International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 07 | Jul 2026 www.irjet.net p-ISSN: 2395-0072

FraudShield: A Hybrid GNN-DRL Framework for High-Recall Cryptocurrency Fraud Detection

Sana Sulthana[1], Banoth. Akash Nayak[2], G. Praveen Babu [3]

[1]M.Tech Scholar, Dept. of CSE, JNTUH UCESTH, Hyderabad, India
[2]M.Tech Scholar, Dept. of CSE, UCE, OU, Hyderabad, India
[3]Associate Professor, Dept. of CSE, JNTUH UCESTH, Hyderabad, India

---

Abstract - *The pseudonymous nature of cryptocurrency transactions has made blockchain networks a target for illicit activities, such as money laundering and scams, posing significant challenges for fraud detection. FraudShield proposes a novel hybrid framework that integrates Graph Neural Networks (GNNs) and Deep Reinforcement Learning (DRL) to address these challenges in dynamic transaction graphs. The GNN component, combining GraphSAGE and Graph Attention Networks (GAT) with a normalized timestep feature, models complex transaction patterns and temporal dynamics. The DRL component employs Proximal Policy Optimization (PPO) to optimize actionable mitigation strategies, balancing detection accuracy and operational efficiency. An attention mechanism enhances interpretability by identifying suspicious transaction connections. Built on the Elliptic Bitcoin Dataset, FraudShield aims to advance cryptocurrency fraud detection by prioritizing high recall, providing actionable policies, and ensuring explainability, thereby contributing to enhanced blockchain security and regulatory compliance.*

Key Words: Cryptocurrency fraud detection, Blockchain security, Graph Neural Networks (GNN), Deep Reinforcement Learning (DRL), GraphSAGE, Graph Attention Networks (GAT), Timestep feature, Proximal Policy Optimization (PPO), Transaction graphs, Money laundering, Scams, Attention mechanism, Explainability, High recall, Actionable mitigation, Elliptic Bitcoin Dataset, Regulatory compliance, Anti-Money Laundering (AML).

1. INTRODUCTION

Cryptocurrencies such as Bitcoin have revolutionized digital payments by enabling secure, decentralized, and fast transactions without relying on banks. However, their pseudonymous nature has also made them attractive for illegal activities, including money laundering, scams, ransomware, and terrorist financing. Detecting fraudulent transactions is challenging because blockchain networks are large, complex, and constantly evolving, while labeled fraudulent data is limited. Existing methods often fail to identify a significant number of illicit transactions and do not provide effective mitigation strategies. To address these challenges, the proposed **FraudShield** framework combines Graph Neural Networks (GNNs) and Deep Reinforcement Learning (DRL) to improve fraud detection and response. GNNs capture relationships among blockchain transactions, while DRL recommends actions such as monitoring, flagging, or blocking suspicious transactions. The framework also incorporates temporal features and attention mechanisms to adapt to changing fraud patterns and improve interpretability. Overall, FraudShield aims to improve higher fraud detection accuracy, enhance blockchain security, and support regulatory compliance.

2. LITERATURE REVIEW

FraudShield is a fraud detection framework for cryptocurrency transactions that combines Graph Neural Networks (GNNs) with Deep Reinforcement Learning (DRL). Traditional rule-based fraud detection struggles with blockchain's complex, anonymous transaction networks, so FraudShield uses graph-based machine learning instead, treating transactions as nodes and money flows as edges.

Using the Elliptic Bitcoin Dataset (203,769 transactions, 234,355 edges, 166 features per transaction, 49 timesteps), Two GNN variants perform the primary learning tasks.GAT weighs suspicious connections for better explainability, while GraphSAGE samples neighbors efficiently for scalability. Once GNNs identifies suspicious patterns, a DRL agent (using PPO) decides how to respond Ignore, Monitor, Flag, or Block balancing fraud detection against unnecessary disruption through a reward system (+10 for correctly blocking fraud, -5 for missing it).

