International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

A Survey on Graph Neural Network Approaches for Fraud Detection in Blockchain Networks

Sirijan[1], Dr. J.C. Miraclin Joyce Pamila[2]

[1]Department of Computer Science and Engineering, Government College of Technology, Coimbatore, India
[2]Head of the Department, CSE, Government College of Technology, Coimbatore, India

Abstract
Due to the fast growth of financial ecosystems supported by the blockchain technology, there is a corresponding increase in the amount of malicious activities like phishing attacks on the Ethereum platform or money laundering using Bitcoin. Standard approaches based on machine learning models and rules can no longer cope with such problems since their nature cannot be modeled using the relational and temporal graph structure that lies within blockchain transactions. Graph Neural Networks have proved themselves to be quite effective in this area by supporting aggregation processes on multi-hop neighborhoods as well as modeling both the attributes and structure of the graph. The aim of this paper is to present a survey on recent developments concerning GNNs for blockchain fraud detection, which have been published between 2022 and 2026. Moreover, several important issues like architecture types, namely homogeneous, temporal, heterogeneous, and hybrid ones, are discussed alongside with the class imbalance problem, for which focal loss and oversampling methods are considered. Finally gaps in research are highlighted, providing the directions for solving them.

Keywords: Graph Neural Networks, Blockchain Fraud Detection, Ethereum Phishing, Bitcoin Illicit Transactions, Temporal Graph Networks, Heterogeneous Graphs, Class Imbalance.

I. Introduction

Blockchain technology has revolutionized the process of recording and validating transactions. By offering a decentralized, immutable, and transparent ledger, it enables economic exchanges without central intermediation. Unfortunately, this makes blockchain technology prone to fraudulent practices. The phishing on Ethereum where the criminals use the imitation accounts to trick victims into sending their private keys or tokens can result in losses worth hundreds of millions of dollars per year. In addition to phishing, money laundering schemes operate on Bitcoin using a series of transactions to hide the origin of the transferred amounts.

Traditional ways of detecting fraud include rule-based systems and machine learning algorithms operating over manually crafted attributes for each transaction or entity involved in it. Though such solutions demonstrate decent results when used in well-formulated contexts, they cannot model the relationships between the elements in the graph as well as the connections they make. The latter often bear much of the discriminative power that allows identifying fraudulent intent. For instance, a single phishing account may look unremarkable, but the fact of its frequent transactions with multiple accounts followed by a fast dispersal of the stolen assets to a few other accounts is easily discernible in a transaction graph.

To resolve the issue, GNNs process graph data by leveraging neighborhood feature aggregation, generating representations that incorporate local and global features simultaneously. From their inception, different types of GNNs, such as Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), GraphSAGE, and their time-evolving and heterogeneous versions, have been extensively utilized for various node classification and graph classification problems. Applications in blockchain fraud detection based on GNNs have significantly increased during the period from 2022 to 2026, making it an emerging research domain requiring a comprehensive survey.

This survey paper examines eleven seminal papers on GNNs for blockchain fraud detection. In addition, six research questions that have not yet been fully answered by previous studies are identified, and specific avenues for future work are suggested. This paper is organized as follows: Section II surveys previous studies. Section III highlights research gaps. Section IV concludes the survey.

II. Existing Work

1. Ethereum Phishing Scam Detection Based on Data Augmentation Method and Hybrid Graph Neural Network Model

DA-HGNN, a new approach developed by Chen et al. (2024), solves both problems related to class imbalance and inadequate feature learning in Ethereum phishing attacks identification. It starts with the creation of eleven features representing the number of transactions, their amounts, and other network-related metrics. Next, the authors use the sliding window sampling technique for each node's chronological transactions to produce more training samples that increase the representation of minority phishing nodes while not generating artificial graph structures. This approach combines graph convolution operations to aggregate the local network structures and graph attention operations to assign weights to neighbors adaptively. They test the proposed algorithm on three different Ethereum datasets obtained through the Etherscan API, including all transactions from January 2022 to April 2024.

