RUTGERS
UNIVERSITY

Detecting Illicit Behavior in Crypto Transaction Network

Contributing Authors:
Howard Wang
Yazhe Huang

May 2, 2025

Graph Methods and Network Analysis
22:544:647:01

Abstract

We aim to detect high-risk wallet addresses in cryptocurrency transaction networks using graph-based methods and Graph Neural Networks (GNNs). By constructing a directed Ethereum transaction graph and applying centrality metrics such as Degree Centrality and PageRank, we identified wallets with suspiciously high connectivity. Louvain clustering further revealed dense fraud-like communities. To enhance detection accuracy, we trained a GCN model using the GCNConv architecture on over 20,000 labeled addresses and 9.8 million transaction edges. The model achieved a validation accuracy of **91.2%** and an F1-score of **0.894**, indicating strong performance in classifying scam addresses. Visualization of learned node embeddings using t-SNE and PCA showed clear separation between scam and benign wallets. These results demonstrate that our graph-based learning approach effectively captures subtle structural and behavioral patterns, offering a scalable solution for scam detection in large-scale crypto networks.

Mission Statement and Research Motivation

According to Packshield's 2024 Crypto Security Report, crypto-related scams surged with $3.01B in stolen assets, up 15% from 2023. As cryptocurrency becomes part of everyday life, security improvements are essential. In October 2024, the U.S. Department of Justice indicted three cryptocurrency companies, including Saitama and Gotbit, for coordinated market manipulation schemes. These involved wash trading and pump-and-dump tactics that created misleading price and volume signals, resulting in investor losses and arrests.

These incidents underscore the need for proactive fraud detection strategies. Our study leverages graph theory and network analysis to uncover behavioral and structural patterns in transaction data, such as mixer activity and cyclical flows, aiming to better detect and prevent similar incidents in the future.

Dataset Description

Our analysis leverages multiple datasets related to Ethereum blockchain activity, combining both labeled and unlabeled transactional data to enable fraud detection via graph-based methods.

The core transaction dataset (`transaction_dataset.csv`) contains historical Ethereum transfers with fields such as `TxHash`, `From`, `To`, `Value`, `Timestamp`, and `isError`. This dataset serves as the basis for constructing our transaction network, where each wallet address is treated as a node and each transaction as a directed edge.

To enhance structural analysis, we incorporate two preprocessed graph snapshots: `first_order_df.csv` and `second_order_df.csv`, which capture transaction paths of different hop depths for richer contextual understanding.

The `addresses.csv` file provides a comprehensive list of wallet addresses involved in the dataset and includes labels used for machine learning training and evaluation purposes.

Additional behavioral enrichment comes from three key supplementary tables:

- 'TI_B.tsv': Contains behavioral indicators related to suspected scam activities, such as high-frequency transfers or repeated links to flagged nodes.
- 'TI_M.tsv': Aggregates mixer-related behavioral flags (e.g., high outgoing diversity, low incoming activity).
- 'CE.tsv': Includes PageRank scores and centrality measures to highlight influential wallets within the transaction graph.
- 'BE.tsv': Contains 54 behavioral and transactional features for over 16,000 Ethereum wallets, including labels for suspicious activity and detailed metrics on transaction patterns, gas usage, account lifespan, and wallet interaction types.

