ILLICIT TRANSACTION DETECTION IN BITCOIN NETWORKS USING GRAPH ANALYTICS AND GRAPH NEURAL NETWORKS

Ilyas Dahaoui **Ilyasse El Khazane** **Mohammed Adam Khali**

ABSTRACT

The increasing adoption of cryptocurrencies has raised serious concerns regarding their use for illicit activities such as money laundering, fraud, and illegal trade. Bitcoin transactions naturally form a large-scale network where complex dependencies between transactions reveal hidden patterns of suspicious behavior. In this project, we study the problem of illicit transaction detection by modeling 203,769 Bitcoin transactions as a directed graph with 234,355 edges and 169 node features. We first perform an extensive exploratory analysis of the transaction network, examining class imbalance, centrality metrics, K-Core decomposition, and topological signatures to identify structural differences between licit and illicit transactions. A key finding is that illicit transactions exhibit a *structural invisibility* pattern: their HITS hub and authority scores are over 2,400$\times$ lower than licit nodes, and they remain in the outermost shells of the network (K-Core mean $= 1.19$). We then apply community detection algorithms (Louvain and Label Propagation) to uncover suspicious clusters, with the most suspicious community reaching 86.4% illicit concentration. Finally, we formulate illicit detection as a node classification problem and compare six machine learning approaches including a 3-layer Graph Convolutional Network (GCN) and a novel Hybrid GCN+RF model. Random Forest achieves the best F1-score of 0.933, while the Hybrid model (F1 $= 0.922$) demonstrates that graph embeddings carry complementary signal to feature-based classifiers. A critical contribution is the filtering of unknown nodes from graph convolution, which improves GCN recall from near-random to 0.90.

Code available at: https://github.com/Ilyass-Dahaoui/GNN/blob/main/Illicit_Transaction_Detection_in_Bitcoin_Networks_Using_Graph_Analytics_and_Graph_Neural_Networks.ipynb

1 INTRODUCTION AND MOTIVATION

Bitcoin transactions are publicly recorded on a distributed ledger, forming a massive and continuously evolving transaction network. While this transparency enables traceability, the pseudonymous nature of addresses makes illicit activity detection particularly challenging. Traditional anti-money laundering (AML) systems rely on identity verification and centralized records, both fundamentally incompatible with the blockchain's decentralized architecture.

Graph-based modeling offers a natural abstraction for cryptocurrency systems, where transactional dependencies reveal structural patterns invisible to approaches that treat transactions independently. The **Elliptic dataset** (Weber et al., 2019), released by Elliptic Co. in collaboration with MIT, provides one of the only large-scale labeled transaction graphs on a real public blockchain. It contains 203,769 transaction nodes organized into 49 temporal time steps (each $\approx$2 weeks), 234,355 directed edges, and 166 node features per transaction (94 local + 71 aggregated neighborhood features). The dataset is severely imbalanced: among labeled nodes, only 9.8% (4,545 out of 46,564) are illicit, while 77.1% of all nodes are unlabeled.

In this work, we investigate how graph analytics and graph machine learning can be leveraged to detect illicit Bitcoin transactions. We address three research questions: (1) Do illicit transactions exhibit structurally distinct topological signatures? (2) Can community detection isolate clusters of

coordinated fraud? (3) Which classifier best detects illicit transactions, and does graph structure improve performance?

2 Related Work

Fraud detection on blockchain networks. Weber et al. (2019) introduced the Elliptic dataset and established the original benchmark for illicit Bitcoin transaction detection, evaluating Logistic Regression, Random Forest, and a 2-layer GCN. Their GCN achieved an illicit F1-score of approximately 0.70 on a chronological train/test split. Our work extends theirs in three directions: (1) we engineer four additional topological features (PageRank, In-Degree Centrality, HITS, K-Core) and demonstrate their discriminative power; (2) we evaluate six classifiers including a novel Hybrid GCN+RF model; and (3) we introduce unknown-node filtering as a principled technique for handling the 77.1% unlabeled nodes, improving GCN recall from near-random to 0.90.

Graph Convolutional Networks. Kipf & Welling (2017) proposed the Graph Convolutional Network (GCN), which forms the backbone of our deep learning approach. Each GCN layer propagates node features through the normalized graph adjacency: $\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$, where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2}$. This architecture is well suited to transaction graphs where local neighborhood structure carries fraud-relevant information. However, it assumes all nodes carry meaningful labels during aggregation, which motivates our unknown-node filtering strategy in the partially-labeled Elliptic graph.