The primary advantage of the DA-HGNN algorithm is its capability to treat data augmentation as a critical stage of the entire pipeline. It uses sliding windows on real-time transaction sequences, thus preserving the order of transactions which would be ignored by random oversampling techniques. Their study shows that DA-HGNN consistently outperforms GCN, GAT, and GraphSAGE models in terms of the F1 score and recall.

One clear downside is that the sliding window algorithm repeatedly analyzes the same node's transactions across many windows, thereby posing a possibility of leaking information during the train/test split process. The eleven designed features also neglect any metrics related to gas consumption or the variety of token types, both of which provide valuable signals in more modern phishing attacks. Finally, the neighborhood aggregation of the model stops at first-degree neighbors, possibly failing to capture multi-hop relations during organized phishing attacks.

2. Ethereum Phishing Detection Based on Graph Neural Networks (TransWalk)

The framework proposed by Xiong et al. (2023), TransWalk, is a two-stage approach which involves using a direction-sensitive random walk method and multi-scale feature extraction techniques. The random walk technique in TransWalk is specific for directed graphs and ensures that edge direction is preserved while extracting neighborhoods in comparison to other works, where the graph is symmetrized in order to simplify computation. This is significant since phishing nodes are represented as asymmetrical nodes within the transaction graph and operate as sinks receiving transactions from numerous victims and

sources transferring them to further destinations, respectively. As a result, symmetry destroys the directionality of edges, hence, making the task harder for GNNs.

Direction preservation can be considered the technical novelty of this paper, as other phishing detectors based on GNNs symmetrize their input graphs. Consequently, TransWalk captures transactional movement patterns that are essential for identifying phishing accounts as well as intermediary mixer nodes and layered chains. Experimental evaluations show improvement in the recall and F1 scores compared to other direction-agnostic methods.

Incorporating the random walk component adds complexity due to the need to choose the right values of the walk length, restart probability, and number of walks per node, all of which are hard to fine-tune and not investigated in the study. Moreover, directed graphs often include dead-end nodes that have either outgoing or incoming edges, leading to premature end of random walks.

3. Phishing Scams Detection via Temporal Graph Attention Network in Ethereum

Temporal models have been justified by Wang et al. (2023) in relation to the fact that phishing attacks on the Ethereum network follow a specific life cycle, including rapid creation of accounts, a high number of malicious transactions aimed at the victims, and subsequent dormancy until the funds are distributed. Static GNN models aggregate all past edges equally without considering their importance based on recency, failing to recognize such a pattern in time. The temporal GAT architecture solves this problem by attaching an exponential decay function for all the edges as a learnable parameter and using the most recent edges for attention weights computation.

The mechanism of time-decay is understandable since it directly incorporates the logic that the behavior occurring closer to the present moment is more likely to represent an individual's current intentions compared to historical data. It works well in line with how phishing accounts are operated in real life, being relatively temporary. Our experiments conducted on a dataset of roughly 3,000 phishing accounts obtained from Etherscan reveal impressive recall gains over the static version of GAT without a substantial loss of precision, demonstrating that incorporating temporal dimension increases performance during the period of fraud execution.

Our model uses only one rate of decay for all the three transaction types, namely ETH transfers, internal calls, and token transfers, although each of them operates on its time scale significantly. For instance, a phishing account can make low-value token dusting attacks in weeks combined with making a high-value ETH transfer in minutes. One decay parameter is incapable of covering such behaviors. In addition, our model lacks an algorithm for solving the cold start issue. Newly created accounts are supposed to perform too few historical transactions to produce a noticeable temporal effect.

4. TTAGN: Temporal Transaction Aggregation Graph Network for Ethereum Phishing Scams Detection

TTAGN by Li et al. (2022) divides the intra-node temporal dynamics from the inter-node relational dynamics, an approach that sets it apart from other decay functions based temporal modeling approaches. The idea is for the GRU module in the model to process the transaction history data of the respective nodes in chronological order for each node to generate the temporal state that captures the evolution dynamics of the node. The temporal state generated by the node is combined with its static attributes to be processed using the graph attention mechanism with respect to neighboring nodes.

