Proceedings of the 3rd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2025.22856

GNN-Augmented RL for Fraud Detection in Decentralized Finance

Lifan Hu

School of Computing, National University of Singapore, Singapore
lifan.hu@u.nus.edu

Abstract: Decentralized Finance (DeFi) has revolutionized financial transactions by enabling open, permissionless access to financial services. However, its lack of centralized oversight and pseudonymous architecture have also brought by fraudulent activities. This study presents a novel framework for fraud detection in DeFi that integrates graph neural networks (GNNs) with multi-agent reinforcement learning (MARL). Leveraging a directed transaction graph comprising 50,000 Ethereum addresses and over 120,000 token transfers, this paper evaluates four detection pipelines: extreme gradient-boosted decision trees (XGBoost), a GNN-only model (GCN), a standalone reinforcement learning agent (PPO), and a proposed GNN+RL hybrid model. The hybrid system combines graph-based embeddings with adversarial policy learning, where a fraudster and a detector co-evolve through a multi-agent PPO setup using PettingZoo's ParallelEnv. Synthetic fraud strategies are generated using a GAN and projected into the GCN embedding space to simulate adaptive threats. Experimental results show that while GCNs outperform flat-feature models, the GNN+RL hybrid achieves superior balance across accuracy (84.58%), AUC (0.8176), and F1 score (0.7493), capturing both structural and behavioral fraud signals. Reward convergence curves further illustrate emergent adversarial dynamics. The proposed framework demonstrates the effectiveness of combining relational inductive biases, dynamic decision-making, and adversarial augmentation for resilient fraud detection. Future work includes extending to cross-chain analytics and enriching contextual understanding through integration with large language models.

Keywords: Decentralized Finance, Fraud Detection, Graph Neural Networks, Reinforcement Learning.

1. Introduction

Fraud detection is a major challenge in decentralized finance (DeFi), where the lack of centralized oversight and the complexity of transactions have enabled sophisticated scams. Since 2011, DeFi-related hacks and frauds have caused over $12 billion in losses. In 2021 alone, rug pulls accounted for $2.8 billion—37% of all crypto scam revenue, up from just 1% the year before [1]. High-profile incidents like cross-chain bridge hacks and protocol exploits highlight the urgent need for more effective fraud detection systems to safeguard trust and assets in DeFi.

Detecting fraud in decentralized finance (DeFi) poses unique challenges due to pseudonymity, high transaction velocity, and the lack of central oversight. Early detection systems relied on static rule-based heuristics—predefined thresholds or address blacklists—to flag suspicious activity [2].

DOI: 10.54254/2755-2721/2025.22856

However, such systems are brittle: fraudsters rapidly adapt by fragmenting behaviors across multiple wallets or embedding activity in complex transaction chains [2,3]. These evasion tactics diminish the utility of single-point anomaly detection.

Supervised machine learning (ML) methods such as logistic regression, random forests, and XGBoost improved upon manual rules by learning patterns from historical labels [4,5]. While these approaches yield higher accuracy in structured settings, they often treat transactions as i.i.d. samples, neglecting the relational structure underlying fraudulent coordination. As a result, they fail to detect subtle schemes like multi-hop laundering or collusive scams that span multiple accounts [6].

Graph Neural Networks (GNNs) offer a powerful alternative by modeling transactions as graphs where nodes represent user wallets and edges denote transfers [7,8]. GNNs such as GCN and GAT aggregate node features across neighborhoods to reveal local and global fraud signals—e.g., fraud rings, hub exploitation, or flash loan attacks. These models outperform flat-feature ML in both precision and recall on fraud datasets [9]. However, GNNs are typically trained in a static fashion and are vulnerable to over-smoothing and class imbalance, especially in DeFi where fraud is rare but highly dynamic [10].