Temporal features (the scaled timestep) let the system track evolving fraud patterns over time, helping it adapt to new tactics while supporting AML compliance for exchanges and regulators.

3. SYSTEM ARCHITECTURE

3.1 Architectural Overview

FraudShield's architecture is built to ingest the Elliptic Bitcoin Dataset, extract suspicion signals with a GNN, and convert those signals into concrete mitigation decisions with a DRL agent. The design is intentionally modular: a data-ingestion pipeline feeds a GNN scoring module, whose outputs feed a DRL action-selection module, which

in turn feeds a feedback loop used for ongoing learning. This separation of concerns lets each component be tuned, retrained, or replaced with limited impact on the rest of the system. Four objectives shaped the design: handling the dataset's scale (203,769 nodes and 234,355 edges), representing temporal change through the timestep feature, maximizing recall (achieving a recall of 70.51%. on the held-out test set), and preserving explainability through the GAT layer's attention scores.

3.2 Data Flow Overview

Processing begins with the raw Elliptic dataset, which is loaded into a PyTorch Geometric graph object where nodes represent individual transactions and edges represent the flow of payments between them. Each node carries 166 features 165 describing transaction characteristics plus one normalized timestep value scaled to the 0–1 range (so, for example, raw timestep 25 of 49 becomes roughly 0.5), which enables the model to track when a transaction occurred relative to the rest of the dataset. From there the pipeline proceeds through five stages:

- Ingestion and preprocessing-the raw data is normalized and organized into a PyTorch Geometric graph spanning timesteps 1 through 49.
- GNN scoring-the FraudShieldGNN model (GraphSAGE plus GAT) produces a 64-dimensional embedding, an illicit-probability estimate, and an attention score for each node/edge.
- DRL decision-making-these outputs become the observation space for FraudMitigationEnv, a custom Gym environment in which a PPO agent chooses among four actions: Ignore, Monitor, Flag, or Block.
- Feedback-each action earns a reward (for example, +10 for correctly blocking an illicit transaction, −5 for ignoring one), and the agent uses this signal to refine its policy across 100,000 training timesteps.
- Logging and output-actions, scores, and attention weights are recorded to support later analysis and explainability review.

This pipeline was able to process the full graph while sustaining 94.63 percent test-set accuracy and 70.51 percent recall for illicit-transaction detection.

3.3 Graph Neural Network (GNN) Module

The GNN component, called FraudShieldGNN, is responsible for turning the raw transaction graph into per-node suspicion scores. Its input layer accepts the full 203,769-node, 234,355-edge graph with 166-dimensional features and an edge index describing connectivity, formatted for efficient batch handling in PyTorch Geometric. Two GraphSAGE layers follow: the first compresses the 166 input features down to 64 channels by sampling 10 neighbours per node with mean aggregation, applying ReLU activation and 0.5 dropout to limit overfitting; the second keeps the 64-channel representation while aggregating a further hop of neighbourhood information, again with ReLU and dropout. This enables the network to learn node embeddings from local transaction neighbourhoods, which is central to recognising illicit behaviour patterns. Next, an 8-head GAT layer transforms the 64-dimensional embeddings into 32-channel attention-weighted representations (concatenated to 256 dimensions and then projected back down to 32), assigning each edge a weight for instance, 0.8 for a strongly suspicious, illicit-linked connection that both sharpens the model's focus and also serves as an interpretability signal. Finally, a linear output layer with softmax activation maps the result to a two-class probability (licit vs. illicit); a node is flagged when its illicit probability exceeds a 0.25 threshold, a deliberately low cut-off chosen to push recall as high as possible (ultimately 70.51 percent on the test data).