The architectural separation between self-temporal encoding and the neighbors' aggregations is theoretically sound and practically proven. The pattern of a fraudulent account becoming inactive, then very active, and then inactive again is different from the structure of its neighborhood, and combining

these two factors in one attention mechanism may result in one overtaking the other. This separation is made clear in one of the first works on detecting fraud in Ethereum networks, namely TTAGN, and further used by temporal graph neural network architectures in this field.

The use of the GRU mechanism necessitates the complete sequence of transactions for every node, resulting in a memory problem for nodes with many past transactions. In practice, the model uses a fixed-length sequence and discards transactions earlier than the window, implying a tendency to prefer recently occurring phishing attacks over those that lasted for a long time but with minimal intensity. Moreover, the model operates under an assumption that the transaction data is sorted and does not contain duplicates.

5. Ethereum Fraud Detection with Heterogeneous Graph Neural Networks

In Kanezashi et al.'s work (2022), the authors undertake one of the first attempts at systematically comparing different heterogeneous GNN architectures for the problem of Ethereum phishing detection. In their work, they build a heterogeneous transaction graph by differentiating several types of nodes: phishing accounts, exchange accounts, DeFi protocol contracts, miner addresses, and externally owned accounts; likewise, edges in the graph represent transfers of Ethereum, internal message calls, and transfers of ERC-20 tokens. The models examined include three homogeneous benchmarks — GCN, GAT, and GraphSAGE — and three heterogenous approaches — RGCN, HAN, and HGT.

The core conclusion of the paper is that RGCN, which uses individual GCN weight matrices for different edge types, achieves significantly better performance than any other model in terms of precision, recall, F1 score, and PR AUC. While HGT yields the best recall in terms of absolute value, it suffers from low precision. The authors suggest that HGT's ability to apply attention similar to the Transformer across meta-paths leads to overfitting to the most prevalent edge types, while RGCN benefits from having the freedom to learn different aggregations independently due to the type-based matrices.

All the models are tested on a static snapshot of the graph, implying that there is no regard for the order in which the transactions take place. The data set employed in this paper corresponds to the years surrounding 2020, and thus it fails to account for the latest tactics deployed by phishers who might be using DeFi protocols for transferring money and permit signature-based phishing attacks. There are no statistical significance tests performed in this study.

6. TokenScout: Early Detection of Ethereum Scam Tokens via Temporal Graph Learning

Wang et al. (2024), however, focus on a closely related yet distinct problem, namely, the early identification of potential frauds in the form of ERC-20 tokens rug pulls, honeypots, and Ponzi structures immediately after the tokens' deployment cycle. The underlying assumption here is that scam tokens tend to display unique temporal features in their transfer graph dynamics within the first hours and days after being deployed in terms of an excessively narrow distribution of tokens among a few predetermined wallets, unusual distribution of wallet holding ratios, and abnormal cycles involving the deployer and associated wallet addresses. Specifically, TokenScout represents the token transfer graph as a continuous-time dynamic graph and updates the node embedding through the message-passing procedure depending on the time passed since the latest interaction at each particular node.

The continuous-time dynamic graph representation of the model makes it possible to identify anomalies in token distribution patterns even in the first few hundred token transactions, thus preventing the emergence of victim losses on a massive scale. Thus, the ability to detect token scams at their early stages represents an additional distinguishing feature of TokenScout compared to other retrospective models that need a certain accumulation of fraudulent transactions for classification.

The initial problem lies with the concept drift: scam token schematics change quickly, and the traits that defined rug pulls at the beginning of 2023 might be different from the ones used towards the end of 2024. The temporal difference between scam implementation and classification as a scam by the regulators creates another problem as recent scams are not well-represented in the dataset. The machine learning model has difficulty in distinguishing genuine early-stage token clustering as a result of team holdings or investment locks versus pre-allocation traits of a future scam.

