Detecting Malicious Blockchain Transactions Using Graph Neural Networks

Samantha Tharani Jeyakumar[1(✉)], Andrew Charles Eugene Yugarajah[2(✉)], Zhé Hóu[1(✉)], and Vallipuram Muthukkumarasamy[1(✉)]

[1] Griffith University, Brisbane, Australia
jeyakumar.samanthatharani@griffithuni.edu.au, {z.hou,v.muthu}@griffith.edu.au

[2] University of Jaffna, Jaffna, Sri Lanka
charles.ey@univ.jfn.ac.lk

Abstract. The adoption of blockchain technology within various critical infrastructures is on the rise. Concurrently, there has been a corresponding increase in its misuse, primarily through the exploitation of its pseudo-anonymous characteristic. Encouraging blockchain adoption and improving security in the decentralised environment require techniques to detect wallets and/or smart contracts owned by malicious entities. Illegal activities such as dark market trades, money laundering, and receiving unlawful payments are performed by connecting various wallets or smart contracts in a meticulous way. A graph can be a potential representation to visualise such interconnections via various patterns, and graph-based data may represent the topological structure of the blockchain network. Recently, Graph Neural Networks (GNN) have been widely used for analysing the structure of complex networks and identifying patterns. This is the first work that considers a generalised graph representation for the Bitcoin and Ethereum networks and analyses their behaviour using a combination of heterogeneous GNN framework's GraphSAGE and Graph Attention Network (GAT). The classification results reveal that the proposed approach modestly improved Bitcoin network analysis, whereas Ethereum smart contract analysis needs further investigation in terms of incorporating other aspects of smart contracts, such as codebase, byte length, and lifetime features.

Keywords: blockchain · ransomewre settlement · ponzi smart contract · graph-based analysis

1 Introduction

Blockchain is a distributed and decentralised digital ledger technology that records transactions securely and transparently. Key properties of this technology include immutability, transparency, pseudo-anonymity, and decentralisation, making it suitable for various applications: tracking manufacturing in the supply chain, health record and insurance claim monitoring, peer-to-peer energy trading, and secure data sharing in IoT [4,9]. The pseudo-anonymity ensures the privacy of participants in blockchain networks. Malicious actors that are receiving ransomware or phishing payments in the form of cryptocurrency, involving dark-market trades, and dealing with money laundering exploited the pseudo-anonymous property to obscure their real identity from legal authorities or financial regulators. For instance, approximately $3.36 billion worth of dark web-related transactions concealed within Silk Road were Seized and those responsible were convicted in 2021, as reported by the U.S. Attorney. Such misuse holds the potential to gradually diminish public confidence in the widespread acceptance of blockchain technology. Additionally, these illegal activities present regulatory challenges in ensuring that the technology is not manipulated for malicious intentions. Preventing illegal activities and supporting the implementation of regularity schemes are urgently needed in the monitoring and analysis of blockchain networks. Large volumes and complex structures of blockchain transactions are significant limitations for the analysis. An efficient analysis needs a meaningful transformation for blockchain data that can inform the interconnection between wallets, smart contracts, and their transactions. A graph is a well-defined data structure for representing relations between different types of nodes and can reflect the interconnections via graph patterns [6,7,16,26].

Graph-based analysis can fall into three categories: node classification, edge classification, and graph classification. The graph-based analysis for blockchain transactions can be beneficial in terms of node classification to classify the behaviour of wallets, smart contracts, or transactions or graph classification to identify groups of wallets owned by mixing services or dark markets. Literature study has identified previous research work using graph-based representation and analysis to identify mixing services [24,27,31], dark market-related trades [20], and Ponzi schemes [32] in Bitcoin and Ethereum networks. The existing studies involved manual processes, domain knowledge, focused on a specific blockchain network, and inefficient resource utilisation. Heuristics-based analysis is mostly subjective based on the selected domain or attack. By considering these limitations and challenges this study made the following contributions.

1. The proposed study considered a generalised graph modelling known as a hypergraph, which allows the analysis of blockchain networks without concern for the different structures of transaction data.
2. Embedding feature generation for this study considered both raw and interconnection information of address-to-transaction and smart contract-to-transaction. This is useful to train a model by considering the self and relational features of nodes (wallet and smart contracts).
3. This is the first study performed on graph-based learning using heterogenous GNN frameworks by combining GraphSAGE and GAT GNNs. The result leads to the implementation of a heterogenous GNN model for real-time blockchain network analysis to identify suspicious behaviour of wallet or smart contracts.

The structure of the paper is as follows: Sect. 2 presents a critical review of related studies of GNN-based blockchain network analysis. Section 3 describes the proposed heterogenous GNN-based classification. Section 4 presents classification results for blockchain transactions and discusses the significance of the proposed GNN-based analysis. Finally, Sect. 5 concludes the paper.

2 Related Work

This section details existing research works related to suspicious transaction detection in blockchain networks. The recent research approaches used Artificial Neural Networks (ANN) [20], Deep autoencoder [24], Convolution Neural Network [22], Graph Convolution Networks (GCN) [30,32], and Random Forest [12,21] to classify malicious blockchain transactions.