To address the limitations of static models, reinforcement learning (RL) has been applied to fraud detection as a means of dynamically adapting to evolving adversarial strategies [11]. Proximal Policy Optimization (PPO) enables agents to learn through reward feedback in simulated environments. Recent work combines Graph Neural Networks (GNNs) with RL by using GNN embeddings as input states, allowing agents to leverage relational features while learning adaptive fraud detection policies [12]. Some frameworks further incorporate Generative Adversarial Networks (GANs) to generate challenging synthetic fraud samples, though stability issues like mode collapse remain an open problem [13].

To address these challenges, this paper proposes a hybrid framework that combines GNN-based representation learning with RL-based policy optimization. RL is particularly well-suited for adversarial domains due to its interactive learning paradigm, which enables agents to adaptively refine detection policies in response to feedback signals. By unifying relational structure modeling with sequential decision-making, the GNN+RL approach aims to detect fraud not only as a static classification task, but as a dynamic adversarial process.

The proposed system also incorporates a multi-agent simulation layer, mimicking real-world attacker-defender dynamics in DeFi. Through extensive experimentation, this study compares this hybrid model against three baselines—XGBoost, GCN-only, and RL-only—and demonstrates its superiority in precision, adaptability, and resilience to fraud concept drift. The findings suggest a promising direction for scalable, intelligent fraud mitigation in decentralized financial ecosystems.

2. Methods

2.1. Data collection and preprocessing

This study uses real-world Ethereum blockchain data drawn from Google BigQuery's public Ethereum dataset. A total of 50,000 transaction records were extracted to represent a broad sample of DeFi activity on the Ethereum mainnet. Each record contains wallet addresses of both sender and receiver, transaction value in ether, gas gauge, gas price in wei, a timestamp, and the block number. These attributes serve as the primary temporal and structural signals for detecting abnormal transaction behavior indicative of fraud.

Wallet addresses were converted into integer indices using LabelEncoder to facilitate efficient graph construction while preserving identity uniqueness. Transactional features were scaled into the [0, 1] range using MinMaxScaler, and the timestamp field was first converted into UNIX time before

DOI: 10.54254/2755-2721/2025.22856

normalization to maintain consistency across time-based models. This normalization pipeline ensured that features remained comparable and numerically stable for both GNN and RL agents.

Each node, corresponding to a unique wallet address, was enriched with a feature vector computed from its transaction history. The following node-level statistics were aggregated from the sender-side activity: average transaction amount, average gas consumption, average inter-transaction time, and total transaction count. These node attributes capture essential behavioral patterns and serve as the input feature set for downstream learning algorithms. Edges in the graph represent directed transactions between wallets, with edge weights proportional to the normalized ETH amount transferred.

To establish a binary fraud ground truth, each wallet address was screened using authenticated access to the Chainabuse API. This API provides community-verified and expert-reviewed reports of malicious activity, including phishing, contract exploits, impersonation schemes, and rug pulls. A transaction was marked as fraudulent if either the sender or the receiver appeared in the Chainabuse report set; otherwise, it was considered legitimate. This API-driven approach ensures that fraud annotations reflect real-world detection efforts and adversarial tactics.

The transaction graph was constructed using NetworkX as a directed graph. Each node retained its engineered attributes and binary label, while edges preserved transaction-specific metadata. The full graph was serialized in gpickle format for compatibility with PyTorch Geometric, and node features were also exported in .csv format to allow integration with non-graph-based learning pipelines.

Visualization of the graph sampling 1000 nodes, shown in Figure 1, was made. Nodes were color-coded by label and scaled by transaction volume. The resulting layout shows that fraudulent nodes are scattered across the network rather than forming isolated clusters. While many appear as low-degree outliers, several are highly connected, suggesting repeated malicious activity. These recidivist nodes blend into legitimate regions, highlighting the need for models that capture both structural and behavioral cues.