7. Tracking Phishing on Ethereum: Transaction Network Embedding Approach for Accounts Representation Learning

According to Lin et al. (2023), phishing detection is formulated as a problem of representation learning, wherein manually crafted features are weak since the malicious parties can analyze published thresholds to learn how to circumvent them. In contrast, the authors suggest training account representations based on transaction graph structures through the use of a modified node2vec algorithm designed specifically for directed, weighted transaction graphs on the Ethereum network. This modified random walk method considers not only the transaction amount but also the number of interactions with neighboring accounts to ensure economically relevant connections.

The key strength of the proposed representation learning technique goes beyond its performance in detecting existing phishing accounts. As an added benefit, the obtained embedding space allows performing similarity searches, which makes it possible to detect new phishing accounts by their proximity to labeled ones in terms of embedding space coordinates.

The absence of fraud detection loss during the embedding process leads to the learning of features of no importance for fraud detection, such as account age and industry. Retrieval based on similarity is especially vulnerable to this problem – if phishing accounts happen to be grouped close to a collection of non-fraudulent accounts in the space of embeddings because of a feature not relevant to the fraud detection problem, the retrieval algorithm will produce false-positive results that are hard to diagnose.

8. Temporal Graph Networks for Graph Anomaly Detection in Financial Networks

In Cai et al. (2024), authors employ the Temporal Graph Networks (TGN) architecture by Rossi et al. for the purpose of financial fraud detection with a loan guarantee network serving as the testing domain. TGN preserves a continuous memory state for every node which is updated at each interaction with other nodes. Specifically, the message function generates an embedding summarizing each interaction based on the features of the two nodes in question and the time passed from their last interaction, and a GRU-based memory update module incorporates that into the memory state of each interacting node.

The key capability of the memory module to capture the historical behavior patterns serves as a basis for its superiority compared to conventional GNNs where the historical context is ignored and the fixed-window temporal models where all interactions before a certain window size are discarded. For example, whether one node had already interacted with some account even several months ago can become an important feature in fraud detection. The paper illustrates that TGN substantially improves the performance over static GNN baseline methods on the AUC-ROC score measure.

However, there are key structural differences between the domain of guarantee network evaluations and the domain of blockchain transaction network evaluations: The guarantee networks are sparse, semantically meaningful, and stable over time, while the blockchain transactions are dense, untraceable, and capable of changing in character within minutes due to the adaptability of fraudsters. Notably, no cryptocurrency-based dataset was used for evaluation by the authors, leaving the memory effect of TGN in the face of class imbalances an unanswered question.

9. Who Will Be Hooked? A Phishing Fraud Detection Model Based on Dynamic Graph Temporal Feature Coding in Ethereum

Li et al. (2025) introduce PFD-TF, which uses the approach of timestamp encoding through sinusoids for detecting phishing attacks on Ethereum. Instead of applying a time decay function or recurrent neural network to encode timestamps, PFD-TF encodes Unix time stamps into high dimensional sinusoids similar to those used in the Transformer model using a series of frequencies over several orders of magnitude. This timestamp encoding can capture both short-period and long-period temporal trends in a unified way; that is, it can simultaneously detect both transactions in bursts within hours and dormant phases in weeks without being biased towards recent transactions.

The computation involved in encoding timestamps through sinusoids is simple since there is no need for any recurrence and iterative processing of sequences, nor do the transactions have to come in chronological order, which makes batching easier. Additionally, this paper proposes a Graph Attention Layer that gives more attention weight to neighboring nodes according to their transaction similarity as well as recentness. Sinusoidal embeddings are useful where the timestamps are evenly distributed, but phishing accounts generally exhibit behavior that is highly uneven in nature, where there are prolonged periods of dormancy, followed by brief periods of activity, and such behaviors may not necessarily be captured by a system where sinusoidal embedding depends on regularly recurring cycles. Another limitation of the framework is that all kinds of transactions are considered to be homogenous entities.