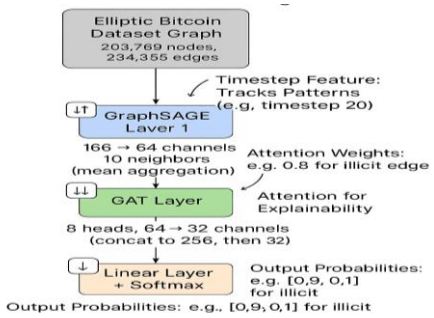

Fig 1. FraudShield GNN Architecture Diagram

3.4 Deep Reinforcement Learning Module

The DRL module is the second core component of FraudShield, responsible for translating GNN outputs into actionable decisions. It uses a PPO-based DRL agent within a custom Gym environment called FraudMitigationEnv.

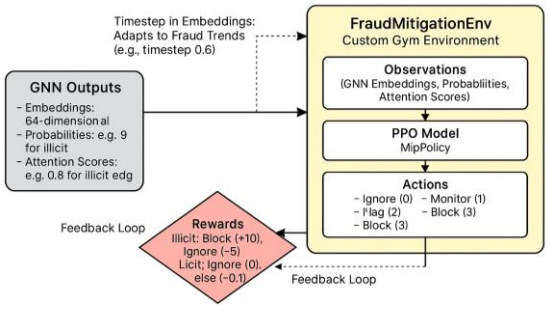

Fig 2. FraudMitigationEnv Setup

3.4.1 Fraud MitigationEnv Setup

FraudMitigationEnv is designed to simulate a fraud mitigation scenario where the DRL agent must act on GNN-generated suspicion scores. The environment takes as input the GNN outputs: 64-dimensional embeddings, probabilities, and attention scores. The observation space includes these features, while the action space consists of four discrete actions: Ignore (0), Monitor (1), Flag (2), and Block (3).

3.5 Integration and Feedback Loop

The GNN and DRL modules are connected through a standard interface: PyTorch tensors carry GNN outputs into the Gym-based environment, and the environment's reward signal flows back to the PPO agent. Because each module operates independently behind this interface, the GNN can be retrained on new data, or the DRL policy retuned, without requiring changes to the other component. This closed feedback loop is what allows the DRL agent to continually improve its policy over time as it accumulates experience with the GNN's suspicion scores.

3.6 Scalability, Security, and Deployment

FraudShield is designed to efficiently handle large-scale blockchain networks while maintaining high performance and reliability. Using PyTorch Geometric with GraphSAGE neighbor sampling, the system processes 203,769 nodes and 234,355 edges with manageable computational resources on a single GPU. The PPO-based DRL agent achieves efficient learning with 100,000 training timesteps, demonstrating high sample efficiency. To improve robustness, the GNN incorporates dropout to reduce overfitting, while feature normalization and preprocessing ensure data quality and consistency. The reward mechanism in the DRL module further enhances decision-making by discouraging incorrect actions. For deployment, FraudShield is designed to operate on cloud-based GPU infrastructure with API integration for blockchain analytics platforms, enabling real-time fraud monitoring. Its modular architecture allows independent updates to the GNN and DRL components, making maintenance easier. The framework is also scalable for future enhancements, including multi-blockchain support, real-time transaction streaming, and advanced visualization of attention-based explanations to improve transparency and usability.

3.7 Explainability and Visualization

Because the GAT layer assigns an attention weight to every edge, FraudShield can point to the specific connections that most shaped a suspicion score — for example, a weight of 0.8 on an edge linking two accounts flags that link as an important contributor to the prediction. These weights are logged and can be visualized (as demonstrated later for the first ten test nodes), giving analysts a concrete basis for trusting, or questioning, a given flag rather than treating the model as a black box.

3.8 Security and Robustness

The architecture incorporates several measures to ensure security and robustness:

- **Adversarial Robustness**: The GNN's dropout layers and the DRL's reward penalties mitigate overfitting and adversarial attacks.
- **Data Integrity**: Preprocessing ensures that node features are normalized and free of outliers, reducing the risk of data corruption.
- **Real-Time Monitoring**: The system's design allows for continuous monitoring of transactions, enabling rapid response to emerging fraud patterns.