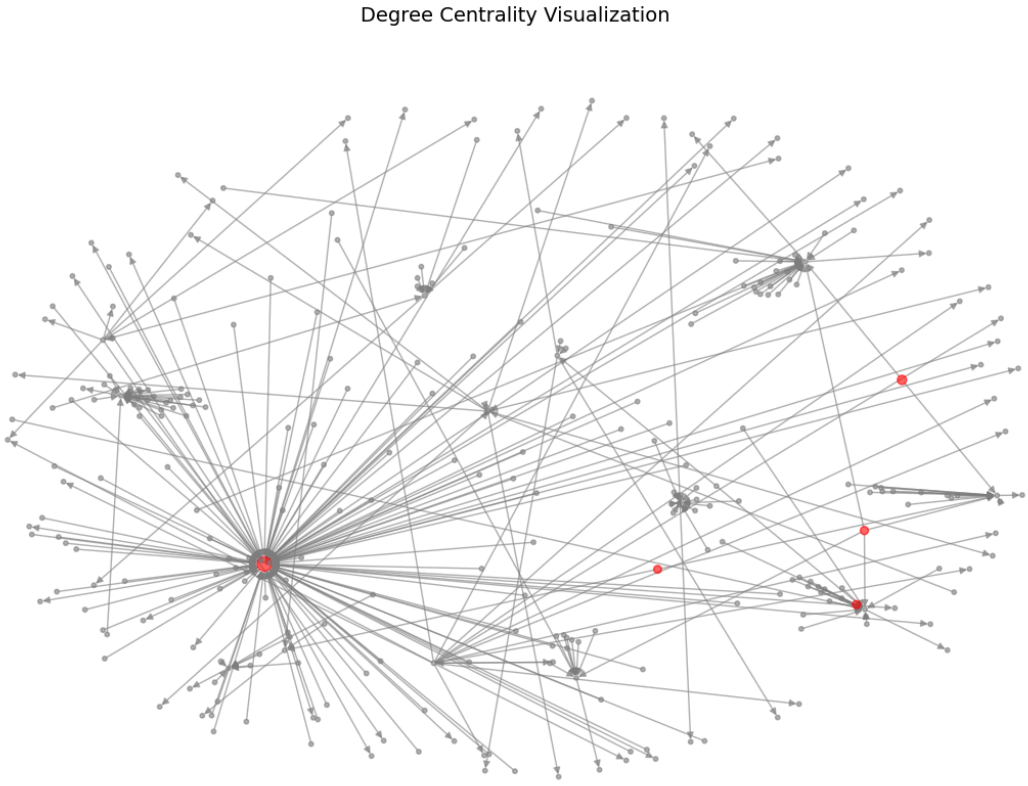
Graph Construction & Network Analysis

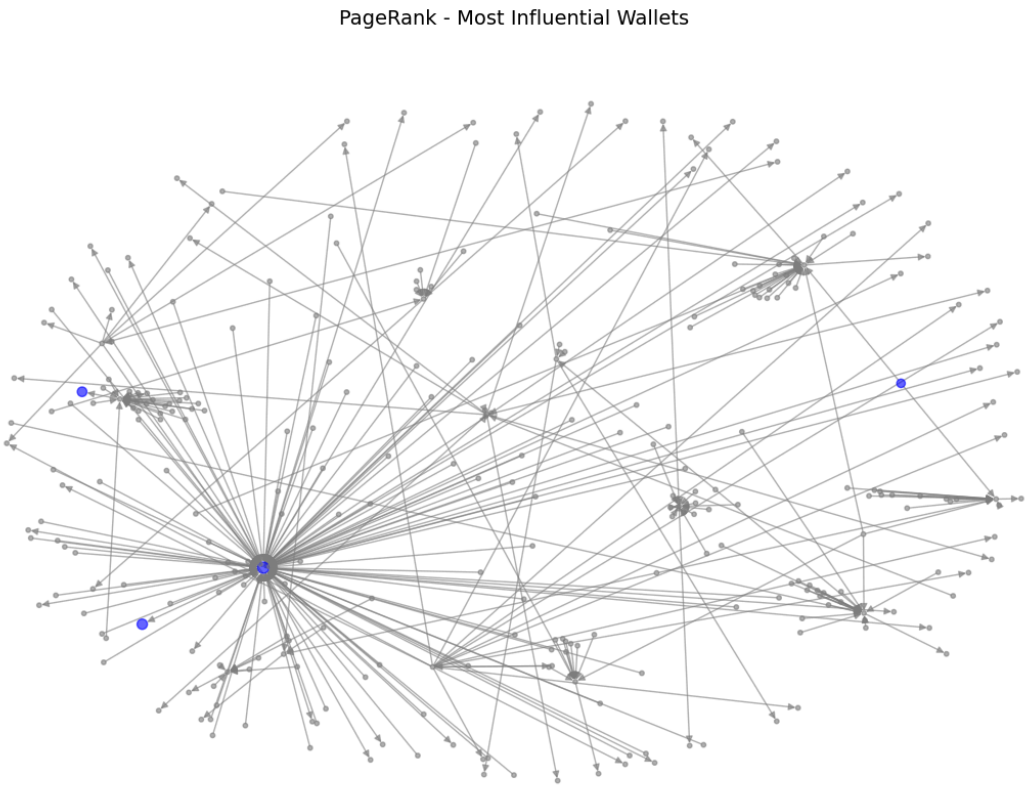
To explore the structure of crypto transactions and detect potentially illicit activities, we first constructed a directed transaction graph. Each node represents a unique wallet address, and each edge denotes a transaction from one address to another, with the transaction value assigned as edge weight. The graph was built using a sampled subset of 10,000 transactions, resulting in **5,689 nodes** and **6,553 edges**.