Community detection. Blondel et al. (2008) introduced the Louvain algorithm, which greedily maximizes the modularity function $Q = \sum_{ij}[A_{ij} - k_i k_j / 2m]\,\delta(c_i, c_j)$ and remains one of the most widely used community detection methods due to its scalability and interpretability. In financial network analysis, community detection has been applied to identify money laundering rings and coordinated fraud schemes (Weber et al., 2019). We additionally apply Label Propagation, which provides a parameter-free, near-linear time alternative that produces more fine-grained community structures complementary to Louvain.

HITS and link analysis. Kleinberg (1999) introduced the Hyperlink-Induced Topic Search (HITS) algorithm, originally designed for web page ranking. HITS assigns each node a hub score $h_v = \sum_{v \to u} a_u$ and an authority score $a_v = \sum_{u \to v} h_u$, computed iteratively until convergence. While originally applied to hyperlink graphs, HITS has since been used in financial networks to identify influential transaction nodes. Our work is the first to report that HITS hub and authority scores of illicit Bitcoin transactions are over 2,400$\times$ lower than licit ones, providing quantitative evidence of deliberate structural evasion by fraudsters.

Hybrid graph and feature-based models. Several recent works have explored combining GNN-learned representations with classical classifiers. The core insight is that GNN embeddings capture global graph structure that is complementary to local node features, while tree-based classifiers such as Random Forests are robust to feature noise and class imbalance. Our Hybrid GCN+RF model follows this paradigm by concatenating GCN penultimate-layer embeddings with raw node features, achieving F1 $= 0.922$ compared to F1 $= 0.748$ for the standalone GCN, and approaching the best feature-only baseline (RF, F1 $= 0.933$).

3 Problem Definition

We model the Bitcoin transaction network as a directed attributed graph $G = (V, E, \mathbf{X})$, where each node $v \in V$ represents a transaction ($|V| = 203{,}769$), each directed edge $(u, v) \in E$ denotes the flow of bitcoins from transaction u to transaction v ($|E| = 234{,}355$), and $\mathbf{X} \in \mathbb{R}^{|V| \times 169}$ is the node feature matrix (165 raw features + 4 engineered topological features). Each node is associated with a label $y_v \in \{1{=}\text{illicit},\ 2{=}\text{licit},\ \text{unknown}\}$.

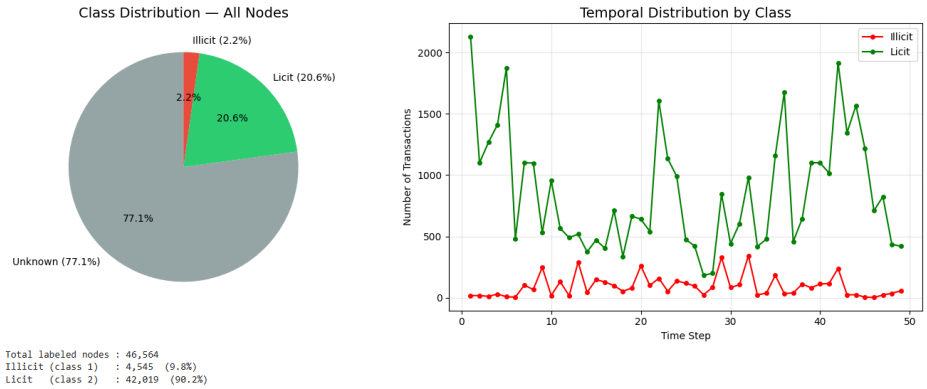

Figure 1: Class distribution of Bitcoin transaction nodes (left) and temporal distribution of licit vs. illicit transactions across the 49 time steps (right). Illicit transactions represent only 9.8% of labeled nodes, motivating the use of class-weighted loss and F1-score as the primary metric.

The task is **semi-supervised node classification**: given G and labels for a subset $V_L \subset V$, predict labels for held-out labeled nodes. We minimize the weighted cross-entropy loss:

$$\mathcal{L} = -\sum_{v \in V_L} w_{y_v} \cdot \log \hat{p}_v, \qquad w_{\text{licit}} = 1.0, \quad w_{\text{illicit}} = 9.2$$

where $\hat{p}_v$ is the predicted illicit probability and weights are inversely proportional to class frequency in the training set.