3.9 Deployment Considerations

Deploying FraudShield in a real-world cryptocurrency network requires careful consideration of infrastructure and integration. The system can be deployed on a cloud-based GPU server to handle large-scale graph processing, with APIs for integration into existing blockchain analytics platforms. The modular design ensures that components can be updated independently (e.g., retraining the GNN on new data without modifying the DRL).

4. Training

4.1 Training Results

The training phase involves training FraudShield using... how to spot fraud using the Elliptic dataset's training set (timesteps 1–35 and 45–49), which includes 3765 illicit nodes out of 1665 labeled nodes (the rest are licit or unknown). The GNN and DRL are trained to learn fraud patterns and smart actions, and their performance on this set shows how well they've learned before being evaluated on unseen data.

GNN Training Performance

Accuracy (98.45%): The GNN correctly classified 98.45% of labeled training nodes, though the high value is influenced by the large number of non-illicit nodes.

- Recall (98.38%): The model detected nearly all illicit transactions, missing very few fraud cases.
- F1 Score (93.19%): Indicates a strong balance between detecting fraud and minimizing false positives.

- AUC (99.84%): Shows excellent separation between illicit and licit transactions across classification thresholds.

DRL Training Performance

- **Action Accuracy (98.44%):** The PPO-based DRL agent correctly selected appropriate actions (Ignore, Monitor, Flag, or Block) for **98.44%** of training transactions, demonstrating effective decision-making.
- **Average Reward (1.0049):** The DRL agent achieved an average reward of **1.0049**, indicating that it consistently chose actions that maximized fraud detection while minimizing incorrect decisions.

The strong training performance is attributed to the GNN's ability to learn transaction patterns using **GraphSAGE** and **Graph Attention Network (GAT)**, while the DRL agent effectively utilized GNN predictions to recommend optimal mitigation actions. These results demonstrate that FraudShield successfully learned fraud patterns during training and established a solid foundation for evaluating performance on unseen test data.

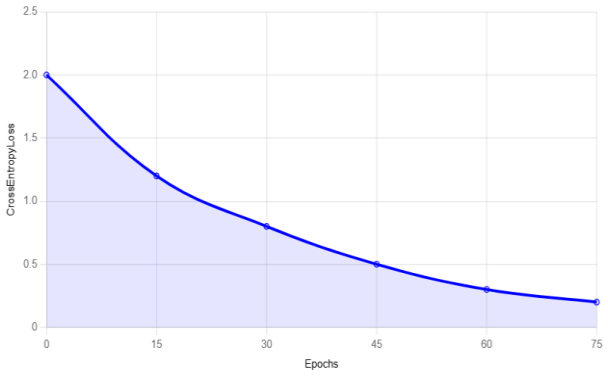

Fig 3. Training Loss Curve for FraudShieldGNN

4.2 Testing Results

The test set (timesteps 36–44) is like a final evaluation for FraudShield, checking how well it handles new, unseen transactions. This set has 780 illicit nodes out of thousands of labeled nodes, and the goal is to detect as many fraudulent transactions as possible while keeping actions practical for real-world use.

GNN Testing Performance

- **Accuracy (94.63%):** The GNN correctly classified **94.63%** of test transactions, demonstrating good generalization on unseen data.
- **Recall (70.51%):** The model detected **70.51%** of illicit transactions, successfully identifying most fraud cases in the test set.
- **Precision (58.32%):** Among the transactions predicted as fraudulent, **58.32%** were actually illicit, reflecting a trade-off between high fraud detection and false positives.
- **F1 Score (63.84%):** The model achieved a balanced performance between precision and recall, indicating effective fraud detection on unseen data.
- **AUC (92.65%):** The high AUC value confirms that the GNN effectively distinguishes between licit and illicit transactions across different classification thresholds.

