Classifying Illicit Bitcoin Transactions using Graph Neural Networks

Xuhui Zhan, Siyu Yang, Tianhao Qu
Data Science Institute, Vanderbilt University

Abstract

This paper addresses the challenge of detecting illicit activity in Bitcoin transaction networks. We use the Elliptic dataset containing Bitcoin transactions labeled as licit, illicit, or unknown. We apply Graph Neural Network (GNN) models to classify unlabeled nodes in this transaction graph. Our focus is not only on achieving high accuracy but also on preserving critical structural properties of the transaction networks. We compare different GNN models including GraphSAGE and Graph Attention Networks (GAT) on both classification performance and graph structure preservation. Our results show that models which better preserve the structural properties of legitimate transaction networks achieve significantly higher test accuracy. Specifically, we find that GraphSAGE's neighborhood sampling approach maintains critical transaction patterns better than GAT's attention mechanism. This study demonstrates the importance of preserving authentic network topology for reliable fraud detection in cryptocurrency networks.

Introduction

Bitcoin and other cryptocurrencies have become popular for both legitimate financial activities and illicit transactions. Detecting illegal activities on cryptocurrency blockchains is challenging because of the pseudonymous nature of transactions and the large volume of data. Traditional methods often fail to capture the complex relationships between transactions.

In this project, we use Graph Neural Networks (GNNs) to classify Bitcoin transactions as licit or illicit. We focus on both the accuracy of classification and how well the models preserve important structural properties of the transaction network. This is important because models that distort the network structure might not generalize well to real-world applications.

The project uses data from Elliptic, a blockchain analytics company. The dataset includes 203,769 Bitcoin transactions (nodes) and 234,355 transaction flows (edges). Each transaction is labeled as licit (20.6%), illicit (2.2%), or unknown (77.2%). Each node has 166 features describing transaction characteristics.

Our main contributions are:

- Implementation and comparison of different GNN models for Bitcoin transaction classification
- Analysis of how well different models preserve important graph properties
- Introduction of a new metric called "distance-to-ideal" for evaluating model reliability
- Evidence that structural preservation correlates with classification performance

Data and Preprocessing

The Elliptic Dataset

The dataset represents Bitcoin transactions over 49 time steps. Each node is a transaction, and edges represent Bitcoin flows between addresses. The 166 features per node include:

- Transaction-level statistics (time, amount)
- Aggregated metrics over time (degree, centrality)

The class distribution shows a significant imbalance:

- Licit: 20.6% - Legitimate transactions not associated with criminal activity
- Illicit: 2.2% - Transactions linked to illegal activity
- Unknown: 77.2% - Transactions in the dataset but not labeled

Preprocessing Steps

We performed the following preprocessing steps:

- Extracted node features and edge lists from different CSV files
- Mapped transaction IDs to indices to make data compatible with PyTorch Geometric
- Separated the graph into different sets: Known (Train, Val, Test) and Unknown

Exploratory Analysis

Our exploratory data analysis revealed several important graph properties:

- **Degree distribution.** Both in-degree and out-degree exhibit a power-law tail—typical of financial networks (see Fig. 1).
- Community Detection: Using the Louvain algorithm, we identified transaction clusters that might indicate coordinated behavior
- Network Structure: One dominant connected component contains the majority of labeled transactions

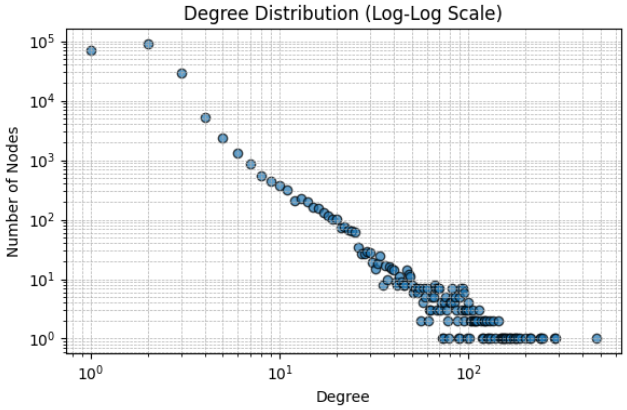

Figure 1: Degree distribution plotted on a log–log scale. The heavy-tailed pattern (few hubs, many low-degree nodes) is consistent with a scale-free transaction network.