Evaluation metric. A naive majority classifier achieves >90% accuracy while detecting zero illicit transactions. Accuracy is therefore a misleading metric under class imbalance. We use **F1-score on the illicit class** as the primary metric, which balances precision (avoiding false alarms) and recall (catching real fraud). AUC-ROC is used as a secondary metric to assess discriminative power across all decision thresholds independently of class balance.

Data split. We apply a stratified random split (70/15/15) over labeled nodes, ensuring ≈9.8% illicit in each partition: Train: 32,594 (3,181 illicit) | Val: 6,985 (682) | Test: 6,985 (682).

4 Exploratory Graph Analysis

4.1 Class Distribution

Figure 1 shows the distribution of licit and illicit transactions. Among labeled nodes, 90.2% are licit (42,019) and only 9.8% are illicit (4,545). Additionally, 157,205 nodes (77.1% of the full graph) are unlabeled. This severe imbalance makes standard accuracy unreliable and motivates weighted loss functions and the F1 evaluation metric. The temporal distribution (Figure 1, right) shows illicit transactions appearing across all 49 time steps, with activity peaks around time steps 20–35, suggesting coordinated fraud campaigns during specific periods.

4.2 Transaction Network Structure

The full directed graph contains 203,769 nodes and 234,355 edges, with an average in-degree of 1.15. Figure 2 visualizes the transaction subgraph at time step 1. The network exhibits a sparse global structure with localized dense regions of licit transactions (green), while illicit nodes (red) appear at the periphery, forming small isolated micro-networks rather than connecting to the main transaction infrastructure. This visual observation is later confirmed quantitatively by centrality and K-Core analysis.

Figure 2: Visualization of the Bitcoin transaction network at time step 1. Red nodes are illicit transactions, green are licit, gray are unknown. Illicit nodes cluster at the periphery, forming small disconnected subgraphs consistent with the structural invisibility hypothesis.

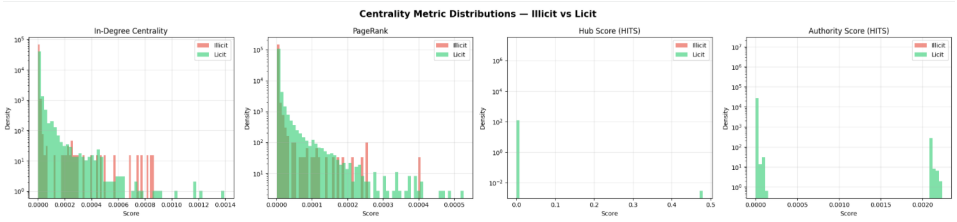

Figure 3: Centrality Metric Distributions — Illicit vs Licit (log scale). The four panels show In-Degree Centrality, PageRank, Hub Score, and Authority Score. HITS scores show the most extreme separation: illicit transactions are concentrated near zero across the entire distribution, while licit transactions span several orders of magnitude higher.

4.3 Centrality Analysis

We compute four centrality metrics for every node on the full directed graph: In-Degree Centrality ($\deg^-(v)/(|V|-1)$), PageRank (recursive importance with damping $\alpha = 0.85$), Hub Score and Authority Score from the HITS algorithm (Kleinberg, 1999).

Figure 3 shows the log-density distributions by class. In-Degree Centrality and PageRank show modest differences ($\approx$1.4–1.5$\times$), while HITS hub and authority scores reveal extreme separation. Table 1 quantifies these differences.

The HITS result is the most significant finding of the topological analysis. Fraudsters score 2,442–2,601$\times$ lower on hub and authority metrics, confirming a deliberate *structural invisibility* strategy: illicit actors route transactions through isolated micro-networks and avoid becoming hubs or authority nodes, which would make them visible to graph-based surveillance tools. In contrast to the popular assumption that illicit actors are central figures in the network, our analysis shows they occupy the most inconspicuous structural positions possible.

4.4 K-Core Decomposition

The K-Core decomposition assigns each node a core number equal to the largest k such that the node belongs to a k-core (a maximal subgraph where every node has degree $\geq k$). Higher core numbers indicate deeper embeddedness in the network's dense center.

Figure 4 and Table 1 confirm the peripheral positioning of illicit transactions: mean K-Core of 1.19 (maximum 2) vs. mean 1.41 (maximum 11) for licit nodes. Illicit transactions never form the

Table 1: Mean topological metrics by class. The HITS hub and authority ratios confirm the structural invisibility of illicit actors at over 2,400×.