To identify key actors in the network, we applied **degree centrality** and **PageRank** algorithms:

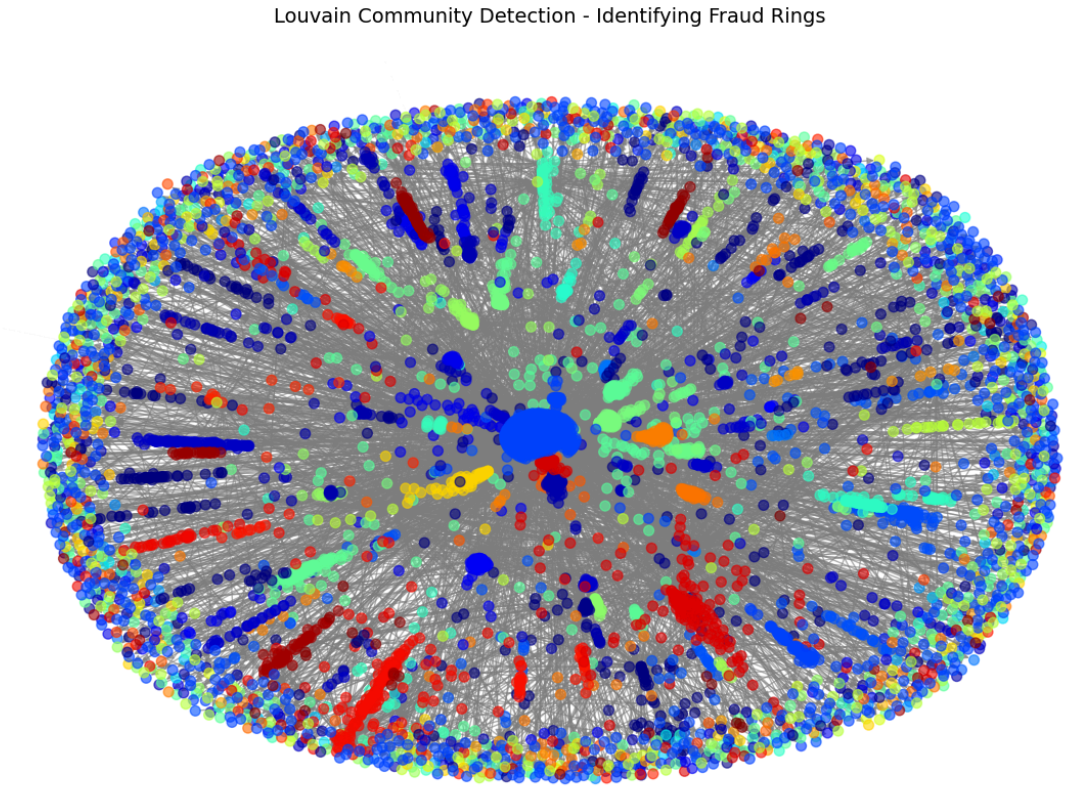
- **Degree Centrality** measures the activity level of each wallet based on its incoming and outgoing connections. High-degree nodes are often indicative of mixers or scam orchestrators.
- **PageRank**, originally developed by Google, evaluates the influence of each node by considering both the quantity and quality of its connections, helping us uncover addresses that play pivotal roles in fund distribution.

We visualized the **top 50 active wallets** based on degree centrality and the **top 50 most influential wallets** by PageRank. Notably, several high-degree wallets also ranked high in PageRank, suggesting strong structural dominance and possible fraud involvement.

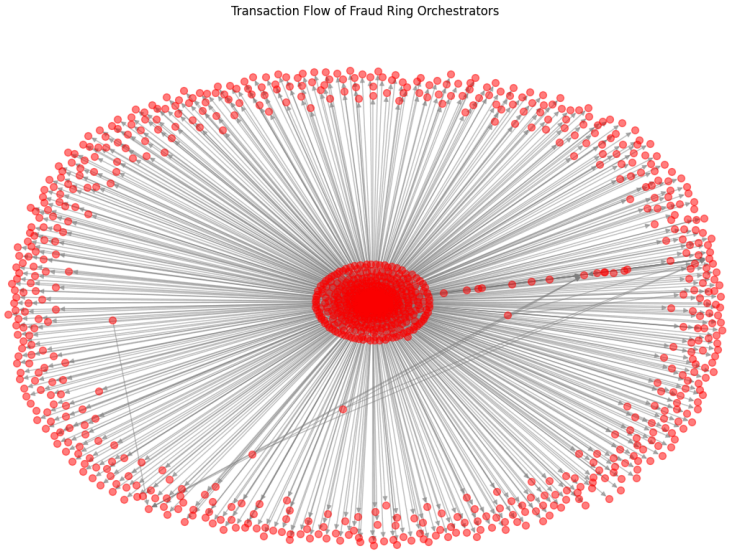




In addition to node-level analysis, we implemented **Louvain community detection** to identify tightly connected subgraphs (communities). This allowed us to localize suspicious fraud rings and visualize their transaction flow separately.