Lee et al. [20] proposed a supervised learning approach using Random Forest (RF) and Artificial Neural Networks (ANN) to classify malicious Bitcoin transactions related to Silk Road dark market trades. Their ANN-based network is designed with an input layer, two hidden layers, and an output layer. Transaction features considered for the classification reflect the number of inputs, outputs, and their values. Noticeably, their classification does not consider the interconnection information between wallet and transactions.

Lihao Nan et al. [24] proposed an address graph-embedding feature-based approach to identify the community of mixing services on the Bitcoin network. They obtained graph embedding features using a deep auto-encoder and fed them into a k-means clustering to identify the community clusters. The local outlier probabilities [19] used in their approach identified nodes related to mixing services. The identified limitations in their approach are the local outlier probabilities method is much slower for large-scale graphs, there are no address-based features involved in the node embedding, and the experiment was not tested with real mixing data.

Mark Weber et al. [30] used GCN to classify binary class Bitcoin transaction network. Their experiment considered raw transaction features as well as transaction-to-transaction graph data for licit and illicit node classification. Their proposed GCN considered a two-layer, runs 1000 epochs employs the Adam optimizer with a learning rate of 0.001 and utilises an embedding vector size of 100. Considerably, their proposed approach is only applicable to the Bitcoin network.

Shanquing Yu et al. [32] proposed a graph convolutional network-based classification model to identify Ponzi scheme smart contracts using transaction networks. They obtained fourteen raw features of smart contracts and node-embedding features of transaction networks. A 32-dimension node embedding vector was obtained using a three-layer GCN architecture. Their classification considered supervised learning approaches: linear regression [23], support vector machine [13], adaptive learning rate optimisation [17], and random forest [8], network embedding-based approach: LINE [28], random-walk-based approaches: deepwalk [25] and node2vec [10]. They found that the combination of basic features with the GCN outperforms the other methods.

Lou et al. [22] proposed an improved convolution neural network to analyse the bytecode image of smart contracts to predict Ponzi schemes. Their proposed approach outperformed the supervised learning approaches Random Forest, support vector machine, XGBoost, and Isolation forest. Noticeably, their proposed preprocessing for bytecodes of each smart contract slows down when a large amount of training and testing data is used.

Xuezhi He et al. [12] proposed a decision tree-based supervised learning approach called Code and Transaction Random Forest (CTRF) to identify Ponzi contracts on Ethereum networks. Their experimental dataset considered word and sequence features of smart contract's code, and transaction features. The dataset was validated against supervised learning approaches KNN, CNN, decision tree, SVM, XGBoost, and CTRF. Their experimental results identified that the sequence feature of smart contract opcode and the transaction features improved model performance in identifying Ponzi contracts.

Lo et al. [21] proposed a GNN framework based on self-supervised Deep Graph Infomax (DGI) and Graph Isomorphism Network (GIN), with Random Forest (RF). Their proposed approach first constructs embedding vectors for Bitcoin transaction networks and then uses them as features to train RF to classify money laundering transactions. Results revealed that their proposed approach outperforms the traditional approaches and obtained a 0.828 F1-score.

The existing approaches stated above involved manual processes, domain knowledge, and high resource utilisation. Heuristics-based node labelling is primarily subjective based on the selected domain or attack. Considering these limitations and challenges, this research work provides an automated generalised graph modelling and GNN-based analysis framework to classify various anomalous behaviours of nodes in blockchain networks.

3 Methodology

This section details the proposed approach for Graph Neural Networks (GNNs) based analysis to classify malicious participants (wallet or transaction or smart contracts). The proposed approach includes four phases: data collection, data modelling, data pre-processing, and analysis, as described in Fig. 1. First, the data collection phase details the experimental data used for generalised graph modelling. Then data modelling phase explains feature extraction [14] of wallet, transactions, smart contracts and graph structure information via hypergraph [15]. Finally, the analysis details the GNN-based classification approaches and their outcomes.

3.1 Data Collection

Data collection describes the scrapping of blockchain transactions for normal and malicious activities in Bitcoin and Ethereum networks. The analysis of this research involves normal & ransomware settlement-based and non-Ponzi & Ponzi scheme-based transactions.

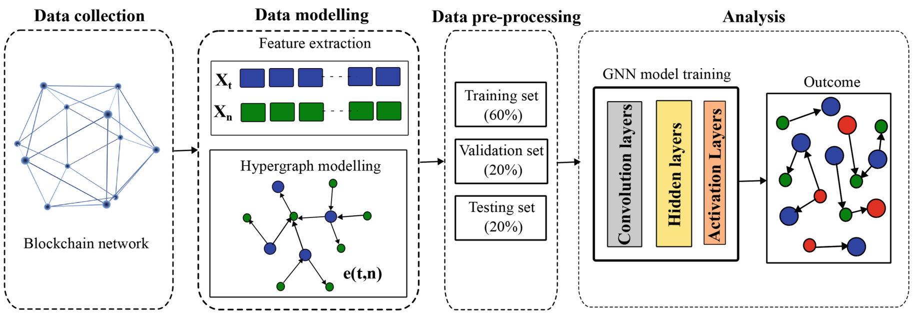

Fig. 1. Overview of the proposed methodology.