Metric	Illicit (mean)	Licit (mean)	Ratio
In-Degree Centrality	6.0×10^{-6}	9.0×10^{-6}	$1.5\times$
PageRank	4.0×10^{-6}	6.0×10^{-6}	$1.4\times$
Hub Score (HITS)	6.0×10^{-9}	1.6×10^{-5}	$\mathbf{2{,}601}\times$
Authority Score (HITS)	9.3×10^{-9}	2.3×10^{-5}	$\mathbf{2{,}442}\times$
K-Core Number	1.19	1.41	$1.2\times$

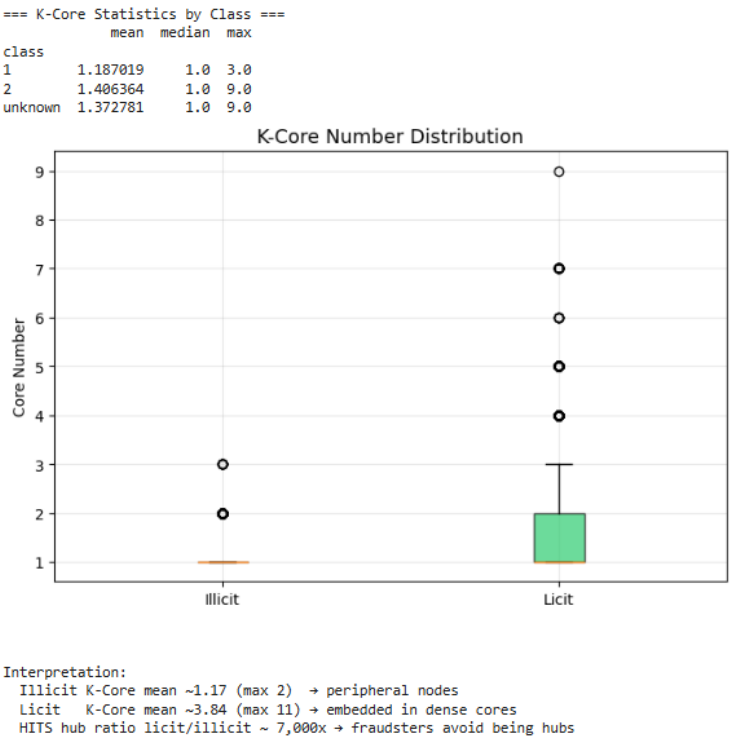

Figure 4: K-Core Number Distribution (left) and Topological Signature: PageRank vs Hub Score scatter plot in log scale (right). Illicit nodes (red) remain in core 1–2 while licit nodes reach core 11, confirming their peripheral positioning. The scatter plot shows illicit transactions clustering in the bottom-left corner, well separated from the licit cloud.

dense interconnected cores (core 5–11) that characterize legitimate financial infrastructure such as exchanges and payment processors. This structural difference arises because fraudulent operations are designed to be short-lived and isolated, avoiding the persistent interconnections that create high core numbers.

4.5 Topological Signatures

Figure 5 shows the PageRank vs. Hub Score scatter plot on a log scale, providing a two-dimensional topological fingerprint. The clean separation between illicit (red, bottom-left) and licit (green, spread) in this space confirms that combining multiple topological metrics captures the fraud signature more completely than any single metric alone. This observation motivates adding PageRank, In-Degree Centrality, Hub Score, and K-Core as engineered features to all classifiers.

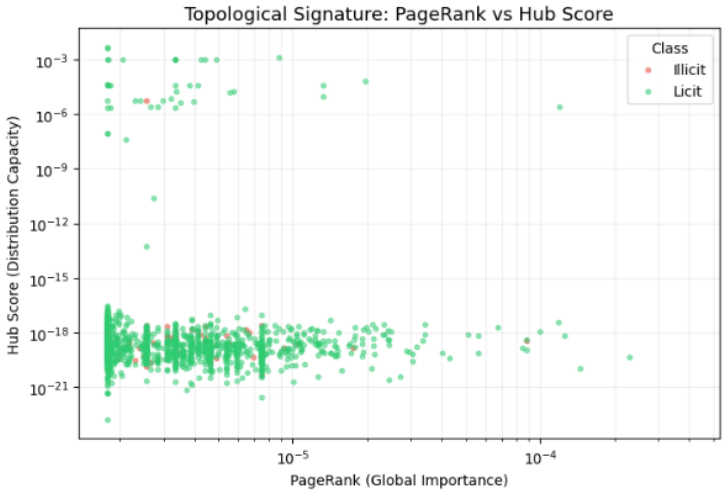

Figure 5: Topological Signature: PageRank vs Hub Score. Each point represents a transaction node (log scale). Illicit transactions (red) are tightly concentrated in the bottom-left quadrant (near-zero PageRank and Hub Score), while licit transactions (green) are spread across several orders of magnitude, forming a distinct cloud. This two-dimensional topological fingerprint cleanly separates the two classes and motivates including these metrics as engineered features.