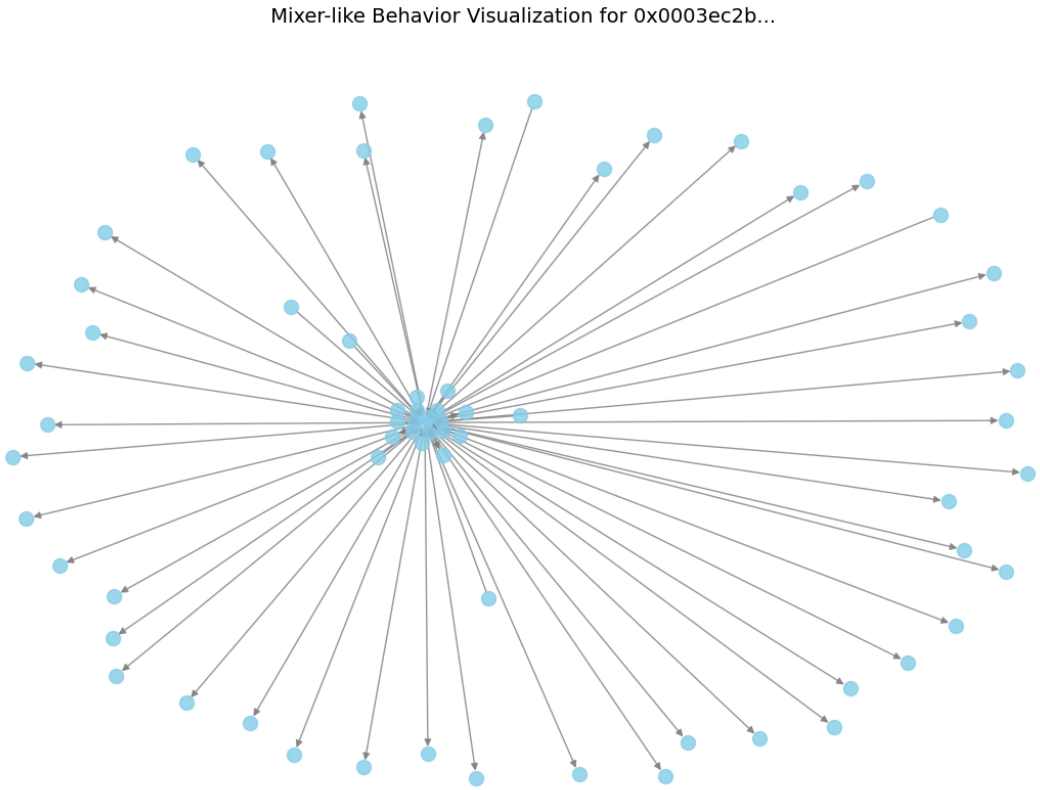


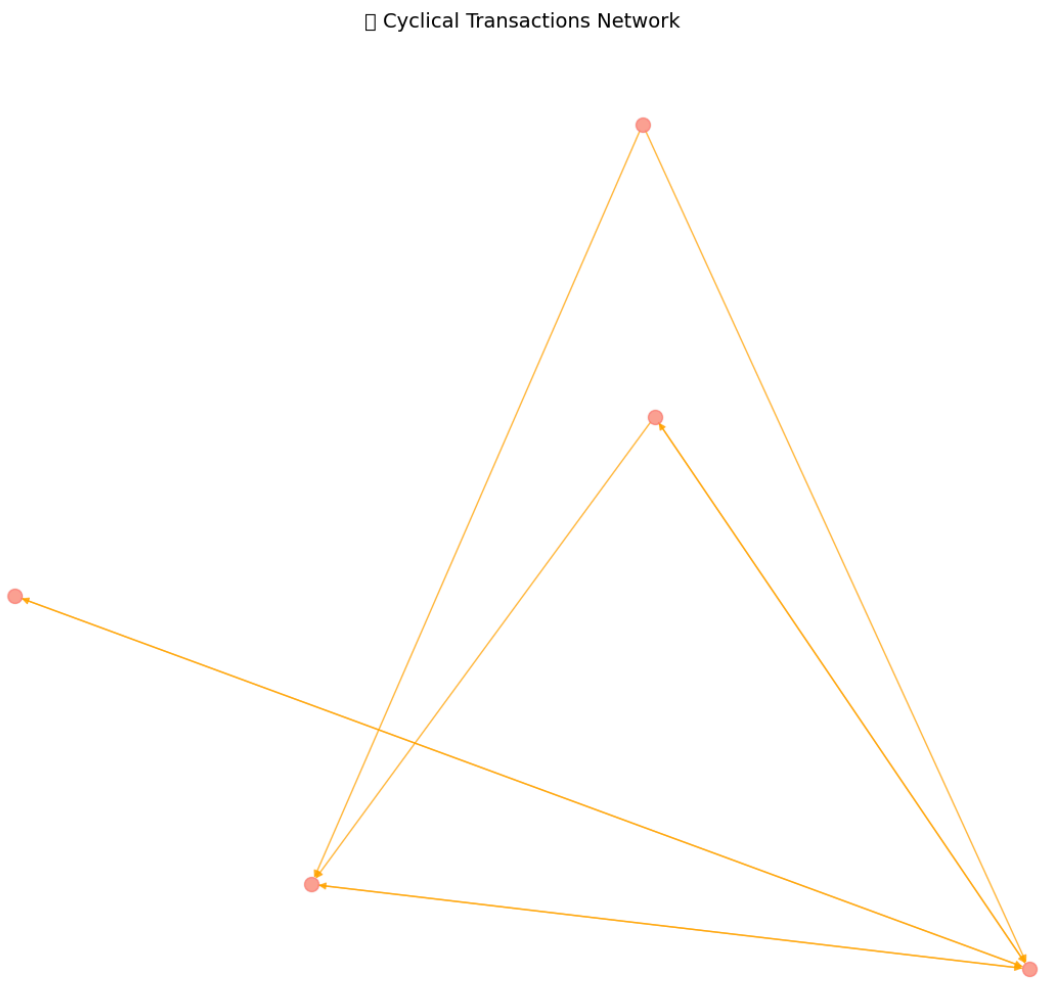
Finally, wallets exhibiting **suspicious behavior patterns**, such as only sending funds without receiving any or being part of a dense fraud cluster, were flagged for further investigation.



Behavior-based Detection

We extended our detection strategy beyond static graph metrics by implementing behavior-based rules to flag suspicious wallets. First, we identified high-activity wallets based on outlier degree values; these wallets often serve as central hubs in suspected fraud rings. Next, we detected "outgoing-only" wallets—addresses that send but never receive funds—potentially indicating cash-out nodes in a laundering sequence. We further visualized **mixer-like patterns**, where wallets received high volumes of small transactions and redistributed them quickly across many addresses. Finally, we flagged **cyclical transaction flows**—closed loops such as A → B → C → A—which are often indicative of obfuscation attempts. These rule-based heuristics served as handcrafted features and prior signals for downstream classification.





The Machine Learning Phase

Overview

To advance the detection of high-risk wallet addresses, we transitioned from rule-based heuristics to a machine learning approach by implementing a Graph Convolutional Network (GCN). Unlike traditional models that treat each address independently, GCNs leverage the underlying transaction network structure to capture relational dependencies between wallet addresses.

The goal was to build a node-level classification model that predicts whether a wallet is associated with scam activities (`label=1`) or is benign (`label=0`).

Dataset & Feature Preparation

We used a curated dataset of approximately 20,000 wallet addresses linked through over 9.8 million token transaction edges. Each node (wallet) was enriched with multiple types of features, including:

- **Behavioral features**:
  - Transaction volume (total ETH sent and received)
  - In-degree and out-degree (number of incoming and outgoing transactions)
  - Centrality scores (e.g., PageRank, Degree Centrality)
  - Behavioral flags (e.g., mixer-like behavior, outgoing-only addresses)
- **Structural features**:
  - Node connectivity patterns extracted from the transaction graph.

By combining behavior-driven and structure-driven features, the model was equipped to learn both individual address behaviors and their broader network roles.

Model Architecture

We implemented a two-layer Graph Convolutional Network (GCN) using the `GCNConv` layers from PyTorch Geometric (`torch_geometric.nn`).