Bitcoin Ransomware Transactions. The labels of normal and ransomware settlement-based Bitcoin network wallets were referred from the BitcoinHeist [5] dataset, then recent 100 transactions corresponding with 16938 wallets were captured using public API [1].

Ethereum Ponzi Smart Contract Transactions. The labels of non-Ponzi and Ponzi smart contracts were referred from the public dataset [2] and the detailed information of smart contract transactions was obtained using public API [3]. The collected transactions include 200 Ponzi and 3590 non-Ponzi smart contracts.

3.2 Data Modelling

This section describes node features that are used as inputs for GNN-based analysis. The node features are derived based on raw transaction information and their maximum, minimum, mean, mode, median, and standard deviation measures. Tables 1, 2, and 3 detail major features of Bitcoin transactions, and wallets, and Ethereum smart contract transactions which are used as initial node properties during generation of graph embedding.

Bitcoin Transactions. The graphs of the Bitcoin network considered in this study included two types of nodes: transaction and wallet. A transaction contains nineteen features explained in Table 1, whereas a wallet consists of sixteen features shown in Table 2.

Ethereum Smart Contract Transactions. The graph of the Ethereum network considered in this study involves two types of nodes: transactions and smart contracts. A smart contract transaction contains five features detailed in Table 3, whereas a smart contract address involves sixteen features described in Table 2.

This research aims to propose a generalised approach to analyse any type of blockchain network. Hypergraph G_h is a generalised graph modelling for different types of blockchain transactions which represents the interactions between

Table 1. Features for Bitcoin transaction.

Features	Description
$inDegree$	number of incoming transactions (UTXOs)
$outDegree$	number of outgoing transactions
$totalInput$	total amount of Bitcoins received from other transactions (UTXO)
$totalOutput$	total amount of Bitcoin sent
$inout - ratio$	ratio between the number of inputs and outputs
$unique - out$	number of unique output addresses involved in a transaction

Table 2. Features for Bitcoin wallets and smart contract addresses.

Features	Description
$asASender$	total number of times a specific address as a sender
$asAReceiver$	total number of times a specific address as a receiver
$totalSpent$	total amount spent by a specific address
$totalReceive$	total amount received by a specific address

Table 3. Node features for Ethereum transactions.

Feature	Description
$betweeness_t$	betweenness centrality value between the transaction and the smart contract
$closness_t$	closeness centrality value between the transaction and the smart contract
$degree_t$	degree centrality value between the transaction and the smart contract
$eigenvector_t$	eigenvector centrality value between the transaction and the smart contract
$balance_t$	balance after the transaction

transactions and wallets or smart contracts. Edge information $e(u, v)$ indicates the type of transaction v (spent or received) corresponding with the wallet or smart contracts u. For this reason, we select a hypergraph that is proposed in the research [15] to extract the graph structure features. Further, these features facilitate training a single model to analyse various types of nodes in the blockchain network. This study considers Bitcoin transactions related to normal and ransomware settlements and Ethereum transactions related to non-Ponzi and Ponzi smart contracts. In the Bitcoin hypergraph, nodes are wallets (normal and ransomware-related) and their transactions. Edges are the type of transaction (spent or received) corresponding with wallets. Bitcoin transactions contain two major elements namely inputs and outputs. Inputs detail the Unspent Transaction Outputs (UTXOs), and outputs explain where the UTXOs are spent. In ransomware settlements inputs represent bitcoins received from the victims and the outputs indicate the wallet that accumulated all bitcoins from the victims. Figure 2 depicts an example of hypergraphs obtained for ransomware settlements, using the experimental data detailed in Sect. 3.1.

In the Ethereum hypergraph, nodes are smart contracts (non-Ponzi and Ponzi) and their transactions. Edges are the type of transaction (spent or received) corresponding with smart contracts. The main elements of Ethereum smart contract transactions are the address of the smart contract, details of the transaction that invoked the smart contract or invoked by the smart contract, and the amount spent/received during contract invocation. In Ponzi scheme settlements, ether was received from new investors and spent immediately for earlier investors. Figure 3 depicts an example of hypergraphs obtained for Ponzi smart contract settlements from the experimental data described in Sect. 3.1.

3.3 Data Pre-processing

The data pre-processing phase received node properties and graph structure-based data from the data modelling phase and divided them into training, validation, and testing sets. This study considered 60% of data for training, 20% of data for validation and 20% of data for testing. Table 4 presents the amount of data considered in each of the three sets of Bitcoin and Ethereum networks.

Table 4. Data allocation for classification.