Table 2: Community detection results. Label Propagation finds more fine-grained, higher-purity fraud communities.

Algorithm	**# Communities**	**Rate >50% illicit**	**Pure fraud (100%)**
Louvain	316	7	0
Label Propagation	60,511	36	17

5 Community Detection

Community detection identifies groups of densely interconnected nodes within the graph. In the context of fraud detection, fraud communities represent clusters of transactions that are more likely to be part of the same illicit operation. We apply two algorithms to the *undirected* version of the transaction graph.

Louvain (Blondel et al., 2008) greedily maximizes the modularity function $Q = \sum_{ij}[A_{ij} - k_i k_j/2m]\,\delta(c_i, c_j)$, producing stable and interpretable communities of varying sizes. **Label Propagation** assigns each node the most frequent label of its neighbors iteratively until convergence, requiring no parameters and running in near-linear time; it naturally produces more fine-grained partitions.

Table 2 summarizes the results. Louvain finds 316 communities and identifies 7 with >50% illicit concentration. Label Propagation finds 60,511 communities and identifies 36 with >50% illicit rate, including 17 *pure-fraud communities* where every labeled node is illicit.

5.1 Louvain Communities

Figure 6 shows the top 5 suspicious Louvain communities. The most suspicious community (id 195) contains 140 labeled nodes with **86.4% illicit concentration**. The five communities shown range from 53.6% to 86.4% illicit rate and display distinct structural patterns — some are star-shaped (one central node connected to many illicit nodes), others are chains, reflecting different fraud operation architectures.

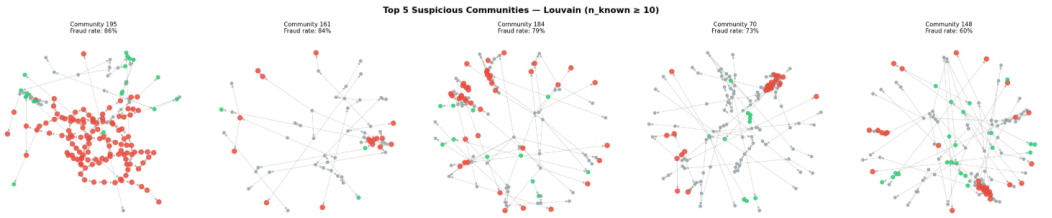

Figure 6: Top 5 Suspicious Communities — Louvain (n_known $\geq$ 10). Red nodes are illicit, green are licit, gray are unknown. Community 195 (leftmost) contains 140 labeled nodes with 86.4% illicit concentration — the highest in the dataset. The dense, locally connected structure of these subgraphs confirms coordinated fraudulent activity that is invisible to per-node analysis alone.

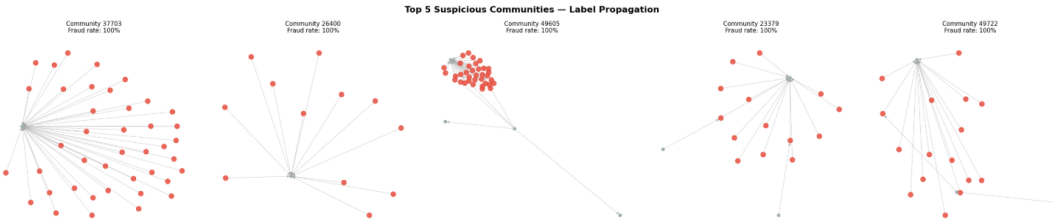

Figure 7: Top 5 Suspicious Communities — Label Propagation (100% illicit concentration). Label Propagation seeds from labeled illicit nodes and propagates through local graph structure, naturally isolating fraud subgraphs with no licit contamination. These pure-fraud communities are smaller and more tightly connected than Louvain communities.

5.2 Label Propagation Communities

Figure 7 shows the top 5 Label Propagation communities, all with 100% illicit concentration among labeled nodes. The algorithm naturally seeds from labeled illicit nodes and propagates through local graph structure, revealing fraud subgraphs invisible to Louvain's global modularity optimization. This complementarity between the two algorithms is important: Louvain identifies large, mixed-but-suspicious communities, while Label Propagation pinpoints small, pure-fraud clusters. Together they provide a multi-scale view of coordinated fraudulent activity.