10. Bitcoin Money Laundering Detection via Subgraph Contrastive Learning

By viewing the detection of money laundering activities using Bitcoins as a subgraph-level representation learning task, Chen et al. (2024) assert that this is essentially a problem that entails working with illicit groups rather than detecting laundering behavior at the level of transaction and wallet nodes. The proposed method creates ego-subgraphs for labeled nodes and employs a contrastive learning approach whereby the representations of the illicit subgraphs are brought closer to each other but separated from licit subgraphs. Positive examples are obtained by comparing the representations of the subgraphs of an illicit node after random augmentation and negative samples through licit nodes.

In essence, the adoption of this group perspective is highly justified since it is a fact that money laundering activities require coordination and collaboration among several accounts. This implies that the behaviors observed at the level of a group are easier to distinguish compared to those exhibited by individual nodes. It should be noted that the proposed approach provides a clear advantage since it does not need all the nodes in a subgraph to be labeled.

The problem of designing subgraphs using the same hop radius presents an important challenge for subgraph design. On one hand, if the hop radius is too small, it fails to capture laundering processes involving many transactions; on the other hand, if it is too big, then several meaningless nodes will get included in the graph structure. In fact, the augmentation techniques used here might end up destroying exactly what makes a subgraph illicit.

11. Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions (MDST-GNN)

The authors Chen et al. (2025) present a novel model named MDST-GNN with multi-distance graph convolution as the key contribution in its architecture. Instead of adding up the output of all the neighborhoods as in traditional GCN, MDST-GNN utilizes distinct weight matrices for the one-hop, two-hop, and three-hop neighborhoods before merging them through a learnable fusion module. The logic behind this approach lies in the fact that the radius of the informative neighborhood differs according to

the type of fraud, where, for example, isolated phishing transactions will only have contact with victim wallets via one-hop neighbors, but complex layering scams would necessitate analysis of three or more hops to be identified.

The temporal part adds to the spatial multi-distance framework with a periodic timestamp encoding strategy based on sine and cosine transformations of several periods defined in advance. In contrast to learnable time encoding, the advantage of such an approach consists in its robustness against temporal granularity coarsening, which is crucial in practice because on-chain timestamps are susceptible to tampering by miners within a limited period. MDST-GNN demonstrates AUC-ROC equal to 0.952 when evaluated on the Elliptic Bitcoin dataset with coarse temporal granularity, whereas TGN obtains AUC-ROC equal to 0.921.

The multi-hop weight matrices increase the parameter count by a factor of the number of hops taken into account, making it more likely to overfit with small training sets. The periodic representation method forces the user to provide the set of periods beforehand, which might not be generalizable to other data sets depending on the transaction rates. The model has been assessed exclusively using the Elliptic Bitcoin dataset and has yet to undergo testing on Ethereum transaction networks, which may exhibit distinct graph diversity.

12. On the Use of Heterogeneous Graph Neural Networks for Detecting Malicious Activities in Cryptocurrencies

Ferretti et al. (2024) compare four heterogeneous graph neural network (GNN) models on the extended Elliptic++ data set, which includes additional node labels representing wallet addresses along with three interaction edges: transfer of money between transactions, connections from wallets to transactions, and co-spending edges between wallets. Four different models examined include heterogeneous GAT, heterogeneous GraphSAGE, HGT, and HAN. The authors test transaction classification accuracy along with account classification accuracy simultaneously, which is a rare but useful approach because in practice, anti-money laundering (AML) investigators need to classify both transactions and accounts at once, and one might not be the better choice than the other.

Heterogeneous GraphSAGE obtains the highest average F1 score for transaction classification, and HGT reaches perfect recall for account classification. It is interesting that GraphSAGE, which uses neighborhood sampling, performs better than HGT, which uses transformers-based self-attention, for transaction classification, just like Kanezashi et al. showed for Ethereum transactions. This implies that type-conditioned aggregation might be more efficient on this dataset than attention mechanisms.