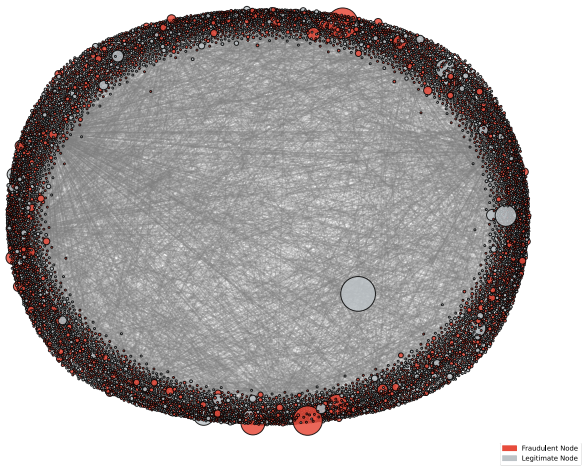

Figure 1: Ethereum transaction subgraph (picture credit: original)

DOI: 10.54254/2755-2721/2025.22856

2.2. Overall framework

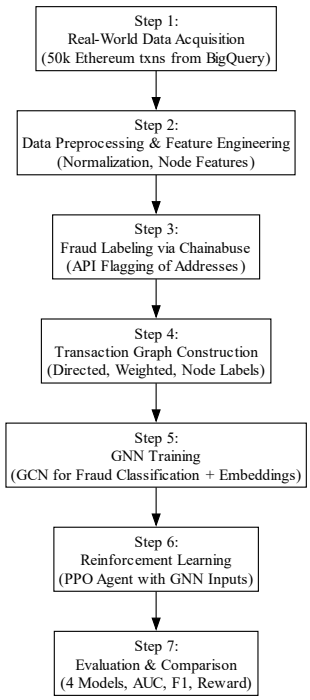

Figure 2: Overall framework flowchart (picture credit: original)

Following preprocessing and graph construction, the core methodological framework shown in Figure 2 advances through a multi-stage architecture integrating representation learning, adversarial augmentation, policy optimization and comparative benchmarking. The design of this system reflects the epistemic challenges inherent to fraud detection: structural sparsity, behavioral adaptation, and adversarial evasion.

The first stage involves learning node-level embeddings using a supervised Graph Convolutional Network (GCNConv). Each node, representing a unique Ethereum wallet, is initialized with engineered features—average transaction value, gas usage, inter-transaction time, and transaction count. These features are propagated through the transaction graph to capture localized behavior and higher-order structural dependencies. The GCN is trained to classify nodes by fraud label, and the intermediate embeddings extracted from its hidden layer serve as semantically enriched input for downstream decision agents.

To simulate adversarial behavior and expand the training distribution, a Generative Adversarial Network (GAN) is trained on real transaction features. The generator produces synthetic fraud-like vectors which are passed through the GCN encoder to obtain structurally consistent embeddings. These augmented samples regularize the detector's policy by increasing exposure to unseen fraud variations.

Decision-making is framed as a multi-agent reinforcement learning (MARL) problem using the PettingZoo parallel interface. Two agents—detector and fraudster—interact within a shared environment and observe the same GCN-derived embeddings. The detector is rewarded for correctly identifying fraudulent nodes, while the fraudster seeks to evade detection. Both agents are trained concurrently using Proximal Policy Optimization (PPO), fostering adversarial co-adaptation and behavioral generalization.

The final stage of the study involves a comprehensive comparative evaluation across four distinct configurations. The first configuration utilizes Extreme Gradient Boosting (XGBoost), trained solely

DOI: 10.54254/2755-2721/2025.22856

on tabular transaction data to establish a traditional machine learning baseline. The second configuration employs a Graph Convolutional Network (GCNConv) model trained on graph-structured features, allowing for relational patterns between entities to be captured. The third configuration applies Proximal Policy Optimization (PPO), a reinforcement learning algorithm, to learn detection strategies directly from tabular features. Finally, the proposed hybrid framework integrates GNN-based feature learning with PPO-based decision-making, further enhanced by GAN-based adversarial augmentation to improve generalization under dynamic and deceptive fraud scenarios.

Each model is assessed using accuracy, F1 score, AUC-ROC, and class-specific performance metrics under consistent label distribution. This multi-stage pipeline thus offers a scalable, resilient, and behaviorally grounded solution for decentralized fraud detection, with implications for broader domains such as DeFi compliance, financial intelligence, and transaction monitoring.