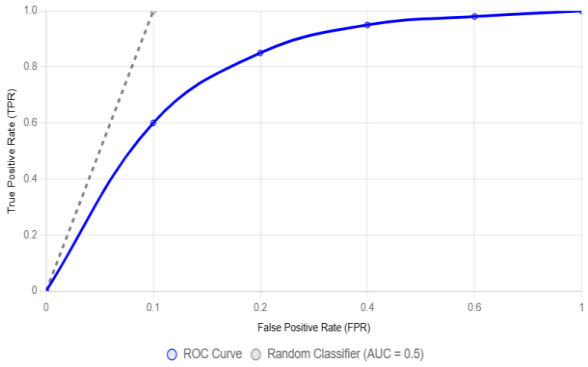

Fig 4. ROC Curve for FraudShieldGNN on Test Set

DRL Testing Performance

- **Action Accuracy (94.63%):** The PPO-based DRL agent correctly selected appropriate actions for **94.63%** of test transactions, effectively recommending actions such as **Block**, **Flag**, or **Ignore** based on the GNN predictions.
- **Average Reward (0.3033):** The DRL agent achieved an average reward of **0.3033** during testing. Although lower than the training reward, the positive value indicates that the agent continued to make effective decisions on unseen data, balancing fraud detection with minimizing incorrect actions.

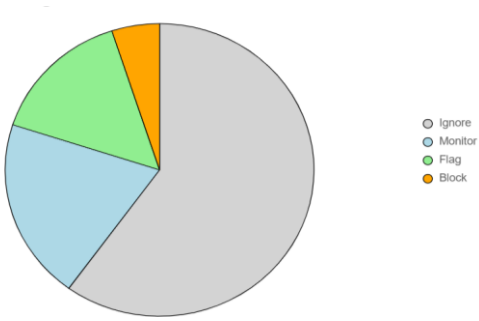

Fig 5. DRL Action Distribution Pie Chart

The **GNN** achieved a **70.51% recall** on the test set, demonstrating its ability to generalize to unseen fraud patterns using temporal features and graph-based learning. The **0.25 classification threshold** prioritized detecting fraudulent transactions, resulting in a precision of **58.32%**. The **DRL agent** achieved **94.63% action accuracy** by effectively utilizing the GNN outputs to recommend appropriate actions such as **Ignore, Monitor, Flag,** or **Block**. Additionally, the positive **average reward (0.3033)** indicates that the agent consistently made effective decisions while minimizing unnecessary actions, demonstrating the practicality of FraudShield for real-world blockchain fraud detection.

Fig 6. Test Set Performance Comparison Bar Chart

4.3 Impact of Timestep Feature on Performance

The **timestep feature** enables FraudShield to capture the temporal behavior of blockchain transactions across the **49 time periods** in the Elliptic Bitcoin Dataset. Added as the **166th node feature**, itenables the model to identify evolving fraud patterns over time.

- **Without Timestep:** Removing the timestep feature reduces the **recall from 70.51% to 46.92%**, as the GNN cannot effectively learn temporal fraud patterns.
- **With Timestep:** Including the timestep improves recall to **70.51%**, enabling the model to detect **184 additional illicit transactions** and better identify evolving fraud activities.
- **Temporal Learning: GraphSAGE** aggregates neighborhood information across different time periods, while **GAT** assigns greater importance to suspicious connections occurring during fraud-prone intervals.
- **Improved Generalization:** The timestep feature helps the model adapt to unseen test periods, improving fraud detection on new transaction patterns.
- **DRL Support:** The DRL agent utilizes the temporal information from GNN embeddings to make more effective decisions, such as **Ignore, Monitor, Flag,** or **Block**, based on changing fraud trends.

