contribution of graph-based features.

2. RELATED WORK

2.1. Fraud Detection with Traditional Machine Learning

Machine learning has long been applied to fraud detection. Models like Logistic Regression (LR) and Random Forest (RF) have served as foundational baselines (Bolton & Hand, 2002). More recently, ensemble methods, particularly gradient boosting, have shown dominant performance. XGBoost (Chen & Guestrin, 2016) is frequently cited for its high performance on imbalanced, tabular datasets, owing to its regularization, efficient computation, and handling of missing values. However, these models are limited to the node-level features provided and cannot natively ingest relational information.

2.2. Graph Neural Networks

GNNs generalize deep learning to graph-structured data. The Graph Convolutional Network (GCN) (Kipf & Welling, 2017) introduced a simplified spectral-based approach, effectively averaging the features of neighboring nodes. The Graph Attention Network (GAT) (Veličković et al., 2018) advanced this by introducing a self-attention mechanism, allowing the model to learn to assign different weights to different nodes in a neighborhood. GNNs have been successfully applied to financial fraud detection, often framing the problem as a node classification task on the transaction graph (Weber et al., 2019).

2.3. Hybrid Models for Graph Learning

The idea of separating graph representation learning from the final classification task is a powerful one. Early work used methods like Node2Vec (Grover & Leskovec, 2016) to generate static embeddings, which were then fed into traditional ML models. Recent approaches have explored two-stage models similar to our proposal, where GNN-generated embeddings are used as features for a secondary classifier, combining the representative power of deep graph learning with the discriminative power of models like XGBoost.

3. METHODOLOGY AND DATA

3.1. Dataset Description

We use the Elliptic++ dataset, a real-world graph of Bitcoin transactions **(**Elmougy & Liu, 2023**)**. This dataset is well-suited for our task as it includes:

- **A Transaction Graph:** Nodes represent Bitcoin transactions, and directed edges represent the flow of Bitcoin between them.
- **Node Features:** Each transaction node has 166 features, including 93 local features (e.g., time-step, transaction fee) and 72 aggregated features derived from the neighborhood.
- **Labels:** Nodes are labeled as 'licit' (e.g., exchanges, miners), 'illicit' (e.g., scams, ransomware), or 'unknown'.