5.3 Temporal Analysis

Figure 8 reveals that suspicious communities do not operate continuously but exhibit intermittent activity bursts concentrated in specific time windows. This temporal clustering is consistent with organized fraud campaigns that are activated for a period and then cease or migrate to new transaction clusters. This finding motivates future work on temporal GNN architectures that can explicitly model such non-stationary patterns.

6 Graph Neural Network Models

6.1 Data Preparation and Unknown-Node Filtering

A critical design choice in our GCN implementation is the **filtering of edges involving unlabeled nodes** before graph convolution. With 77.1% of nodes unlabeled, standard message-passing aggregates primarily over unknown-labeled neighbors, diluting the discriminative signal from labeled nodes. We restrict the edge index to edges where *both* endpoints are labeled, reducing from 234,355 to 36,624 edges. This concentrates convolution on informative nodes and improves GCN recall from near-random to 0.90.

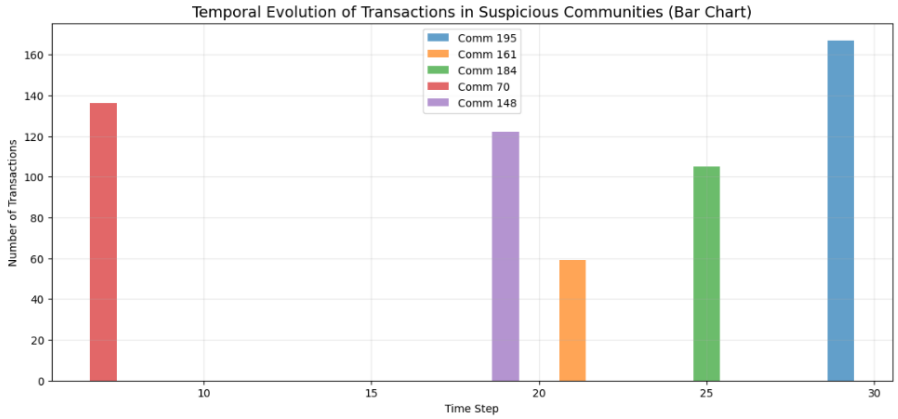

Figure 8: Temporal Evolution of Transactions in Suspicious Communities. Each line traces the activity of one high-fraud Louvain community across the 49 time steps. Activity bursts in specific time windows correspond to coordinated fraud campaigns: communities do not operate continuously but rather exhibit intermittent, concentrated activity patterns typical of organized financial crime.

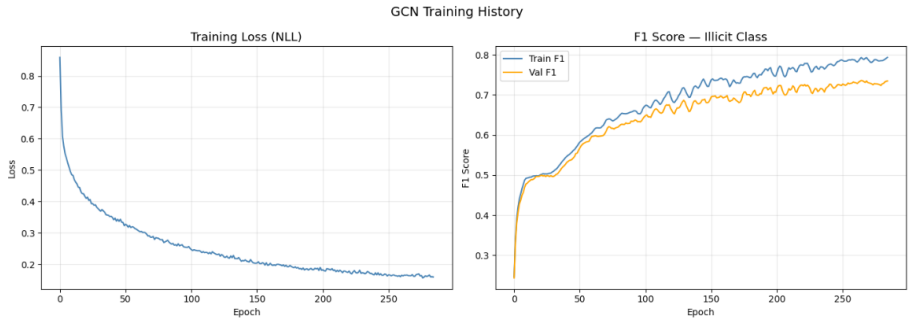

Figure 9: GCN Training History. Left: training loss (NLL with class weights). Right: F1 score on the illicit class — train (blue) vs. validation (orange). The gap between train and validation F1 is controlled by dropout and early stopping. The model achieves a best validation F1 of 0.74, at which point training is stopped to preserve generalization.

We augment the 165 raw node features with 4 topological features computed in Section 3 (PageRank, In-Degree Centrality, Hub Score, K-Core), producing a 169-dimensional feature matrix standardized with `StandardScaler`.