Methodology

Our methodology consists of three main parts:

1. Direct classification with base models
2. Data augmentation with base models or label propagation
3. Graph property preservation analysis

Base Models

We implemented two GNN architectures:

- GraphSAGE: Uses neighborhood sampling to aggregate information from nearby nodes
- GAT (Graph Attention Network): Uses attention mechanisms to weight the importance of neighboring nodes

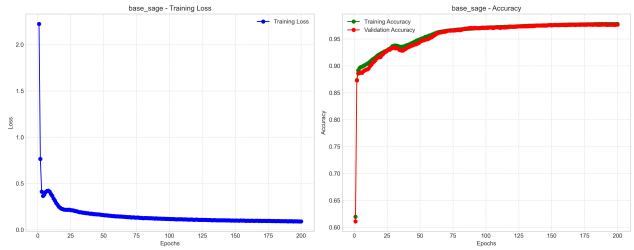

Figure 2: Training–loss (left) and accuracy (right) curves for the `base_sage` model over 200 epochs.

Both models were trained on the original training set and used to predict labels for all unknown nodes.

Data Augmentation

We created augmented training sets by adding 30% of unknown nodes with predicted labels to the original training set. We used three augmentation strategies:

- Label Propagation Augmentation: Using label propagation algorithm
- GraphSAGE-based Augmentation: Using labels predicted by GraphSAGE
- GAT-based Augmentation: Using labels predicted by GAT

For each augmentation strategy, we trained both GraphSAGE and GAT models on the augmented data, resulting in a total of 8 models:

- base_sage and base_gat
- label_propagation_sage and label_propagation_gat
- sage_sage and sage_gat
- gat_sage and gat_gat

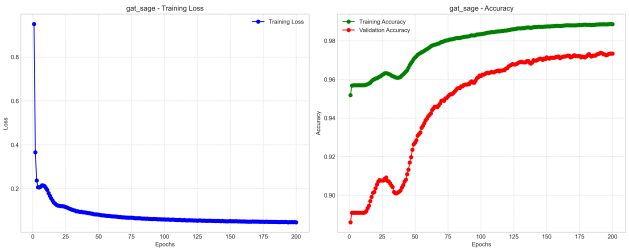

Figure 3: Training–loss (left) and accuracy (right) curves for the `gat_sage` model over 200 epochs.

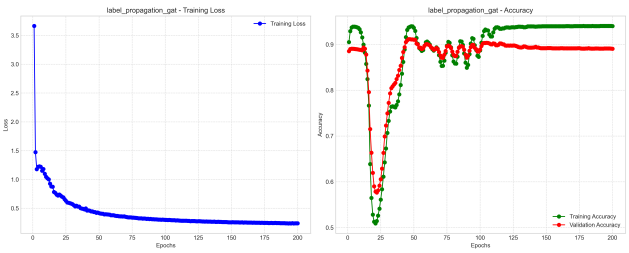

Figure 4: Training–loss (left) and accuracy (right) curves for the `label_propagation_gat` pipeline over 200 epochs.

Graph-Property Preservation Analysis

A robust detector must *not only* label individual nodes correctly, *but also* reproduce the structural "fingerprints" of licit and illicit transaction networks. To quantify, for each model, **how far its predicted subgraphs stray from the training-set profile**, we use the three-step procedure as described below.

1. **Compute six structural metrics for every licit & illicit subgraph.**

Homophily
Propensity for same-class links, estimated by the *edge-to-node* ratio within each class (higher $\Rightarrow$ denser clusters of similar transactions).

Density
Overall link density $\frac{2|E|}{|V|(|V|-1)}$; low density may signal isolated, covert activity, whereas high density reflects normal market traffic.

Average Clustering Coefficient
Mean local triangle ratio, highlighting how tightly nodes group into communities.

Average Degree Centrality
Mean normalised degree, i.e., how many counterparties a typical node transacts with.

Average Betweenness Centrality
Mean brokerage score that reveals bridge nodes—potential "money-mule" behaviour.

Largest-Component Ratio
Fraction $|V_{\text{LCC}}|/|V|$ of nodes in the giant component, measuring fragmentation.

2. **Build an *ideal* baseline via bootstrap.** We sample 70% of the training nodes $N_{\text{boot}} = 10$ times, recompute the six metrics, and record the bootstrap mean v_{train} and standard deviation σ_{boot} for each metric—capturing the natural, data-driven variance under perfect predictions.

3. **Measure the distance to ideal.** For a predicted metric v_{pred} we first compute

$$S_{\text{raw}} = \begin{cases} 1, & v_{\text{pred}} = v_{\text{train}} = 0, \\ 0, & v_{\text{pred}} v_{\text{train}} = 0, \\ 1 - \min\Big(\frac{|v_{\text{pred}} - v_{\text{train}}|}{\max(|v_{\text{pred}}|, |v_{\text{train}}|)}, 1\Big), & \text{otherwise.} \end{cases} \tag{1}$$

$$S_{\text{norm}} = \begin{cases} 1, & \sigma_{\text{boot}} = 0 \wedge \Delta = 0, \\ 0, & \sigma_{\text{boot}} = 0 \wedge \Delta > 0, \\ \max\Big(0,\, 1 - \min\Big(\frac{\Delta}{3\sigma_{\text{boot}}}, 1\Big)\Big), & \text{otherwise,} \end{cases} \tag{2}$$

where $\Delta = |v_{\text{pred}} - v_{\text{train}}|$. The single-metric *distance to ideal* is $D = |S_{\text{norm}} - 1|$; we average six such D's to obtain $D_{\text{licit}}, D_{\text{illicit}}$ and report $D_{\text{combined}} = \frac{1}{2}(D_{\text{licit}} + D_{\text{illicit}})$.