Blockchain	Node type	Training	Validation	Testing
Bitcoin	Transactions	17122	6423	6354
	Addresses	10162	3388	3388
	Edge list	21896	7593	7213
Ethereum smart contract	Transactions	4707	1740	1766
	Addresses	240	80	80
	Edge list	4707	1740	1766

3.4 Grap Neural Network-Based (GNN) Analysis

This section details Grap Neural Network (GNN)-based analysis to classify malicious wallets or smart contracts and their transactions in blockchain networks as shown in Fig. 1. The proposed GNN-based approach contains three layers: the input layer, the GNN layer, and the prediction layer. A detailed description of each layer is as follows:

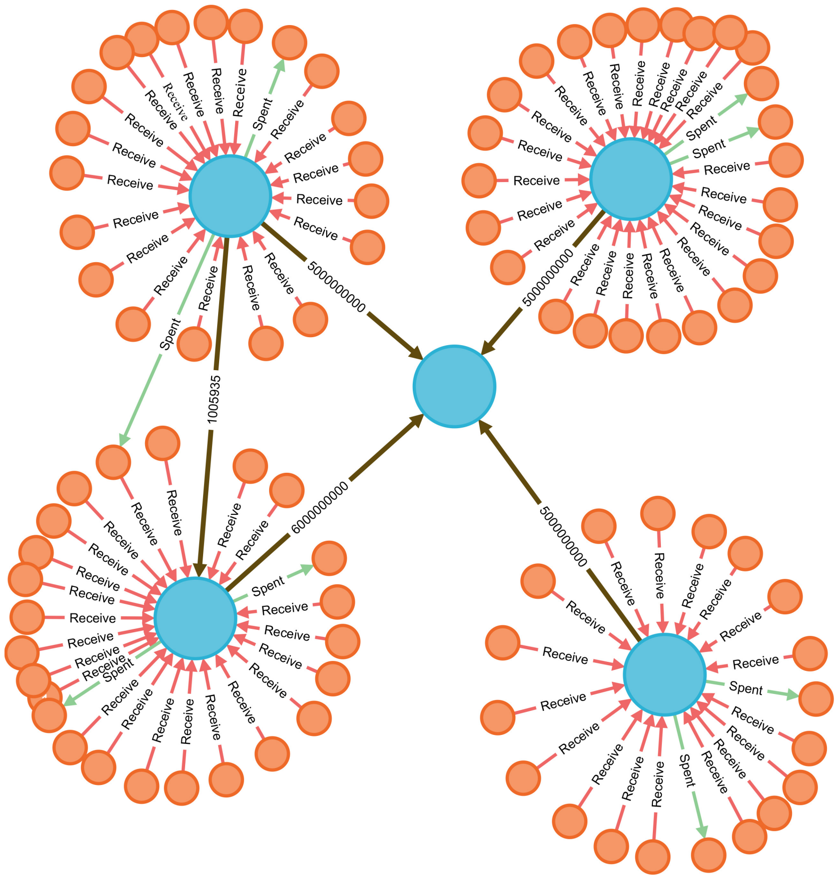

Fig. 2. Hypergraph for ransomware settlement. Here blue circles are transactions, orange circles are the wallets, red arrows represent inputs and green ones are the outputs. (Color figure online)

Input Layer: The input layer consists of node features, edge relations (spent or receive), and graph structure. The features $\alpha_t \in \mathbb{R}$ and $\alpha_n \in \mathbb{R}$ represent transaction features and wallet or smart contract features, respectively. These features are passed as an embedding to d-dimensional hidden features $h_i^{l=0}$ via simple linear projection. The edge information $\beta_{tn} \in \mathbb{R}$ consider the type of transactions (received or spent) corresponding with a wallet or smart contract. Similar to the node features, edge information is also considered for embedding to d-dimensional hidden feature $e_{ij}^{l=0}$. Finally, graph structure $e(t, n)$, from hypergraphs inputs as connection information where t is a spending or receiving transaction and n can be a wallet or smart contract.

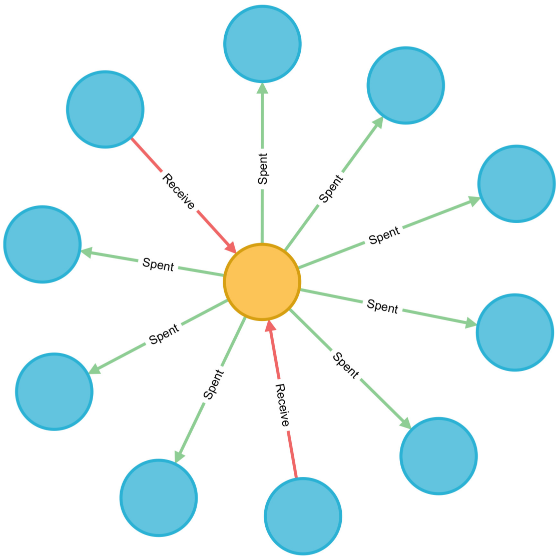

Fig. 3. Hypergraph for Ponzi smart contract. Here yellow circle represents the Ponzi smart contract, the blue circles represent the transactions, the red arrows spending transactions, and the green ones represent receiving transactions. (Color figure online)