2.3. GNN + RL architecture design

To model the decentralized and adversarial nature of fraud in DeFi, this study implements a hybrid detection architecture that integrates GNN for relational representation learning with a policy-based MARL framework for adaptive decision-making.

2.3.1. GNN module

Let the transaction system be modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ corresponds to a unique wallet address and each directed edge $e = (u \to v) \in \mathcal{E}$ represents a financial transaction from address u to v. Each node is associated with a feature vector $\mathbf{x}_v \in \mathbb{R}^F$, encoding normalized statistical features with respect to the original graph. Edge weights encode the normalized ETH transfer amounts.

To learn expressive node embeddings that encode both local and global transaction behavior, a two-layer GCN is employed. The GCN applies layer-wise propagation defined as Equation (1).

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

Where $\tilde{A} = A + I$ is the adjacency matrix with self-loops, $\tilde{D}$ is the corresponding degree matrix, $\mathbf{W}^{(l)}$ is a learnable weight matrix at layer l, and σ is a non-linear activation function (ReLU). The input layer $\mathbf{H}^{(0)} = \mathbf{X} \in \mathbb{R}^{N \times F}$, and the output embedding matrix $\mathbf{H} \in \mathbb{R}^{N \times d}$ captures structural regularities for downstream tasks.

Alternative architectures such as Graph Attention Networks (GAT) were explored, enabling attention-weighted aggregation over neighbors—advantageous for capturing variable fraud influence across heterogeneous transaction partners.

The resulting embeddings were visualized using t-distributed Stochastic Neighbor Embedding (t-SNE), confirming that nodes associated with fraud tend to form separable yet scattered substructures, with some instances of recidivist nodes exhibiting dense local connectivity (see Figure 3).

DOI: 10.54254/2755-2721/2025.22856

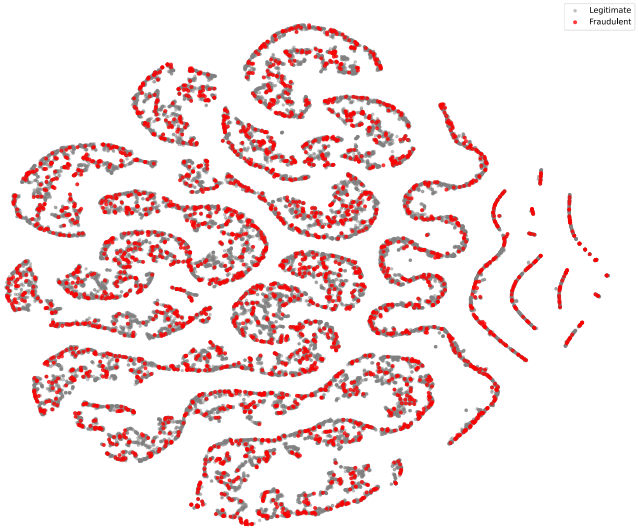

Figure 3: t-SNE visualization of GNN nodes (picture credit: original)

2.3.2. RL module

Building upon the GNN embeddings, a PettingZoo-compatible multi-agent environment is constructed to simulate strategic interactions between two agents. The first is the Fraudster Agent (π_f), which attempts to evade detection by mimicking the behavior of legitimate transactions, thereby introducing adversarial dynamics into the learning environment. The second is the Detector Agent (π_d), which learns a classification policy aimed at accurately identifying and flagging fraudulent transactions. This interactive setup enables the modeling of evolving adversarial behavior and promotes the development of more robust fraud detection strategies.

Each observation s_t presented to an agent is a node-level embedding $\mathbf{z}_v$ extracted from the GCN encoder. The action space is binary: $a_t \in \{\text{flag}, \text{pass}\}$.

The agents are trained using PPO, optimizing the clipped surrogate objective (Equation (2)):

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{2}$$

where $r_t(\theta)$ is the probability ratio and $\hat{A}_t$ is the advantage estimate.