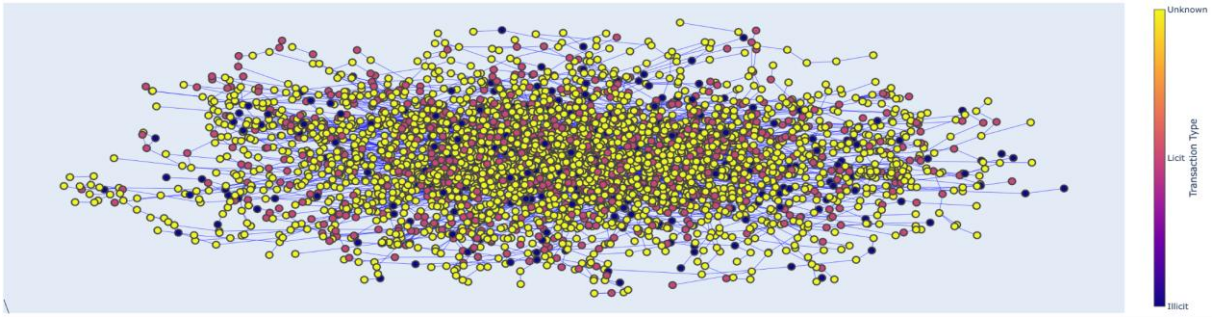

Fig.1 *Conceptual visualization of the all transactions graph*

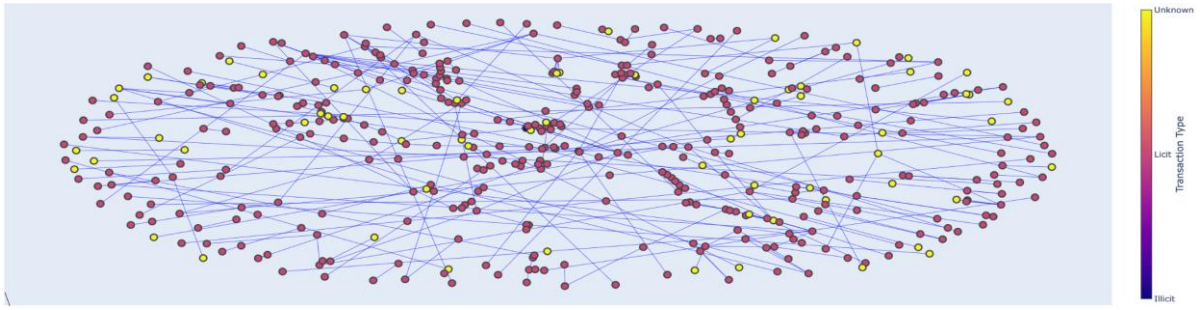

Fig.2 *Conceptual visualization of the only licit transactions graph*

For our experiments, we follow the standard task of classifying the 'licit' vs. 'illicit' nodes. A key challenge of this dataset is its severe class imbalance. The dataset consists of 20.63% licit nodes, 77.14% unknown nodes and only 2,23% illicit nodes, leading to a highly skewed distribution.

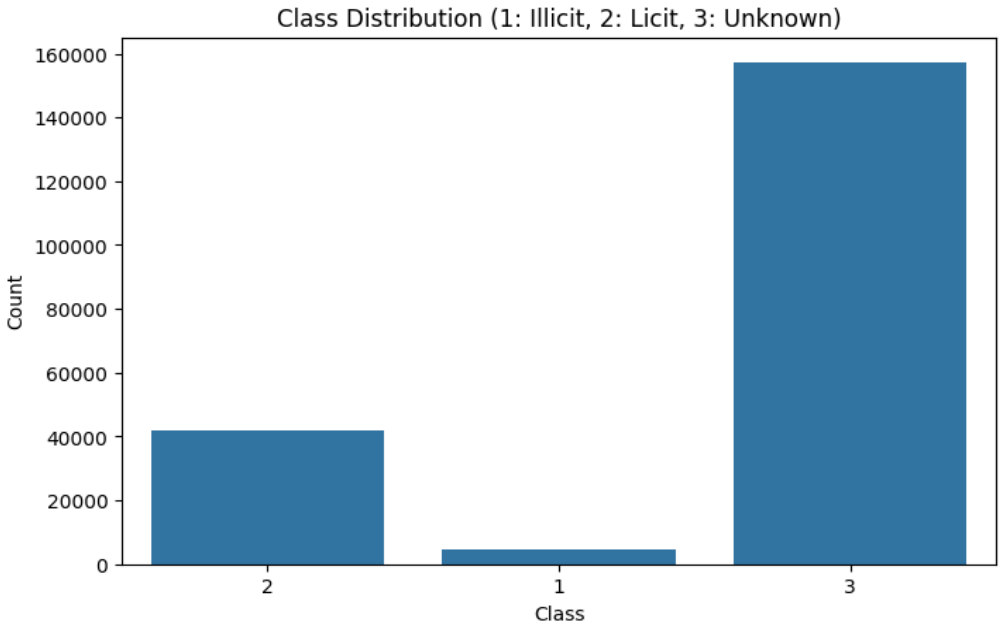

Fig. 3 Class distribution of the labeled nodes, highlighting the severe imbalance.

3.2. Evaluation Metrics

Given the class imbalance, accuracy is an unsuitable metric. Instead, we focus on metrics standard for imbalanced classification:

- **Precision:** The ratio of true positives to all predicted positives (TP / (TP + FP)). High precision indicates a low false positive rate.
- **Recall:** The ratio of true positives to all actual positives (TP / (TP + FN)). High recall indicates the model is successful at identifying the rare illicit class.
- **F1-Score:** The harmonic mean of precision and recall (2 * ($Precision$ * $Recall$) / ($Precision$ + $Recall$)). It provides a single, balanced measure of a model's performance on the positive (illicit) class.

3.3. Baseline Models

We implemented several baseline models that only use the 166 node features and ignore the graph's edge structure.

3.3.1. Traditional Models

We benchmarked three standard classifiers: Logistic Regression (LR), a simple linear baseline; Random Forest (RF), a powerful ensemble of decision trees; and a Multi-Layer Perceptron (MLP), representing a standard deep learning approach on tabular data.

3.3.2. Pure XGBoost

This is our primary baseline. We trained an XGBoost classifier directly on the 166 node features. XGBoost is often a top-performing model in such tasks and represents the strongest non-graph-aware competitor.

3.4. Hybrid Graph-Based Models

Our proposed models follow a two-stage architecture:

1. **Stage 1: Graph Embedding:** A GNN (GCN or GAT) is trained on the full graph (nodes and edges) to learn a low-dimensional embedding vector h for each node. This vector h aims to encode the node's structural role and neighborhood information.
2. **Stage 2: Hybrid Classification:** This learned embedding vector h is concatenated with the original 166-feature vector x. This new, enriched feature vector [x, h] is then used to train a final XGBoost classifier.