***L*× GNN Layer:** The GNN layer consists of the L layer neural network. The L layer deep network corresponds to $L-$hop neighbourhood aggregation across the entire network. This is an iterative process, which can be visualised as a message-passing mechanism where each node (wallet/smart contract/transaction) receives updates from all its neighbours. The updated feature vector h_i^{l+1} for wallet or smart contract or transaction i is simply a function of its previous feature vector h_i^l and feature vectors of all its neighbours j as described in Eq. (1).

$$h_i^{l+1} = f(h_i^{l+1}, h_j^l : j \rightarrow i) \tag{1}$$

The blockchain network involves different types of nodes, changes dynamically, and is large in volume. By considering these constraints, this study used two heterogeneous GNN models Graph Attention Network Convolution (GAT) [29] and the GraphSAGE Convolution (SAGE) [11] are the extended versions of Graph Convolution Network (GCN) [18]. The selected convolution networks are capable of considering different types of nodes and their properties during the training. The architecture of GAT needs whole graph information for node representation (embedding), whereas SAGE generates node representation by sampling. Both SAGE and GAT can predict unseen nodes without re-training.

The proposed GNN-based approach for blockchain network analysis utilised both SAGE and GAT convolution layers. Learning details of SAGE and GAT are as follows:

- **GraphSAGE:** The GraphSAGE learns representations for each node by considering information from its neighbouring nodes. GraphSAGE achieved this learning in a two-step process: sampling and aggregation. The detailed description for GraphSAGE embedding is as follows:

$$h_v^0 \leftarrow x_v, \forall v \in V \tag{2}$$

$$h_N(v)^k \leftarrow f_{aggregate}(\{h_u^{k-1}| \ \ \forall u \in N(v)\}) \tag{3}$$

$$h_v^k \leftarrow \sigma(W^k(h_v^{k-1} \ \ || \ \ h_{N(v)}^k)) \tag{4}$$

$$h_v^k \leftarrow h_v^k/||h_v^k||_2, \forall v \in V \tag{5}$$

At first, all wallets or smart contracts and transactions in the hypergraph are initialised to their original feature vector x_v as described in Eq. (2). Then the feature aggregation at level k is processed as described in Eq. (3), here $N(v)$ denotes a list of neighbours of node v. Finally, the embedding vector at kth level updating via concatenates h_v^{k-1} and $h_{N(v)}^k$ embeddings of wallet or smart contract and transaction, where —— denotes concatenation, then takes a dot product of it and a learnable weight vector $\vec{W}^k$ and applies an activation function σ in the end as stated in Eq. (4). The general distribution of node embedding is achieved in GraphSAGE via normalisation as in Eq. (5).

- **GAT:** Graph Attention Network Convolution (GAT) employs attention mechanisms to determine how much focus each node in a graph should give to its neighbouring nodes. The attention mechanism allows nodes to selectively aggregate information from their neighbours, giving more weight to nodes that are more relevant to the current task. The details of the attention mechanism are as follows:

$$z_i^{(l)} = W^{(l)} h_i^{(l)}, \tag{6}$$

$$e_{ij}^{(l)} = LeakyReLU(\vec{U}^{(l)T}(z_i^{(l)}||z_j^{(l)})), \tag{7}$$

$$\alpha_{ij}^{(l)} = \frac{exp(e_{ij}^{(l)})}{\sum_{k \in N(i)} exp(e_{ik}^{(l)})}, \tag{8}$$

$$h_i^{(l+1)} = \sigma(\sum_{j \in N(i)} \alpha_{ij}^{(l)} z_j^{(l)}) \tag{9}$$

The linear transformation of lower layer embedding $h_i^{(l)}$ of a wallet or smart contract and transaction and their learnable weight matrix $W^{(l)}$ approached as described in Eq. (6). A pair-wise un-normalised attention score between a wallet or smart contract and its neighbour transactions computes via concatenates z embeddings of the wallet or smart contract and the transaction then takes a dot product of it and a learnable weight vector $\vec{U}^{(l)}$ and applies a *LeakyReLU* in the end as detailed in Eq. (7). A softmax operation applies to normalise the attention scores on each node's incoming edges as detailed in Eq. (8). Finally, the embeddings from the neighbours are aggregated together and scaled by the attention scores as in Eq. (9).

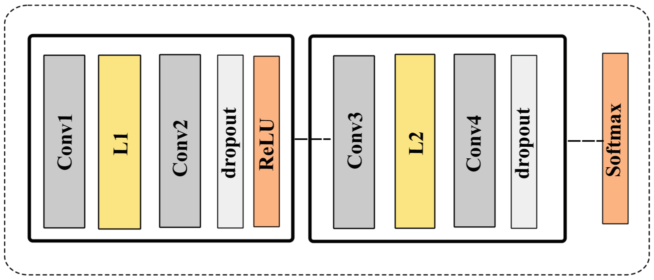

Fig. 4. Structure of the proposed GNN network for the analysis of blockchain network. Here, $Conv1$, $Conv2$, $Conv3$, and $Conv4$ represent the SAGE or/and GAT convolution layers, $L1$, and $L2$ are the liner layers, and dropout value $p = 0.2$. $ReLU$ and $Softmax$ are the activation functions.