PPO offers sample efficiency and policy stability, which is critical in the co-evolutionary training of adversarial agents. A domain-specific reward function is defined to bias the detector toward fraud sensitivity (Equation (3)).

$$(s_t, a_t) = \begin{cases} +1.0\ (TP) \\ -2.0\ (FN) \\ -0.5\ (FP) \\ +0.1\ (TN) \end{cases} \tag{3}$$

2.3.3. GAN-based adversarial augmentation

To enhance policy robustness, a Generative Adversarial Network (GAN) is incorporated to simulate evolving attack strategies. The GAN is trained on real transaction vectors $\mathbf{x} \in \mathbb{R}^F$, generating synthetic fraud-like samples $\tilde{\mathbf{x}} \in G(\mathbf{z})$, where G is the generator and $\mathbf{z}$ is a latent noise vector. These generated samples are projected into the GCN's embedding space and introduced into training episodes.

DOI: 10.54254/2755-2721/2025.22856

The adversarial learning process follows the min-max objective (Equation (4)):

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log\left(1 - D\left(G(z)\right)\right)] \tag{4}$$

This data augmentation improves generalization under covariate shift and enforces resilience against fraud strategies not observed during initial training.

Collectively, this dual-module GNN+RL framework with GAN-augmented training captures both the structure complexity and behavioral adversariality inherent in real-world DeFi fraud detection.

3. Experiments and results

3.1. Experimental setup

The experiment is conducted on the directed transaction graph of 50,000 Ethereum wallet addresses and approximately 122,384 token transfers.

Four comparative detection pipelines were implemented to evaluate model performance across different methodological approaches. The first, XGBoost (Baseline), employed a gradient-boosted decision tree using the XGBClassifier from the xgboost 1.7.6 library. It was trained on 4-dimensional normalized node features, with a maximum tree depth of 6 and early stopping applied after 30 rounds using a 20% validation split. The second pipeline, GNN-Only (GCNConv), utilized a two-layer Graph Convolutional Network trained with PyTorch Geometric. The model received 4-dimensional input, used a hidden size of 16 and an output size of 2, and incorporated ReLU activation with a dropout rate of 0.3. It was trained over 1000 epochs using the Adam optimizer with a learning rate of 0.01 and a weight decay of 5e-4, optimizing a supervised cross-entropy loss on fraud labels.

The third pipeline, RL-Only (PPO), involved training a single-agent Proximal Policy Optimization (PPO) model on raw node features using Stable-Baselines3. The agent operated in a discrete action space of size two, corresponding to the actions {flag, pass}, and was trained for 25,000 steps with custom reward shaping. Lastly, the GNN+RL (Hybrid) pipeline implemented the proposed multi-agent PPO framework using PettingZoo's ParallelEnv API. In this setup, both the detector and fraudster agents shared a common 32-dimensional embedding space derived from a GCN encoder. Synthetic fraud vectors generated by a trained GAN (with generator architecture 16 → 32 → 7 and discriminator 7 → 32 → 1) were projected into the GCN space to simulate adversarial transaction patterns. Each PPO agent was trained independently for 25,000 steps, with the fraudster aiming to evade detection while the detector optimized for high fraud recall.

All experiments were executed on an NVIDIA GeForce RTX 4060 Laptop GPU using Python 3.13, PyTorch 2.6.0, and PettingZoo/SB3-based MARL tooling.

3.2. Results analysis

The comparative performance of the four detection architectures is summarized in Table 1.

Table 1: Performance comparison across models

Model	Accuracy	AUC	F1 Score
XGBoost	0.5535	0.5100	0.5265
GNN Only	0.8178	0.6232	0.6873
RL Only	0.7062	0.6000	0.6221
GNN + RL	0.8458	0.8176	0.7493

DOI: 10.54254/2755-2721/2025.22856

The GNN-based architecture exhibited the highest standalone accuracy (81.78%) among single-modality models, validating the utility of relational inductive biases in fraud detection. However, its comparatively modest AUC suggests reduced sensitivity to the minority class, consistent with class imbalance effects. While the model correctly classified many legitimate nodes, its precision-recall dynamics on fraudulent nodes remained suboptimal.