The model architecture included:

- **Input Layer**: Receives node features (dimension = number of input features).
- **Hidden Layer**: One GCN layer with ReLU activation, allowing aggregation of 1-hop neighbor information.
- **Output Layer**: Another GCN layer projecting the hidden representations into a 2-class output (scam / benign).

The architecture can be expressed as:

$$H^{(1)} = ReLU(GCNConv(X, A))$$

$$\widehat{Y} = Softmax\left(GCNConv\left(H^{(1)}, A\right)\right)$$

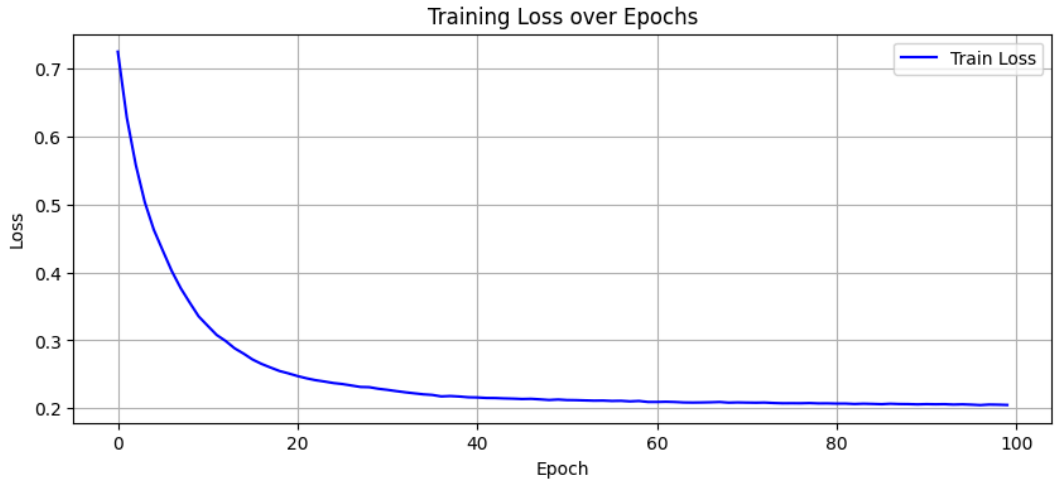
Where:

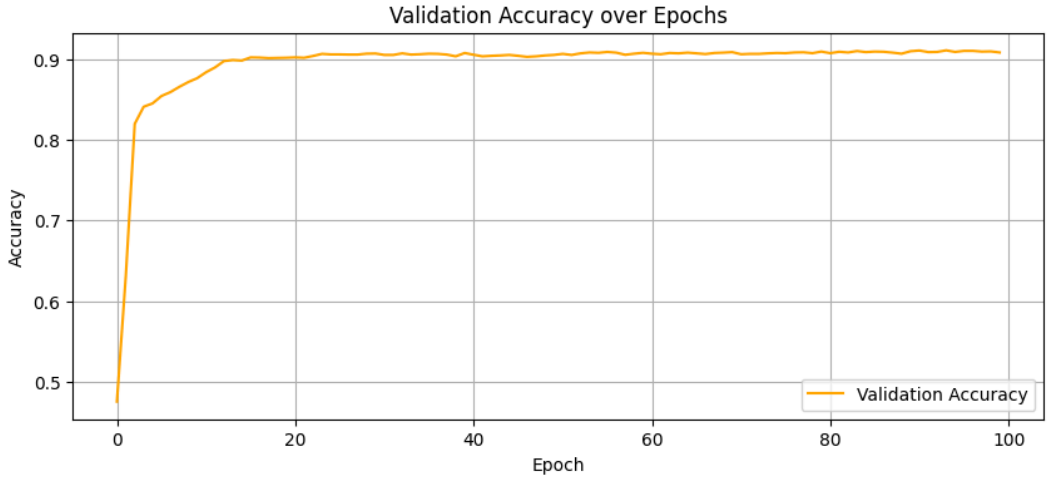
- X = Node features
- A = Adjacency matrix of the graph
- $H^{(1)}$ = Hidden node representations
- $\widehat{Y}$ = Predicted probabilities for each class