None of the models employ any class imbalance correction techniques; all models are trained with the same standard cross-entropy loss on highly unbalanced data. In the case of HGT, it seems possible that its perfect recall in terms of identifying the wallets category could be attributed to a bias that makes it predict the minority class in all cases, instead of having discriminatory power. No confidence intervals or p-values are provided, nor any analysis performed across multiple splits of the dataset.

III. Identified Research Gaps

Based on the above review of literature, there are certain important research gaps that need to be filled for the development of efficient blockchain fraud detection based on graph neural networks.

1. Limited Cross-Chain Generalization

Research Gap: Almost all the currently available methods for detecting fraud using GNNs have been trained and validated on one particular blockchain only, either Bitcoin or Ethereum, without any analysis

carried out on newer chains like Solana, Polygon, or Avalanche. The attackers generally switch chains through bridges to hide their tracks, but none of the methods described in the survey consider cross-chain transactions. A method that works well to detect phishing attacks on Ethereum will fail to detect any fraud when run on Polygon because of differences in feature distributions, transaction speeds, and economics.

How the Gap Can Be Filled: There is an intuitive direction to explore. Using transfer learning and domain adaptation, one could train the same model on the large and well-labeled Elliptic Bitcoin dataset and then tune the model by applying graph domain adaptation to smaller datasets related to other blockchains such as Ethereum and Solana. With the creation of cross-chain graphs containing bridge transactions as inter-domain links, multiple blockchains can be trained together, thus sharing representation while learning different classifiers for each domain.

Scope of the Contribution: A benchmark dataset that contains labeled illegal addresses that cover multiple chains, where there is a link between the chains, would be ideal for conducting cross-chain generalization studies. There needs to be an evaluation framework that includes zero-shot transfer, which assesses the ability of a model trained in one chain to identify fraud in another chain without fine-tuning.

2. Adversarial Robustness to Graph Manipulation

Research Gap: GNNs are well-known for being vulnerable to attacks based on the manipulation of graph topology. With respect to the task of detecting fraud in blockchain, an attacker familiar with the algorithm's mechanism of aggregating neighbors will be able to introduce or delete edges in such a way that the phishing account will look like a legitimate one in the graph. For instance, in case an account is discovered due to its large fan-out ratio, the funds may pass through a series of intermediate accounts with small fan-outs. This manipulation was not tested by any of the reviewed works.

How the Gap Can Be Filled: Training schemes based on adversarial learning which have been modified for graph neural networks like RobustGCN or Graph Adversarial Training could be introduced into the pipeline of fraud detection. Techniques which are certifiably robust in the sense that they offer provable guarantees of detection performance in the case of structural perturbation of graphs represent a much stricter approach but require significant computational power. A pragmatic way forward would be to conduct an empirical investigation of the impact of different types of structural attacks on the performance of current methods.

Scope of the Contribution: A unified standard test for assessing adversarial robustness in GNNs for blockchain fraud, defining both attack types and evaluation metrics, will enable future methods not only to demonstrate robustness but also to allow comparison between different architectures regarding their robustness properties.

3. Concept Drift and Temporal Non-Stationarity

Research Gap: Patterns in blockchain fraud attacks change very quickly, with attackers adapting to detection techniques. Common phishing attacks in 2021, like basic impersonation sites collecting MetaMask authentication information, have been largely outmoded by 2024 by attacks based on permission signatures and approvals. Therefore, models trained using past labels for fraudulent activity could miss new types of attacks that were not part of their training data. None of the reviewed studies analyze the decline in model performance when there is a gap between when training data was collected and when the model was evaluated.

How the Gap Can Be Filled: Continuous learning architectures that enable models to learn new labelled instances gradually without erasing old information fit perfectly within the definition of the drift problem. The time-series validation approach, which trains models on data in period T and validates them on period

T+k, can measure the speed of degradation and set up rational re-training schedules based on facts. An anomaly detection system that does not need training data can be used alongside other triggers to detect structurally different accounts that do not conform to existing fraudulent behaviour.