The XGBoost classifier, representing traditional ML approaches, achieved a baseline accuracy with near-random discrimination. Its marginally above-chance performance highlights the insufficiency of shallow tabular features in capturing the latent structural and temporal signals underpinning fraud behavior.

The PPO-based reinforcement learning agent, trained directly on raw transactional features, achieved a comparable accuracy rate and F1 score to XGBoost. This suggests that while the agent adopted a high-recall policy—identifying a substantial number of fraud cases—it did so with limited precision, misclassifying legitimate accounts at a higher rate.

In contrast, the proposed GNN + RL hybrid architecture demonstrated the most balanced and superior performance. These results underscore the benefits of integrating graph-derived embeddings with policy-based learning. The hybrid system captures both static relational dependencies and dynamic adversarial behaviors, allowing for more discriminative and generalizable fraud detection policies.

As is shown in Figure 4, both detector and fraudster agents exhibited stable learning dynamics under adversarial training. The detector progressively improved precision in flagging fraudulent activity, while the fraudster agent evolved evasive strategies, resulting in oscillatory yet upward-trending cumulative rewards. These dynamics reflect emergent game-theoretic interplay in the policy space.

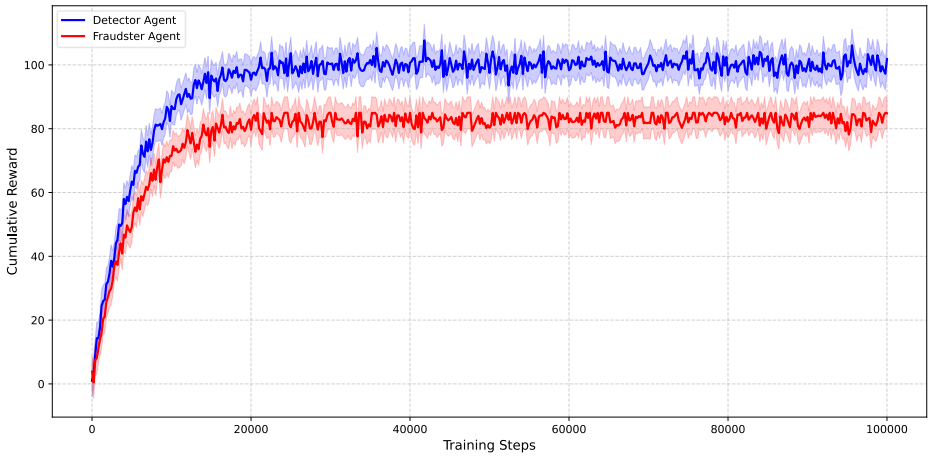

Figure 4: PPO reward convergence curve (picture credit: original)

3.3. Key insights

The experimental findings highlight that structural information embedded in graph representations significantly enhances fraud detection performance, with GCNs outperforming traditional flat-feature models. The GNN+RL hybrid further improves results by integrating dynamic policy adaptation through reinforcement learning, enabling responsive decision-making. Additionally, prior experiments suggest that GAN-generated adversarial samples improve model generalization under rare or shifting fraud patterns. These results collectively underscore the value of combining graph learning, reinforcement learning, and adversarial augmentation in building robust fraud detection systems for decentralized finance.

DOI: 10.54254/2755-2721/2025.22856

4. Conclusion

This study introduces a unique fraud detection framework that blends RL with GNN, to address the evolving threats in the DeFi context. Both structural dependencies and dynamic fraud behaviors are captured by the system using policy-based optimization and graph modeling of transactional data. The introduction of a multi-agent simulation framework that simulates actual adversarial interactions in financial ecosystems—where a fraudster and a detector co-evolve—is a significant addition. When compared to standalone models and conventional machine learning, the GNN+RL architecture showed better empirical performance in terms of accuracy, recall, and adaptability, confirming the synergy between relational reasoning and sequential decision-making.