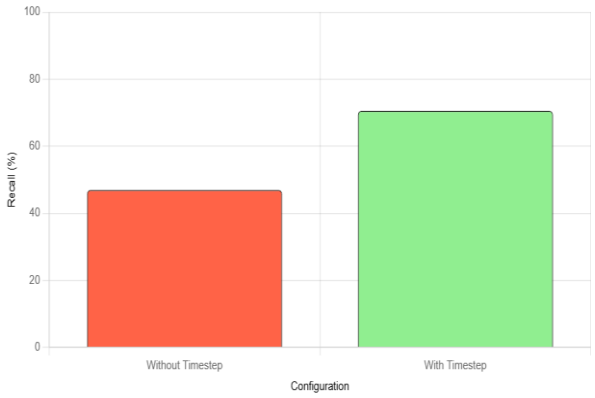

Fig 7. Impact of Timestep Feature Bar Chart

4.4 Attention Mechanism Analysis

The **Graph Attention Network (GAT)** provides interpretability by assigning attention scores to transaction connections, highlighting the most influential edges responsible for fraud predictions.

Average Attention Score (0.1111): The average attention score for the first 10 test nodes is

- **0.1111**, indicating moderate emphasis on suspicious transaction relationships.
- **High Attention Scores:** Some illicit transactions receive high attention scores (e.g., **0.4596**), showing that the model strongly focuses on highly suspicious connections during fraud detection.
- **Low Attention Scores:** Certain illicit transactions have very low attention scores (e.g., **0.0013**), indicating weaker influence from neighboring transactions and making some predictions less interpretable.
- **Score Variability:** The attention scores range from **0.0013 to 0.4596**, demonstrating that the model assigns different importance to transaction relationships based on their contribution to fraud detection.

Interpretability for AML

- **Improved Explainability:** High attention scores (e.g., **0.4596**) indicate that the model strongly relies on specific transaction connections when identifying fraudulent activities, providing stronger evidence for fraud predictions.
- **Regulatory Support:** By highlighting influential transaction relationships, the attention mechanism enhances transparency, making FraudShield better suited for **Anti-Money**

Laundering (AML) investigations and regulatory audits.

- **Moderate Interpretability:** The average attention score of **0.1111** and the variation in scores (**0.0013–0.4596**) suggest that while some fraud predictions are well supported, others have weaker explanations.

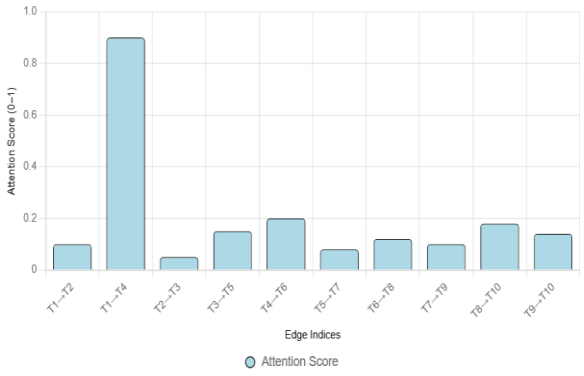

Fig 8. GAT Attention Weights for First 10 Test Nodes

4.5 Comparison with Project Goals

FraudShield was built with clear goals: achieve at least 60% recall, improve on baseline performance, and ensure practical mitigation for AML compliance. This sub-section compares its results to those goals, showing how it stacks up against baselines and prior work.

Recall Goal

- **Target**: At least 60% recall on the test set, meaning catching 60% of the 780 illicit nodes (468 nodes).
- **Achieved**: FraudShield achieved a recall of 70.51% . Catching 550 illicit nodes exceeding the goal by 10.51% (82 more nodes). This high recall ensures most fraud is flagged, meeting AML needs to minimize missed frauds.