Scope of the Contribution: What is absolutely necessary is an Ethereum phishing longitudinal benchmark corpus that contains at least three years of such phishing attacks along with time-stratified evaluation splits and annotations for the types of attacks that took place over those three years.

4. Explainability and Regulatory Compliance

Research Gap: The most common types of GNN fraud detectors are typically black-box models that provide only a prediction score without any human-comprehensible explanation of the reasons behind the decision making. This represents a substantial hurdle to adoption in financial compliance environments, where laws, such as the Anti-Money Laundering law in many countries, require the rationale for freezing accounts or rejecting transactions to be recorded in documentation. Existing GNN explainability algorithms like GNNExplainer and PGExplainer have not been widely applied in blockchain fraud detection studies yet.

How the Gap Can Be Filled: Explanation methods adapted to the blockchain domain need to specify the exact transactions, parties involved, or time-related features that had the largest impact on the classification of fraud. The explanations provided should not only meet criteria like fidelity and faithfulness as surrogate measures, but also undergo testing in user studies with AML specialists, who could validate whether explanations provide any help in their investigative work. Explaining decisions based on SHAP values calculated on GNN predictions is an option worth considering, as it does not involve altering the network architecture.

Scope of the Contribution: A standardized explainability testing suite for block chain fraud detection GNNs with metrics tailored to AML regulations would significantly hasten research in this field while making it easier for regulators to accept GNN fraud detection systems.

5. Scalability to Production-Scale Transaction Volumes

Research Gap: Each of the Bitcoin and Ethereum blockchains handles hundreds of thousands to millions of transactions on a daily basis, resulting in graphs that are too big to perform full-graph GNN training computations. The biggest graphs reported in any of the reviewed literature are those containing no more than about half a million nodes. Neighborhood sampling approaches like GraphSAINT and Cluster-GCN have been proposed in other applications where large graphs are involved, yet they have never been tested for their effectiveness in detecting fraud from the blockchain.

How the Gap Can Be Filled: The use of mini-batch sampling techniques modified to support directed, heterogeneous, temporally structured graphs as dictated by the requirements of the blockchain environment can significantly lower training costs and maintain detection accuracy. Techniques that allow incremental learning by updating the model parameters based on each new incoming transaction, rather than periodic training over the entire history of the graph, will help save costs and time spent on each detection. Hierarchical graphing techniques that can work on the level of transactions groups will make the size of the graph smaller.

Scope of the Contribution: The experimental analysis of the system's scalability with regard to training speed, inference latency, memory overhead, and detection accuracy for various sampling and partitioning approaches at the scales equivalent to actual blockchain networks will create the empirical foundation necessary to make informed engineering choices when deploying anti-fraud mechanisms.

6. Insufficient Handling of Heterogeneous Edge Semantics

Research Gap: Some of the surveyed papers create heterogeneous networks, where nodes are of different types, but edges are homogenized, as all transactions are treated as one type of edges. However, each of these transactions has a unique economic interpretation, and each serves a specific purpose in the execution of fraudulent schemes. In particular, transferring simple ETH amounts, transferring ERC-20 tokens, calling smart contracts for internal purposes, and approving or permitting transactions have entirely different meanings, and mixing them all under one type of edges loses valuable information for detecting fraud.

How the Gap Can Be Filled: Complete heterogeneity in graph generation through the differentiation of all types of edges along with architectures like RGCN or HGT which train unique aggregation functions per each type of edges can help the model make use of differences in semantic meanings among categories of transactions. Relation-specific attention mechanisms can also be used to assign dynamic weights to edges rather than static weights depending on the edge type, thereby enabling the model to attend more to transaction categories significant for particular types of frauds.

Scope of the Contribution: A comparative study that quantitatively measures the effectiveness of edge semantic modeling in object detection using various types of edge semantic schemas in increasing order of complexity, starting from homogeneity to heterogeneity, using the same evaluation dataset will justify the need for the added complexity involved.