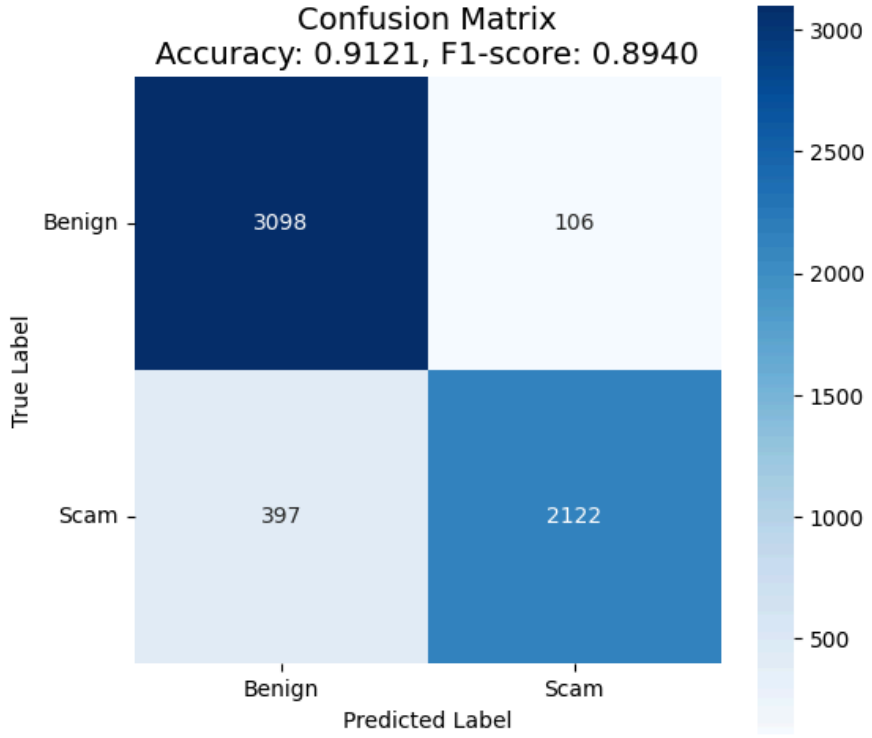
Training Procedure

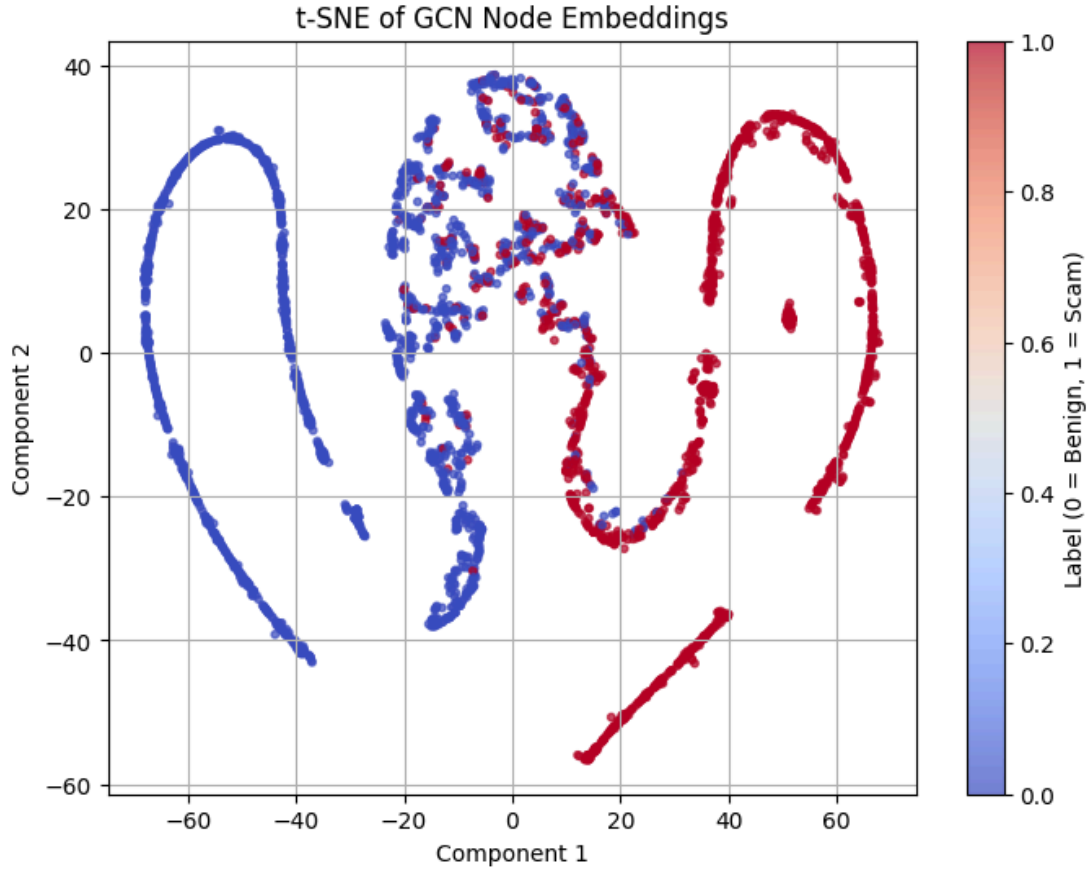
- **Loss function**: Cross-Entropy Loss for binary classification.
- **Optimizer**: Adam optimizer with a learning rate of 0.01
- **Training epochs**: 100 epochs.
- **Device**: Trained using GPU (CUDA) for faster convergence.
- **Validation split**: 80% training / 20% validation.
- **Early stopping**: Monitored validation accuracy and loss trends to avoid overfitting.