Interpretation. A perfect classifier gives $D_{\text{combined}} \approx 0$, i.e. its predicted licit/illicit subgraphs are statistically indistinguishable from the real ones after accounting for natural variation. Higher values warn that node-level accuracy may conceal topological artefacts relevant to downstream forensic tasks.

Results

Model Performance

According to Table 1, our evaluation of the eight models showed varying levels of performance:

- Best Test Accuracy: base_sage (97.4%)

Model	Final Training Loss ↓	Validation Accuracy ↑	Test Accuracy ↑
base_gat	0.272	0.921	0.921
base_sage	0.090	**0.977**	**0.974**
label_propagation_gat	0.239	0.912	0.907
label_propagation_sage	0.110	0.964	0.965
gat_gat	0.182	0.906	0.918
gat_sage	**0.047**	0.974	0.970
sage_gat	0.188	0.910	0.916
sage_sage	0.049	0.973	0.970

Table 1: Model Performance: Train, Val, Test

- Best Validation Accuracy: base_sage (97.7%)
- Lowest Training Loss: gat_sage (0.047)

Overall, models using GraphSAGE consistently outperformed GAT-based models in terms of classification accuracy.

Structure Preservation Analysis

According to observe Figure 1, the distance-to-ideal metric revealed interesting patterns:

- Models that better preserved licit network structure achieved significantly higher test accuracy
- We found a strong negative correlation between licit structure preservation and test accuracy (r = -0.99, p = 0.0002)
- There was a moderate correlation for illicit preservation (r = -0.56, p = 0.25)

This suggests that maintaining the structural properties of legitimate transaction networks is particularly important for accurate classification.

Why GraphSAGE Performs Better

GraphSAGE models better preserved graph properties compared to GAT models for several reasons:

- Neighborhood Aggregation: GraphSAGE uses uniform sampling of neighbors, treating all connections equally, which helps maintain network structure
- Parameter Complexity: GAT's attention mechanisms have more parameters that can lead to overfitting and structure distortion
- Structural Stability: GraphSAGE maintains better consistency between properties of known and unknown nodes

These characteristics make GraphSAGE more suitable for financial transaction networks where preserving structural patterns is crucial.

Conclusion

Our research demonstrates that preserving authentic network topology is critical for reliable fraud detection in cryptocurrency networks. The strong correlation between structure preservation and classification accuracy suggests that this should be a key consideration when developing models for financial fraud detection.

Key findings from our work include:

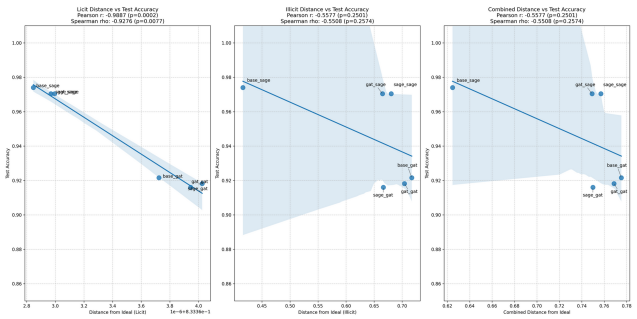

Figure 5: Correlation between distance-to-ideal metrics and test accuracy. Left: licit subgraph; Middle: illicit subgraph; Right: combined distance.

- The proposed distance-to-ideal metric provides a valuable complementary evaluation beyond traditional accuracy measures
- GraphSAGE's neighborhood sampling preserves essential graph properties more effectively than GAT's attention mechanism
- Preserving licit subgraph structure is more indicative of model reliability than preserving illicit patterns

Future work could extend this approach to temporal networks, develop structure-optimized models, improve explainability, and scale to larger cryptocurrency datasets.

References

- Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.E. (2019). Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics.
- Zhu, X., Ghahramani, Z.,& Lafferty, J. (2003). Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, 912–919.
- Hamilton, W., Ying, Z., Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. Neural Information Processing Systems (NIPS).
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y. (2018). Graph Attention Networks. International Conference on Learning Representations (ICLR).
- Elliptic Data Set. (2019). Elliptic Data Set - Bitcoin Transactions. Retrieved from https://www.kaggle.com/ellipticco/elliptic-data-set