IV. Conclusion

The above survey covers 12 important papers on GNNs for blockchain fraud detection proposed in the years 2022–2026, encompassing the whole range of architectures, starting with static homogeneous GNNs through temporal, heterogeneous, and contrastive learning paradigms. Some general conclusions drawn from the literature can be stated. Firstly, graph structure itself is a valuable source of information on malicious intent beyond any features-based model, as all papers implementing a GNN architecture in this survey have shown superior results over their respective baselines based on purely engineered features. Secondly, temporality significantly boosts performance in phishing and money laundering detection tasks, as it reflects the dynamic nature of a typical fraud campaign, although the best way to exploit temporality depends on the particular fraud and the nature of the underlying transaction graph. Thirdly, heterogeneity yields better performance than homogeneity when there is indeed a difference in the kinds of nodes and edges in the underlying graph, which applies to the Ethereum blockchain.

There are six areas where further progress must be made before we can achieve adequate coverage of research issues:

generalisation across chains, adversarial robustness against tampering with the graph structure, concept drift and time-dependency, explainability and compliance, scalability, inadequate treatment of heterogeneous edges. The latter appears to be the issue that needs urgent addressing in practice, as accuracy is not enough for achieving regulatory compliance and even an infinitesimal improvement in F1 score will never replace an explanation to the compliance officer as to why the given account has been flagged.

The area of GNNs in blockchain fraud detection has come a long way since our review began, but there seems to be significant distance still to cover before academic SOTA can be considered adequate for real-world application, which must be robust, reliable, and interpretable by the compliance community.

References

1. Z. Chen, S.-Z. Liu, J. Huang, Y.-H. Xiu, H. Zhang, and H.-X. Long, "Ethereum Phishing Scam Detection Based on Data Augmentation Method and Hybrid Graph Neural Network Model," Sensors, vol. 24, no. 12, p. 4022, Jun. 2024.
2. A. Xiong et al., "Ethereum Phishing Detection Based on Graph Neural Networks," IET Blockchain, May 2023.
3. L. Wang, M. Xu, and H. Cheng, "Phishing Scams Detection via Temporal Graph Attention Network in Ethereum," Information Processing & Management, vol. 60, no. 4, p. 103412, 2023.
4. S. Li, G. Gou, C. Liu, C. Hou, Z. Li, and G. Xiong, "TTAGN: Temporal Transaction Aggregation Graph Network for Ethereum Phishing Scams Detection," in Proc. ACM Web Conference 2022, pp. 661–669.
5. H. Kanezashi, T. Suzumura, X. Liu, and T. Hirofuchi, "Ethereum Fraud Detection with Heterogeneous Graph Neural Networks," in MLG Workshop @ KDD 2022.
6. H. Wang, H. Li, Y. Liu, and Y. Xiang, "TokenScout: Early Detection of Ethereum Scam Tokens via Temporal Graph Learning," in Proc. ACM CCS 2024.
7. Z. Lin, X. Xiao, G. Hu, Q. Li, B. Zhang, and X. Luo, "Tracking Phishing on Ethereum: Transaction Network Embedding Approach for Accounts Representation Learning," Computers & Security, vol. 135, p. 103479, 2023.
8. L. Cai et al., "Temporal Graph Networks for Graph Anomaly Detection in Financial Networks," arXiv:2404.00060, 2024.
9. C. Li, R. Liu, Y. Zhang, N. Xie, and Q. Zeng, "Who Will Be Hooked? A Phishing Fraud Detection Model Based on Dynamic Graph Temporal Feature Coding in Ethereum," in Proc. BWTAC 2024, Springer CCIS 2277, 2025.
10. Z. Chen et al., "Bitcoin Money Laundering Detection via Subgraph Contrastive Learning," Sensors, vol. 24, no. 6, Mar. 2024.
11. L. Chen et al., "Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions," Advanced Intelligent Systems, 2025.
12. S. Ferretti, G. D'Angelo, and V. Ghini, "On the Use of Heterogeneous Graph Neural Networks for Detecting Malicious Activities: A Case Study with Cryptocurrencies," in Proc. ACM OASIS Workshop 2024.