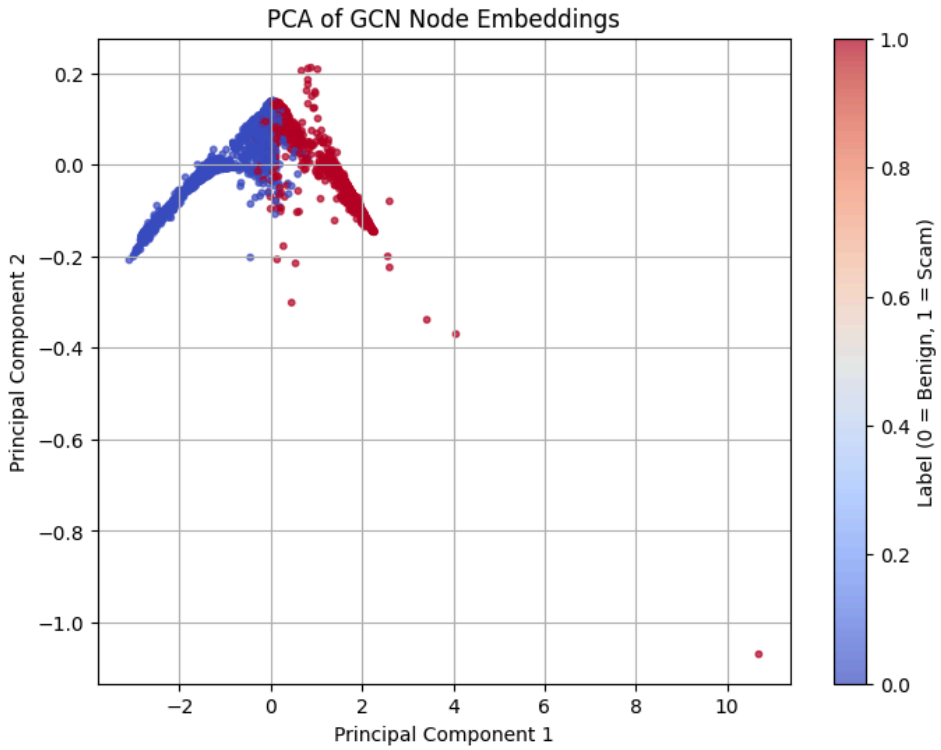
During training, the model leveraged message passing: each node's feature was iteratively updated by aggregating information from its neighbors, enabling learning of both local and global fraud patterns.






Confusion Matrix
Accuracy: 0.9121, F1-score: 0.8940
Benign
Scam
True Label
3098
106
397
2122
Benign
Scam
Predicted Label
3000
2500
2000
1500
1000
500


t-SNE of GCN Node Embeddings
Component 2
Component 1
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60
Label (0 = Benign, 1 = Scam)
1.0
0.8
0.6
0.4
0.2
0.0



Evaluation Metrics

Key performance results achieved were:

- **Validation Accuracy**: **91.21%**
- **F1-Score**: **0.8940**
- **Confusion Matrix**:
  - True Positives (TP): 2,122
  - True Negatives (TN): 3,098
  - False Positives (FP): 106
  - False Negatives (FN): 397

These metrics indicate that the model was effective in identifying scam addresses while minimizing both false alarms and missed detections.

Interpretation of Results

The high validation accuracy and F1-score reflect the model's strong ability to distinguish between fraudulent and benign wallets, even when fraudulent nodes attempt to camouflage their behavior through complex transaction flows.

Further visualization of the learned node embeddings using dimensionality reduction techniques (e.g., t-SNE, PCA) showed clear separation between the two classes, reinforcing the model's effectiveness