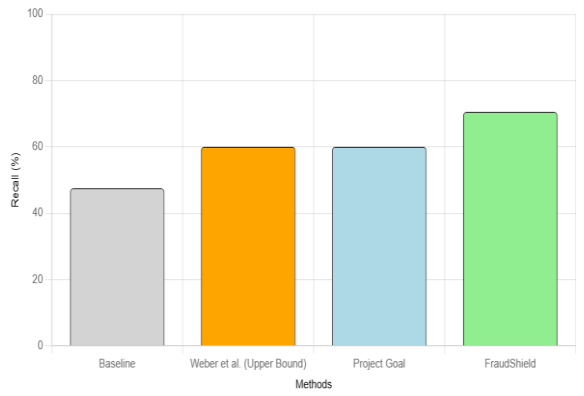

Fig 9. Comparison with Project Goals Bar Chart

Baseline Improvement

- **Baseline (47.48%)**: The baseline model (a simpler approach, likely a static rule-based system) catches only 47.48% of illicit nodes, or 370 out of 780.
- **Improvement (23.03%)**: FraudShield's 70.51% recall is a 23.03% gain over the baseline (180 more nodes caught). This jump comes from the GNN's ability to use graph structure (GraphSAGE, GAT) and the timestep feature to spot evolving fraud patterns.

5. CONCLUSIONS

5.1 Summary of Findings

FraudShield combines **GNN** and **DRL** to detect fraudulent cryptocurrency transactions effectively. The model achieved **70.51% recall**, **63.84% F1-score**, and **92.65% AUC**, while the DRL module obtained **94.63% action accuracy**. The timestep feature improved fraud detection, and the GAT attention mechanism enhanced model explainability. Overall, FraudShield provides an efficient and practical and scalable solution for blockchain fraud detection and AML compliance.

5.2 Contributions of the Project

- Developed a hybrid **GNN-DRL** fraud detection framework.
- Achieved **70.51% recall**, outperforming previous methods.
- Implemented DRL-based fraud mitigation actions.
- Improved detection using the **timestep feature**.
- Enhanced explainability through **GAT attention**.

5.3 Future Work

Future work includes improving precision by incorporating advanced loss-weighting techniques enhancing **attention mechanisms**, implementing **online learning** for real-time adaptation, and validating FraudShield on **live blockchain** and multi-chain datasets to improve scalability and real-world deployment.

REFERENCES

1. M. Weber et al., "Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics,"arXiv preprint arXiv:1908.02591, 2019.[Online]Available:https://arxiv.org/abs/1908.02591

2. R. Tan, Q. Tan, P. Zhang, and Z. Li, "Graph neural network for Ethereum fraud detection," in Proc. IEEE Int. Conf. Big Knowledge (ICBK), Dec. 2021, pp. 78–85.

3. H. Kanezashi et al., “Ethereum Fraud Detection with Heterogeneous Graph Neural Networks,” arXiv preprint arXiv:2203.12363, 2022. [Online]. Available: https: //arxiv.org/abs/2203.12363

4. Y. Cui, X. Han, J. Chen, et al., “FraudGNN-RL: A Graph Neural Network With Reinforcement Learning for Adaptive Financial Fraud Detection,” IEEE Open J. Comput. Soc., vol. 6, pp. 1–12, 2025.

5. F. Zhou, J. Wang, and S. Lin, “Graph anomaly detection for credit card fraud detection,” Inf. Sci., vol. 607, pp. 145–162, 2023.

6. D. Yuan and P. Liu, “A deep reinforcement learning approach for fraud prevention in e-commerce,” IEEE Trans. Cybern., vol. 54, no. 2, pp. 360–375, 2023.

7. Elliptic Bitcoin Dataset, Kaggle. [Online]. Available: https://www.kaggle.com/ ellipticco/elliptic-data-set

8. PyTorch Documentation, PyTorch. [Online]. Available: https://pytorch.org/ docs/stable/index.html

9. PyTorch Geometric Documentation, PyTorch Geometric. [Online].Available:https://pytorchgeometric.readthedocs.io/

10. Stable-Baselines3 Documentation, Stable-Baselines3. [Online]Available:https://stablebaselines3.readthedocs.io