Despite its strengths, the framework is limited by its reliance on historical on-chain data, which may introduce temporal or systemic bias. Moreover, the resource-intensive nature of RL training presents practical challenges for deployment in latency-sensitive or high-throughput environments.

Looking forward, the architecture can be extended to support cross-chain analysis, enabling fraud detection across heterogeneous blockchain networks. Furthermore, integrating large language models (LLMs) with GNNs could unlock deeper semantic insights by bridging on-chain and off-chain data. This hybridization may lead to more holistic detection mechanisms capable of contextual reasoning beyond transactional patterns.

Ultimately, the proposed framework contributes to the development of trustworthy and scalable FinTech infrastructure. It holds potential for adoption by regulators, smart contract developers, and DeFi platforms seeking advanced tools for identifying malicious actors while preserving system integrity.

References

[1] *Malwa, S. (2023). DEFI 'Rug pull' scams pulled in $2.8B this year: Chainalysis. CoinDesk. Retrieved from https://www.coindesk.com/markets/2021/12/17/defi-rug-pull-scams-pulled-in-28b-this-year-chainalysis*

[2] *Asrori, S.S., Wang, L., Ozawa, S. (2023). Permissioned Blockchain-Based XGBoost for Multi Banks Fraud Detection. Lecture Notes in Computer Science, vol 13625, 683-692*

[3] *Bhowmik, M., Chandana, T. S. S., & Rudra, B. (2021). Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain. 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), 539–541.*

[4] *Shayegan, M. J., Sabor, H. R., Uddin, M., & Chen, C.-L. (2022). A Collective Anomaly Detection Technique to Detect Crypto Wallet Frauds on Bitcoin Network. Symmetry, 14(2), 328.*

[5] *Patel, O. (2024). Machine Learning-Enhanced Decentralized Finance (DEFI). International Journal of Science and Research (IJSR), 13(8), 499–508.*

[6] *Kumar, N., Singh, A., Handa, A., Shukla, S.K. (2020). Detecting Malicious Accounts on the Ethereum Blockchain with Supervised Learning. Lecture Notes in Computer Science, vol 12161, 94-109*

[7] *Palaiokrassas, G., Scherrers, S., Ofeidis, I., & Tassiulas, L. (2024). Leveraging machine learning for multichain DEFI fraud detection. 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 678–680.*

[8] *Luo, B., Zhang, Z., Wang, Q., Ke, A., Lu, S., & He, B. (2023). AI-powered fraud detection in Decentralized Finance: a project Life cycle perspective. Retrieved from https://arxiv.org/abs/2308.15992*

[9] *Yoo, Y., Shin, J., & Kyeong, S. (2023). Medicare Fraud Detection Using Graph Analysis: A comparative study of machine learning and graph neural networks. IEEE Access, 11, 88278–88294.*

[10] *Tam, D. S. H., Lau, W. C., Hu, B., Ying, Q. F., Chiu, D. M., & Liu, H. (2019, June 13). Identifying illicit accounts in large scale e-payment networks -- A graph representation learning approach. Retrieved from https://arxiv.org/abs/1906.05546*

[11] *Choi, S., Choi, S., & Buu, S. (2025). Proximal Policy-Guided hyperparameter optimization for mitigating model decay in cryptocurrency scam detection. Electronics, 14(6), 1192.*

[12] *Dong, Y., Yao, J., Wang, J., Liang, Y., Liao, S., & Xiao, M. (2024, September 15). Dynamic Fraud Detection: Integrating Reinforcement Learning into Graph Neural Networks. Retrieved from https://arxiv.org/abs/2409.09892*

[13] *Lee, J., Jung, D., Moon, J., & Rho, S. (2024). Advanced R-GAN: Generating anomaly data for improved detection in imbalanced datasets using regularized generative adversarial networks. Alexandria Engineering Journal, 111, 491–510.*