The proposed GNN-based approach shown in Fig. 4 choose $L = 4$ and align Conv1 = GAT, Conv2 = SAGE, Conv3 = GAT, Conv4 = SAGE, liner layers $L1$ and $L2$, dropout value $p = 0.2$, learning rate 0.01, and the activation functions $ReLU$ and $Softmax$. The alignment of the SAGE convolution layer followed by the GAT convolution layer, first extracts the aggregate information from the target wallet's or smart contract's neighbour transactions, giving more weight to transactions that are more relevant to normal or malicious settlement behaviour. This weighted outcome provides an informative sample for the SAGE layer and improves the learning of the classification models.

Prediction Layer: The prediction layer utilises GNN-based node embedding outcomes to predict malicious wallets or smart contracts and transactions in Bitcoin and Ethereum networks. In this layer, we designed a cross entropy-based loss function, Adam optimiser for model optimisation, and applied gradient descent to improve classification. This helps the proposed GNN-based approach learn more task-based discriminative node embeddings for each wallet, smart contracts and transactions.

4 Evaluation

This section first details the experimental setup and the classification results obtained for hypergraph-based Graph Neural Network (GNN) analysis. Further, this section analyses and discusses the significance of the proposed approach by comparing the results reported in related works. The implementation of the proposed graph-based analysis was carried out on a computer with Ubuntu 22.04.2 LTS x86_64, 12th Gen Intel i9-12900 and 16085MiB/ 128511MiB, and PyTorch geometric with 3.10 kernel version.

The experimental setup of this study considered three different experimental setups using combinations of selected convolution networks. The first setup only considered the SAGE convolution network (Conv1 = Conv2 = Conv3 = Conv 4

= SAGE). Second, only considered GAT convolution network (Conv1 = Conv2 = Conv3 = Conv 4 = GAT) and the final one is the proposed approach explained in Sect. 3.4. The settings for the linear layers $L1$ and $L2$ remain consistent across all three setups, with a fixed dropout rate of $p = 0.2$, a learning rate of 0.01, and the activation functions of ReLU and Softmax being unchanged. The classification results were obtained for three different dimensions (64,128, and 256) of graph embedding vectors. The experiment of this study classified Bitcoin transactions and wallets, Ethereum smart contract transactions and addresses.

During the training, d dimension output from $Conv1$ is fed into a linear layer to obtain d dimensional linear output. The liner outputs are then fed into $Conv2$, which provides another d dimensional embedding vector. The output vectors produced by $Conv2$ are passed to the dropout layer with $p = 0.2$ to prevent overfitting during training. The output of the dropout layer is passed through a $ReLU$ activation function to ensure that negative neuron outputs are rectified to zero. The outcome of the $ReLU$ is transferred to $Conv3$ and produces a two-dimensional vector of values. The outcome is passed to the linear layer $L2$ to obtain linear output values. Then the linearly transformed outcomes are fed to the $Conv4$ layer and provide another two-dimensional vector of values. The outcome from the $Con4$ layer is fed to the dropout layer with $p = 0.2$ to prevent overfitting during training. Finally, the $Softmax$ layer processes the outcomes to ensure that the output values are between 0 and 1, representing the probability of each class (normal or malicious) for each type of node (wallet/smart contract/transaction). The output from the $Softmax$ is compared with the actual labels of wallet or smart contracts or transactions to identify loss. The loss value is fed back to the Adam optimiser to update the weights in each hidden layer for a new round of training. These iterations (500 epochs) increase the classification accuracy for the training set. Finally, a test set is used to obtain precision, recall, and F1 scores. These evaluation measures were selected to compare the experimental results with the results presented in literature studies.

Table 5 presents results for the classification of normal and malicious Bitcoin transactions. The 64 and 256 dimensions of the proposed GNN-based approach achieved high precision, recall, and F1-score, whereas for 128 dimensions SAGE convolution obtained the highest result. Based on the classification result for Ethereum smart contract transactions presented in Table 6 the proposed approach obtained a high precision, recall, and F1-score for 64 and 128 dimensions, whereas SAGE convolution obtained a high recall and F1-score for 256 dimensions.

The classification results for Bitcoin wallets are presented in Table 7 specifies that for 64 and 128 dimensions, the proposed approach obtained high precision, recall, and F1-score, whereas for 256 dimensions GraphSAGE obtained the best results. Results for the Ethereum smart contract address classification are detailed in Table 8, which reveals that for all dimensions of embedding vectors the proposed approach obtained a high recall and F1-score. For 128 and 256 dimensions, the GAT obtained the highest precision value.

Overall, the proposed GNN obtained a 0.8978 F1-score for Bitcoin transactions with 256 dimensions and a 0.8857 F1-score for Bitcoin wallets with 128 dimensions. For the Ethereum smart contracts, the proposed GNN obtained a 0.8481 F1-score with 256 dimensions for transactions and a 0.8399 F1-score with 64 dimensions for addresses. The learning time of the proposed GNN is comparatively higher than the SAGE-based GNN and less than the GAT-based GNN.

Table 5. Classification results for Bitcoin transactions.

