2023

Graph neural networks for ethereum fraud detection

Charity Mwanza

GRAPH NEURAL NETWORKS FOR ETHEREUM FRAUD DETECTION

Charity Mwanza

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master's of Science in Business Analytics
in
The Department of Information Systems, Supply Chain, and Analytics
to
The Graduate School
of
The University of Alabama in Huntsville
May 2023

Approved by:

Dr. Hieu Pham, Research Advisor
Dr. Yi Tan, Committee Member
Dr. Qingyun Zhu, Committee Member
Dr. Wai Mok, Department Chair
Dr. Jason Greene, College Dean
Dr. Jon Hakkila, Graduate Dean

Abstract

GRAPH NEURAL NETWORKS FOR ETHEREUM FRAUD DETECTION

Charity Mwanza

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Business Analytics

Information Systems, Supply Chain, and Analytics

The University of Alabama in Huntsville
May 2023

Detecting fraudulent transactions on the Ethereum network can help cryptocurrency companies that operate on the Ethereum platform protect their users from exposure to fraudsters. The most common fraudulent activities in the cryptocurrency network include phishing and smart Ponzi schemes. Since cryptocurrency technology is still young, most investors lack knowledge of how the smart contacts used in the Ethereum platform operate; hence, they cannot evaluate the risks they are exposed to when carrying out cryptocurrency transactions. The key role that this paper looks at is the application of graph neural networks in the extraction of features of users in the Ethereum platform and their respective transactions to classify them as either fraudulent or not fraudulent. The classification results produced by this research show that application of graph neural networks can be used to detect fraudulent transactions and help managers of the Ethereum platform take the necessary actions toward curbing fraudulent activities.

Acknowledgements

With sincere gratitude, I would like to acknowledge my supervisor Dr. Hieu Pham, who made this work possible. His guidance and advice carried through all the stages of this project. I would also like to thank the UAH College of Business staff for their immense support, especially the committee members, for making the defense process an enjoyable one. I would also like to extend my sincere gratitude to my family members for their continuous support. Finally, I am grateful to God for his guidance day by day and for letting me through all the difficulties. It's God who let me finish my degree, and I will keep on trusting Him for my future.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

As the cryptocurrency market continues to gain popularity over the years, security concerns in this new blockchain technology have become prevalent, calling for research to be carried out to counter these drawbacks. By the end of 2021, cryptocurrency crime reports hit a new high mark, with illicit addresses receiving approximately 14 billion USD in 2021, an increase from the $7.8 billion mark in 2020 (Chainanalysis, 2022). This crime report is directly proportional to the total growth of total transactions in the cryptocurrency market. In the 2021 financial year, the total transaction volume in the crypto market grew by 567% from 2020, reaching 15.8 trillion USD. This implies that the total illicit activities in the blockchain network were approximately 0.15% of the total volume of transactions, which is a small proportion but still very significant.

Many investors in the blockchain industry do not have a technical understanding of how a cryptocurrency network works; hence, they invest due to the temptations of the appreciating market, exposing themselves to financial crime activities which lead to significant economic losses. The anonymous nature of users in the crypto market makes it hard to track the addresses that carry out fraudulent activities in a blockchain network. Traditionally, crime detection was carried out by tracking the source code of the transactions, but in blockchain technology, these source codes are hidden, making it even harder for these fraudulent users to be identified (Liu *et al.*, 2022). This paper uses deep learning Graph Neural Network algorithms to study the features of the nodes and edges

of these fraudulent activities in order to classify the likelihood of a transaction being fraudulent in the Ethereum blockchain platform.

1.1 Background

Ethereum is an open-source blockchain platform that provides a globally decentralized infrastructure for executing smart contract programs using virtual machines. This platform was created primarily to solve the problems caused by the rigid nature of the preceding cryptocurrency – the bitcoin platform, which is the world's first and largest cryptocurrency network (Kushwaha *et al.*, 2022). Bitcoin uses limited scripting language that forces developers to build on the existing bitcoin infrastructure or build their blockchain from scratch. Moreover, the bitcoin platform also provides limited transaction types, storage capabilities, and data types. Like bitcoin and other cryptocurrency platforms, the Ethereum platform provides a peer-to-peer transaction network and uses cryptographic functionalities such as digital signatures, hashes, and digital currency. In addition to these similarities, the Ethereum blockchain provides higher transparency, neutrality, and auditability and reduces censorship, which has helped increase its popularity in the blockchain industry. Although Ethereum offers the above advantages over other blockchains, the Ethereum platform was not created primarily as a digital payment network but as a platform for users to extract smart contracts. The Ether currency used in the platform serves as a utility currency that users use to pay for the Ethereum platform resources.

Although banks and governments have invested much in measures to detect, control, manage and prevent financial fraud, people still manage to find ways of

circumventing the system and carrying out fraudulent activities. Most fraudulent activities in the financial industry include bribery, money laundering, and other crime-related transactions such as drug and substance business, human trafficking activities, and mismanagement of public funds.

The introduction of blockchain technology provides a bigger platform for fraudulent financial activities, with fraudsters creating noble schemes of carrying out these activities at the expense of innocent investors. Unlike traditional banking, where transactions are regulated by third bodies such as banks and governments and where laws can easily be enforced, blockchain technology has no third parties to protect investors from fraudsters (Liu et al., 2022). Fraudsters take advantage of the anonymity of the blockchain network to carry out their illicit activities. The Ethereum network's current internal security system can solve fraudulent activities such as double spending but has no controls to identify, detect or flag suspicious transactions.