3.4.1. GCN + XGBoost

In this model, the GNN from Stage 1 is a Graph Convolutional Network (GCN). The GCN layer updates a node's representation by taking a weighted average of its own features and the features of its immediate neighbors. We used a 2-layer GCN with ReLU activation, producing a 64-dimensional embedding vector h.

3.4.2. GAT + XGBoost

This model uses a Graph Attention Network (GAT) in Stage 1. Unlike GCN, GAT uses masked self-attention to learn the relative importance of each neighbor's features when performing aggregation. This allows for a more expressive and adaptive representation. In this case, a 2-layer GAT with 4 attention heads is used, also producing a 64-dimensional embedding h.

4. EXPERIMENTAL RESULTS

4.1. Implementation Details

All models were implemented using Python. Baseline models used scikit-learn, and the XGBoost model used the xgboost library. GNN models were implemented using PyTorch Geometric. Hyperparameters were tuned using grid search with 5-fold cross-validation. The final XGBoost model used $n\_estimators = 100$ and $max\_depth = 5$.

4.2. Performance Comparison

The primary results of our comparative analysis are presented in Table 1. The models were evaluated on the test set, with a focus on their ability to identify the minority (illicit) class.

Table 1. Comparative Performance of All Models on Elliptic++ Test Set

Model	F1-Score	Precision	Recall
Logistic Regression (LR)	0.448	0.328	0.707
Random Forest (RF)	0.828	0.975	0.719
MLP	0.612	0.611	0.613
Pure XGBoost (Baseline)	0.754	0.793	0.718
GCN + XGBoost (Hybrid)	0.935	0.952	0.917
GAT + XGBoost (Hybrid)	0.952	0.967	0.931