GNN	Dimension of the embedding vector								
	d = 64			d = 128			d = 256		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SAGE	0.8796	0.8169	0.8471	0.9140	0.8716	**0.8923**	0.8061	0.7404	0.7719
GAT	0.8344	0.8534	0.8438	0.8842	0.8095	0.8452	0.8863	0.8265	0.8554
Proposed GNN	0.8988	0.8828	**0.8907**	0.9104	0.8538	0.8812	0.9223	0.8745	**0.8978**

Table 6. Classification results for smart contract transactions.

GNN	Dimension of the embedding vector								
	d = 64			d = 128			d = 256		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SAGE	0.6908	0.9501	0.8000	0.6815	0.9239	0.7844	0.6912	0.9534	0.8013
GAT	0.8667	0.6116	0.7172	0.8882	0.6558	0.7545	0.8501	0.8021	0.8254
Proposed GNN	0.8499	0.8430	**0.8464**	0.9615	0.6950	**0.8068**	0.9050	0.8414	**0.8720**

The classification results presented in Table 5 to 8 reveal the significance of the hypergraph-based GNN classification and the proposed GNN-based approach via high evaluation scores.

The comparison for binary class classification results of the Bitcoin transactions is provided in Table 5 and results presented in related studies are detailed in Table 9. There is no related literature for Bitcoin wallet classification, hence

Table 7. Classification results for Bitcoin wallets.

GNN	Dimension of the embedding vector								
	d = 64			d = 128			d = 256		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SAGE	0.8436	0.8926	0.8674	0.8508	0.8680	0.8594	0.8212	0.9094	**0.8631**
GAT	0.8410	0.8758	0.8581	0.8176	0.8221	0.8199	0.7956	0.8535	0.8235
Proposed GNN	0.8576	0.9027	**0.8796**	0.8746	0.8971	**0.8857**	0.8385	0.8826	0.8599

Table 8. Classification results for smart contract addresses.

GNN	Dimension of the embedding vector								
	d = 64			d = 128			d = 256		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SAGE	0.5846	0.8085	0.6786	0.5735	0.6783	0.8297	0.5946	0.9362	0.7273
GAT	0.7742	0.5106	0.6154	0.8387	0.5532	0.6667	0.7813	0.6329	0.5319
Proposed GNN	0.7924	0.8936	**0.8399**	0.7679	0.9149	**0.8349**	0.7288	0.9149	**0.8113**

no comparison details are provided. Similarly, the comparison of classification results of Ethereum smart contract addresses are provided in Table 8 and the results reported in the literature are detailed in Table 9.

Table 9. Comparison details for the classification results obtained via proposed GNN and the results in the related literature.

Blockchain	Approach	F1-score
Bitcoin	Artificial Neural Network	0.8854 [20]
	Deep Autoencoder	0.2081 [24]
	Graph Convolution Network	0.628 [30]
	Inspection-L	0.828 [21]
	Proposed GNN	**0.8978**
Ethereum	GCN	0.8963 [32]
	CNN	**0.959** [22]
	Code and Transaction Random Forest (CTRF)	0.909 [12]
	Proposed GNN	0.8399

Based on Table 9 the proposed GNN-based approach produced the most promising results for Bitcoin networks. Whereas, the results for the Ethereum smart contracts reveal the significance of the smart contract features related to code structure, byte contract length, and lifetime in the identification of suspicious behaviour [12] and [22].

The unique graph patterns for ransomware settlements and Ponzi contracts shown in Fig. 2 and Fig. 3 reveal that to distinguish the behaviour of the normal and malicious Bitcoin wallets we have to focus on up to four hops, whereas for the smart contract, it's only one. This could be the reason for the decrease in classification performance when $L > 4$. In terms of the dimension of the vector, the performance gets reduced when $d < 64$ or $d > 256$.

The main limitation of the above analysis is that the study was performed using transaction data stored in a local machine. The transaction node's properties used for smart contract analysis do not consider the raw data and code-based features.

Future work will investigate an improved GNN-based approach for Ethereum network analysis and integrate the proposed approach for a real-time and interactive monitoring tool to enhance the decision-making of end-users at blockchain-based systems, including critical infrastructures, by providing meaningful visualisation and early warnings.

5 Conclusion

This research work investigated the effectiveness of generalised heterogeneous graph modelling and proposed a GNN-based approach to predict malicious wallets and/ or smart contracts and their transactions in blockchain networks. The proposed hypergraph-based GNN analysis gave promising F1 scores for the prediction of malicious wallets, smart contracts, and transactions. The results obtained for Bitcoin network classification based on the proposed approach achieved marginal improvement compared to the results reported in related studies. The Ethereum smart contract-based classification results indicate the need for including the code and lifetime-based features of smart contracts in suspicious behaviour identification. The proposed generalised GNN-based approach may integrate with the real-time blockchain network to monitor and analyse malicious behaviour. Such integration is beneficial in terms of prompt alerts or early warnings for forensic analysers, law enforcement authorities, and financial regulators to maintain a secure and trusted blockchain ecosystem, which is essential for the adoption and success of blockchain technology across various industries.