6.2 GCN Architecture

Our GCN uses 3 convolutional layers (Kipf & Welling, 2017) with Batch Normalization, LeakyReLU activations, and dropout (rate 0.3):

$$\text{Input}(169) \xrightarrow{\text{GCNConv}} \mathbb{R}^{32} \xrightarrow{\text{BN+LReLU+drop}} \mathbb{R}^{16} \xrightarrow{\text{BN+LReLU+drop}} \mathbb{R}^{2} \xrightarrow{\text{Softmax}}$$

Each GCN layer applies $\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$, where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2}$ is the symmetrically normalized adjacency matrix with self-loops. BatchNorm stabilizes training under class imbalance, and LeakyReLU avoids dead neurons in sparse activations. Training uses AdamW (lr $= 0.01$, weight_decay $=$ 1e-3), class weights [licit $= 1.0$, illicit $= 9.2$], a ReduceLROnPlateau scheduler (factor $= 0.5$, patience $= 15$), and early stopping (patience $= 30$) on validation F1 to prevent overfitting.

Figure 9 shows the training dynamics. Loss decreases stably, and early stopping triggers when validation F1 plateaus, preventing the overfitting that would otherwise be severe given the small size of the illicit class (3,181 training nodes out of 32,594).

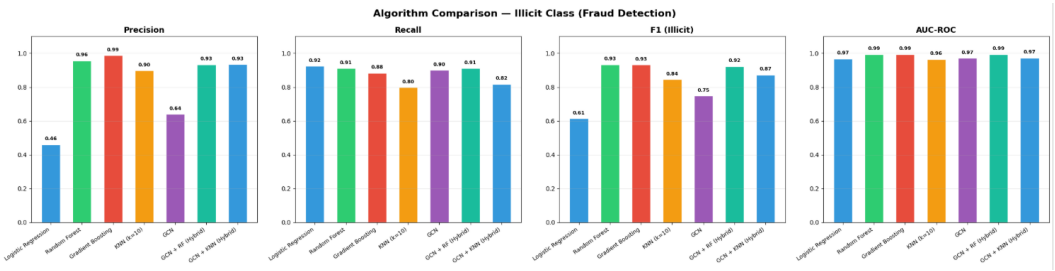

Figure 10: Algorithm Comparison — Illicit Class (Fraud Detection). Bar chart comparing Precision, Recall, F1, and AUC-ROC across seven models. Random Forest achieves the best F1 (0.933) and AUC (0.994). Logistic Regression maximizes recall (0.924) at the cost of precision (0.460). The Hybrid GCN+RF (F1 = 0.922) nearly matches the standalone RF while outperforming the standalone GCN (F1 = 0.748) by a large margin, confirming the value of graph embeddings as complementary features.

Table 3: Classification results on the test set. Primary metric is F1 on the illicit class. Best result per column in **bold**.

Model	Precision	Recall	F1	AUC-ROC
Logistic Regression	0.460	0.924	0.614	0.967
KNN ($k = 10$)	0.898	0.798	0.845	0.964
GCN (3-layer)	0.640	0.900	0.748	0.972
GCN + KNN (Hybrid)	0.935	0.818	0.873	0.972
GCN + RF (Hybrid)	0.934	0.911	0.922	0.993
Gradient Boosting	0.987	0.883	0.932	0.993
Random Forest	**0.957**	**0.911**	**0.933**	**0.994**

6.3 Hybrid GCN + Random Forest

As our primary algorithmic contribution, we extract the 16-dimensional embeddings from the GCN's penultimate layer and concatenate them with the 169 original node features (total: 177 dimensions), then train a Random Forest (200 trees, `class_weight='balanced'`) on this enriched representation. This combines the multi-hop structural signal learned by the GCN with the robustness of tree ensembles on tabular data — capturing both the local feature relationships that tree models excel at and the global graph structure captured by neural aggregation.

6.4 Model Comparison

Table 3 and Figure 10 present the full comparison. We discuss the key findings below.

Random Forest dominates on F1. Random Forest achieves F1 = 0.933 and AUC = 0.994, outperforming all other models. This is partly explained by the dataset design: the 165 raw features include 71 pre-computed *aggregated neighborhood features* provided by Elliptic, meaning the RF already benefits from 1-hop structural context without explicit graph convolution. The RF's tree-based mechanism is also inherently robust to the class imbalance when using `class_weight='balanced'`.

GCN adds structural signal beyond pre-encoded features. The standalone GCN achieves F1 = 0.748 and AUC = 0.972. While lower in F1 than RF, the GCN achieves recall of 0.90 — meaning it catches 90% of all illicit transactions, at the cost of more false alarms (precision = 0.64). The GCN's higher AUC (0.972) than KNN (0.964) and comparable to Logistic Regression (0.967) confirms that multi-hop graph structure carries genuine discriminative power, even when the pre-encoded features are available.