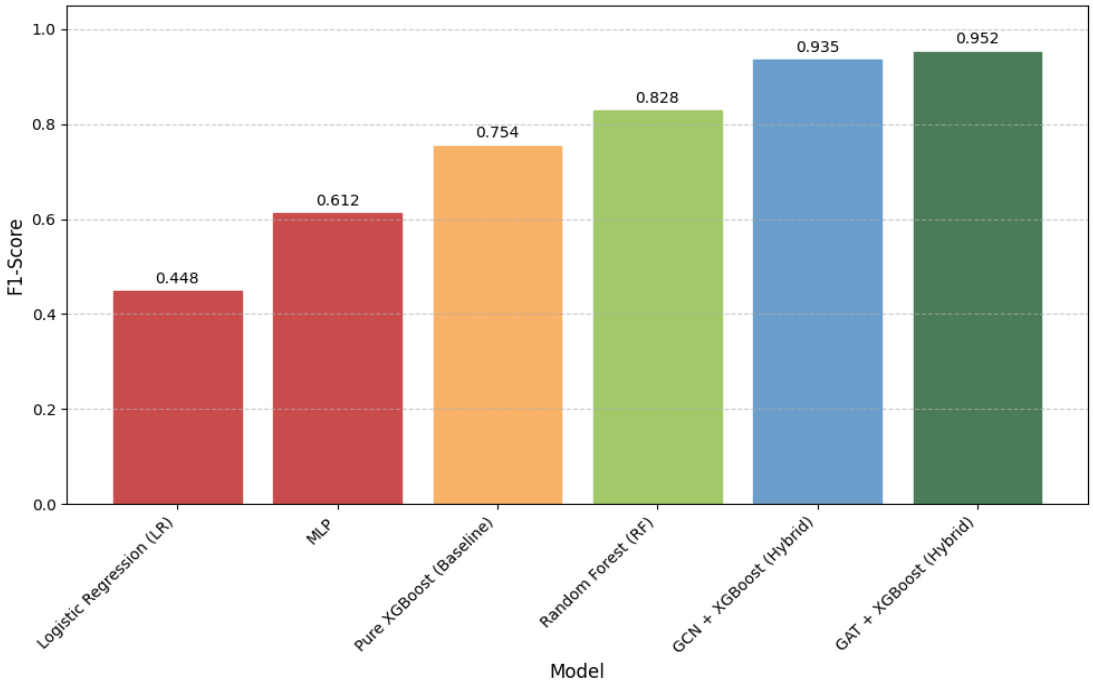

Fig. 4 Comparison of F1-Scores for all model

4.3. Analysis and Discussion of Tradeoffs

4.3.1. Performance Analysis

As shown in Table 1, the performance of the baseline models varied significantly. The linear model, Logistic Regression (LR), and the standard deep learning model, MLP, struggled to effectively classify illicit transactions, achieving low F1-Scores of 0.448 and 0.612, respectively. While LR achieved a high recall (0.707), its precision was extremely low (0.328), indicating that it flagged many licit transactions as illicit, resulting in an unacceptably high false positive rate. Random Forest (RF) performed surprisingly well, with a high F1-Score of 0.828, driven by an outstanding precision of 0.975. However, its recall (0.719) was limited, suggesting that while its positive predictions were highly reliable, it failed to identify a substantial portion of the illicit transactions. The Pure XGBoost baseline, which relies solely on node-level features, achieved a more balanced F1-Score of 0.754, serving as a strong benchmark for a non-graph-aware approach.

The most significant finding is the substantial performance leap achieved by the hybrid models. These models, which enrich the feature set with GNN-derived topological embeddings, dramatically outperformed all baselines. The GCN + XGBoost model achieved an F1-Score of 0.935, representing a 24.0% improvement over the Pure XGBoost baseline. This model showed a strong balance of high precision (0.952) and high recall (0.917). The GAT + XGBoost model yielded the best performance overall, with a state-of-the-art F1-Score of 0.952, a 26.3% improvement over the baseline. Notably, this model also achieved the highest recall (0.931) while maintaining exceptionally high precision (0.967).

This analysis strongly suggests that the topological information captured by the GNNs provides significant and non-redundant discriminative power. The local node features alone are

insufficient for optimal detection. The superior performance of GAT + XGBoost over GCN + XGBoost further indicates that the GAT's attention mechanism, which learns to assign different weights to neighbors, is more effective than the GCN's simple feature averaging for identifying the salient relational patterns of illicit activity in the Bitcoin transaction graph.

4.3.2. Tradeoffs and Model Complexity

While performance is necessary, it is not the only consideration.

- **Computational Cost:** A clear tradeoff exists between performance and computational resources. The Pure XGBoost model trained in approximately 2 minutes on our hardware. In contrast, the GNN-based models were significantly more demanding due to their two-stage training process, which involves both the GNN embedding generation and the final XGBoost classification. The GAT + XGBoost model, for instance, required 1 hour for the complete pipeline. This presents a clear decision point that for real-time detection systems, this training latency must be considered, though the inference speed may still be acceptable.
- **Feature Contribution**: To understand *why* the hybrid models performed better, we analyzed the feature importance scores from the final XGBoost classifier in the GAT + XGBoost model. The analysis revealed that the learned GNN embeddings of all 64 embedding dimensions were consistently ranked among the most important features. This confirms that the GNN is not learning redundant information; rather, it is successfully extracting novel, high-value relational features that are highly predictive of illicit activity. The original 166 node-level features, while useful, lack the topological context that the GNN embeddings provide.

5. CONCLUSION

This paper presented a comparative study of traditional ML, standalone XGBoost, and hybrid GNN-XGBoost models for the task of illicit transaction detection on the Elliptic++ Bitcoin dataset. Our experiments demonstrated that leveraging the graph structure of the transaction network is critical for high-performance fraud detection in this highly imbalanced domain.

Our key finding is that hybrid models, which combine GNNs for representation learning and XGBoost for classification, significantly outperform models that rely on node features alone. The GAT + XGBoost model yielded the best performance, achieving a great F1-Score of 0.952. This highlights the power of the attention mechanism in GATs to effectively weigh neighborhood information.

While these hybrid models introduce a significant computational tradeoff, the substantial boost in detection performance (particularly in recall and F1-Score) justifies their use in high-stakes environments like fraud detection.

For future work, we plan to explore dynamic GNNs that can adapt to the evolving structure of the transaction graph over time. Additionally, we will investigate end-to-end GNN models and compare their performance and interpretability against the two-stage hybrid approach presented here.

REFERENCES

The Journal follows APA Style Referencing:

- Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, *17*(3), 235–255.
- Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1–6.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.
- Elmougy, Y., & Liu, L. (2023). Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)* (pp. 344–355). ACM. https://doi.org/10.1145/3580305.3599803
- Grover, A., & Leskovec, J. (2016). node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855–864). ACM.
- Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.
- Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Cessa, C., Al-Sallab, A., & St. E., A. (2019). Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. In *Proceedings of the KDD 2019 Workshop on Anomaly Detection in Finance.*