References

1. Blockchain data API (2021). https://www.blockchain.com/api/blockchain_api
2. Ponzi smart contract (2021). https://www.kaggle.com/datasets/xblock/smart-ponzi-scheme-labels
3. Ethereum transaction dataset (2022). https://api.blockcypher.com/v1/eth/main/txs/
4. Jeyakumar, S.T., Ko, R., Muthukkumarasamy, V.: A framework for user-centric visualisation of blockchain transactions in critical infrastructure. In: Proceedings of the 5th ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI 2023), pp. 44–52. Association for Computing Machinery, New York, USA (2023). https://doi.org/10.1145/3594556.3594624
5. Akcora, C.G., Li, Y., Gel, Y.R., Kantarcioglu, M.: Bitcoinheist: topological data analysis for ransomware detection on the bitcoin blockchain. arXiv preprint [Web Link] (2019)
6. Akoglu, L., Tong, H., Koutra, D.: Graph-based anomaly detection and description: a survey (2014)
7. Brambilla, M., Javadian Sabet, A., Kharmale, K., Sulistiawati, A.E.: Graph-based conversation analysis in social media. Big Data Cogn. Comput. **6**(4) (2022). https://doi.org/10.3390/bdcc6040113, https://www.mdpi.com/2504-2289/6/4/113

8. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
9. Di Francesco Maesa, D., Mori, P.: Blockchain 3.0 applications survey (2020)
10. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
11. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs (2018)
12. He, X., Yang, T., Chen, L.: CTRF: ethereum-based ponzi contract identification. Secur. Commun. Netw. **2022** (2022)
13. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
14. Jeyakumar, S., Eugene Yugarajah, A.C., Rathore, P., Palaniswami, M., Muthukkumarasamy, V., Hóu, Z.: Feature engineering for anomaly detection and classification of blockchain transactions, March 2023. https://doi.org/10.36227/techrxiv.22329805.v1, https://www.techrxiv.org/articles/preprint/Feature_Engineering_for_Anomaly_Detection_and_Classification_of_Blockchain_Transactions/22329805
15. Jeyakumar, S., Hóu, Z., Eugene Yugarajah, A.C., Palaniswami, M., Muthukkumarasamy, V.: Visualizing blockchain transaction behavioural pattern: a graph-based approach, March 2023. https://doi.org/10.36227/techrxiv.22329889.v1, https://www.techrxiv.org/articles/preprint/Visualizing_Blockchain_Transaction_Behavioural_Pattern_A_Graph-based_Approach/22329889
16. Kılıç, B., Özturan, C., Sen, A.: Analyzing large-scale blockchain transaction graphs for fraudulent activities. Big Data Artif. Intell. Digit. Finan. **253** (2022)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)
18. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017)
19. Kriegel, H.P., Kröger, P., Schubert, E., Zimek, A.: Loop: local outlier probabilities. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1649–1652 (2009)
20. Lee, C., Maharjan, S., Ko, K., Hong, J.W.-K.: Toward detecting illegal transactions on bitcoin using machine-learning methods. In: Zheng, Z., Dai, H.-N., Tang, M., Chen, X. (eds.) BlockSys 2019. CCIS, vol. 1156, pp. 520–533. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-2777-7_42
21. Lo, W.W., Kulatilleke, G.K., Sarhan, M., Layeghy, S., Portmann, M.: Inspection-l: self-supervised GNN node embeddings for money laundering detection in bitcoin. Appl. Intell. **53**, 1–12 (2023). https://doi.org/10.1007/s10489-023-04504-9
22. Lou, Y., Zhang, Y., Chen, S.: Ponzi contracts detection based on improved convolutional neural network. In: 2020 IEEE International Conference on Services Computing (SCC), pp. 353–360 (2020). https://doi.org/10.1109/SCC49832.2020.00053
23. Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to Linear Regression Analysis. Wiley, Hoboken (2021)
24. Nan, L., Tao, D.: Bitcoin mixing detection using deep autoencoder. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), pp. 280–287. IEEE (2018)
25. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
26. Samantha Tharani, J., R.K., Muthukkumarasamy, V.: A framework for user-centric visualisation of blockchain transactions in critical infrastructure (2023)

27. Shojaeenasab, A., Motamed, A.P., Bahrak, B.: Mixing detection on bitcoin transactions using statistical patterns. arXiv preprint arXiv:2204.02019 (2022)
28. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077 (2015)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018)
30. Weber, M., et al.: Anti-money laundering in bitcoin: experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591 (2019)
31. Wu, J., Liu, J., Chen, W., Huang, H., Zheng, Z., Zhang, Y.: Detecting mixing services via mining bitcoin transaction network with hybrid motifs. IEEE Trans. Syst. Man Cybern. Syst. **52**(4), 2237–2249 (2021)
32. Yu, S., Jin, J., Xie, Y., Shen, J., Xuan, Q.: Ponzi scheme detection in ethereum transaction network. In: Dai, H.-N., Liu, X., Luo, D.X., Xiao, J., Chen, X. (eds.) BlockSys 2021. CCIS, vol. 1490, pp. 175–186. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-7993-3_14