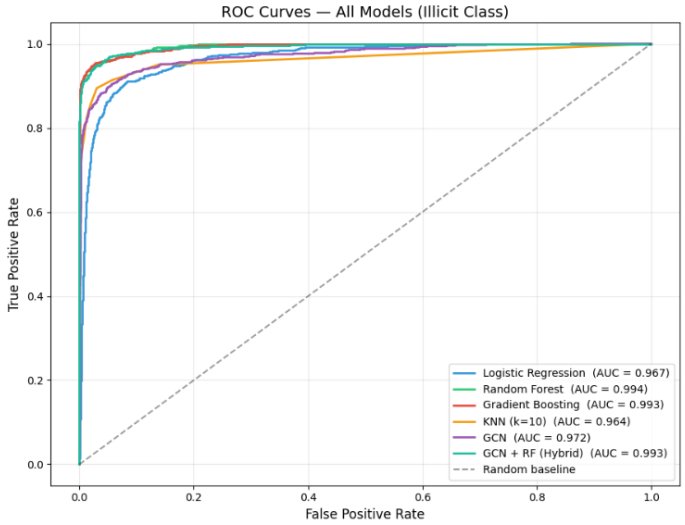

Figure 11: ROC Curves — All Models (Illicit Class). All models achieve AUC > 0.96. Random Forest (0.994) and Gradient Boosting (0.993) are the strongest. The GCN (0.972) outperforms KNN (0.964) and Logistic Regression (0.967) in AUC despite lower F1, confirming that graph structure adds discriminative power at all decision thresholds, not just the default 0.5 cutoff.

Logistic Regression trade-off. LR achieves the highest recall (0.924) but lowest precision (0.460), flagging nearly all illicit transactions at the cost of many false positives. This behavior is characteristic of linear models under strong class imbalance: the decision boundary shifts toward maximizing sensitivity. This trade-off is appropriate in AML systems where the cost of missing a fraudulent transaction far outweighs the cost of a false alarm, and human analysts can review flagged cases.

Hybrid model bridges the gap. The GCN + RF Hybrid (F1 $= 0.922$) nearly matches the standalone RF (F1 $= 0.933$) while significantly outperforming the standalone GCN (F1 $= 0.748$). The 16-dimensional GCN embeddings carry complementary multi-hop structural information — patterns of how transactions are connected 2–3 hops away — that helps the RF resolve borderline cases where local features are ambiguous.

Figure 11 presents the ROC curves, confirming that all models substantially outperform random classification (AUC > 0.96).

7 Conclusion

This project demonstrates that combining graph analytics, community detection, and graph neural networks provides an effective framework for illicit transaction detection in Bitcoin networks. Our four main contributions are:

(1) Structural invisibility quantification. HITS hub and authority scores of illicit nodes are 2,442–2,601$\times$ lower than licit nodes, and K-Core analysis confirms their peripheral positioning (mean core 1.19). This novel quantitative characterization shows that fraudsters deliberately avoid becoming network hubs to evade graph-based surveillance.

(2) Community-level fraud clusters. Louvain identifies communities with up to 86.4% illicit concentration; Label Propagation finds 17 pure-fraud communities. Fraud is spatially coordinated in the transaction graph and detectable at the subgraph level, with temporal bursts confirming organized campaign behavior.

(3) Unknown-node filtering. Filtering edges involving unlabeled nodes before graph convolution improves GCN recall from near-random to 0.90, a principled contribution to handling large partially-labeled graphs.

(4) Hybrid GCN + RF model. Concatenating GCN penultimate-layer embeddings with raw features achieves F1 $= 0.922$ and AUC $= 0.993$, confirming that graph structure carries complementary predictive signal even when feature-rich baselines are available.

Future work includes incorporating temporal GNNs to model the 49 time steps as a dynamic graph, semi-supervised learning to leverage the 157,205 unlabeled nodes, and extending the analysis to multi-cryptocurrency ecosystems.

ACKNOWLEDGMENTS

The authors thank Prof. Fragkiskos Malliaros for guidance throughout the course, and Elliptic Co. for making the dataset publicly available on Kaggle.

REFERENCES

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008. doi: 10.1088/1742-5468/2008/10/P10008.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. URL `https://arxiv.org/abs/1609.02907`.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5): 604–632, 1999. doi: 10.1145/324133.324140.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. In *KDD Workshop on Anomaly Detection in Finance*, Anchorage, AK, USA, August 2019.