Like traditional contracts, smart contracts create laws between users and enforce them. Smart contracts use computer codes deployed on the Ethereum network to govern and control operations between parties in the Ethereum network and execute the contract when the terms of those contracts are met (Bistarelli *et al.*, 2022). Some of the advantages of smart contracts include instant execution of transactions, public record keeping of transactions, and transparency of the system by ensuring the contract terms are visible to all parties. The codes of these smart contracts are publicly available, and users can test their authenticity before using them. Each smart contract uses RIPEMD-160, a unique contact address 160-bit hash calculated by the cryptography hash function, which makes

it extremely difficult for fake contract attackers to attack the Ethereum network successfully.

Smart Ponzi schemes, the most common fraudulent activity in the Ethereum network, use social engineering to manipulate users into joining a Ponzi scheme with high rates of return and few risks. Ponzi schemes benefit early investors who acquire more investors into the scheme. The new investors' capital is utilized to pay early investors until there is insufficient cash to pay out old investors (Bartoletti *et al.*, 2020). Investors develop a false sense of trust blinded by high return expectations.

Phishing is a process used by fraudsters to trick cryptocurrency users into revealing their personal information that can help fraudsters access their wallets, such as private keys and passwords. Like many standard scams, fraudsters send emails to users impersonating Ethereum exchange support and security, and thereby lure the users into giving out their data. Fraudsters are usually interested in users' crypto wallet private keys, which can give them access to the user's account, where they can withdraw or transfer money to their accounts. According to FBI data, phishing scams in 2021 affected more than 323,000 people, and total transactions from these scams amounted to 44.2 million USD (FBI Report, 2022).

1.2 Literature Review

Yuan *et al.* (2020), in their attempt to detect phishing scams in the Ethereum network, use a three-step process of extracting account features and classifying whether these accounts are phishing accounts or not. In their research, labeled phishing accounts with their corresponding transaction records are used to study standard features of phishing scam activities. The network embedding method, *node2vec,* is used to extract

features from these accounts and the records of their transactions. One support vector machine algorithm is applied to the features extracted to classify whether the accounts are phishing accounts. These authors compare the results of the *node2vec* embedding method and those of the support vector machine algorithm, where they get an *F-1* score of 0.846. Although this research provides excellent accuracy for detecting phishing scams in the Ethereum network, the methodology cannot detect other fraudulent activities in the network, such as Ponzi schemes. A different machine learning methodology is needed to detect all fraudulent activities in the Ethereum network rather than detecting one fraudulent activity.

In another research study towards classifying phishing accounts in the Ethereum network by Chen *et al.* (2020), a graph-based cascade method is applied to extract features for the accounts dataset. This research proposed a dual-sampling ensemble algorithm for the accounts classification process. In their research, the dataset used had a small proportion of phishing accounts, making it hard to accurately extract features that can be generalized for all phishing accounts. Moreover, their research also faces a major drawback of lack of adequate information on all phishing accounts and the inability of the research to detect other fraudulent activities apart from phishing.

Ashfaq *et al.* (2022) used XGBoost and random forest machine learning algorithms to classify transactions in the Ethereum network as either fraudulent or not fraudulent. In their research, these machine learning algorithms study transaction patterns and classify their likelihood of being fraudulent. The research also calculates precision and AUC metrics of 0.6 and 0.89, respectively. The major challenge with this research is that it classifies fraudulent transactions but not the specific accounts carrying

the fraudulent activities. In every fraudulent transaction, the two parties involved are a fraudster and the victim, which this research fails to acknowledge.

Chapter 2. Methodology

2.1 Graph Neural Networks (GNNs)

A graph neural network is a type of deep learning neural network that operates on a graph structure, to extract information from a dataset for decision-making. According to Zhou *et al.* (2020), graph neural networks have increased in popularity over the years in research domains such as social networks, chip placement, drug discovery, forecasting, and bioinformatics. GNNs are mostly applied in node classification and edge classification problems. A graph in computer science refers to a data structure composed of nodes *(V)* and edges *(E)*.

$$G = (V, E)$$

The nodes are a set of vertices connected by the edges. Edges in a graph can either be directed or undirected.

In the node classification problem, each node *V* is characterized by the information of its neighboring nodes connected to it. GNNs carry out node classification by studying the information of the nodes connected to a particular node and using that information to label other unlabeled nodes. The node features in a GNN are stored in a feature matrix that is formed by stacking together the dimensional features from the nodes, forming a *Vxd* matrix. The h_v, defined as $h_v = f(x_v, x_{co}(v), h_{ne}(v_e), X_{ne}[v])$, represents the feature vector of a node collected from its neighbors, where

- $x_{co}(v)$ represents the features of the edges connecting with *V*,
- h_{ne} represents the embedding of the neighboring nodes,
- X_{ne} represents the features of the neighboring nodes, and
- *f* represents a transition function that projects these inputs into a dimensional space.

Edges, on the other hand, are represented using an adjacency matrix. If two nodes *i* and *j* are connected, $e_{ij} = 1$ and $e_{ij} = 0$, these nodes can be represented in an adjacency matrix. This is observed in the directional graph in Figure 2.1 and Table 2.1.

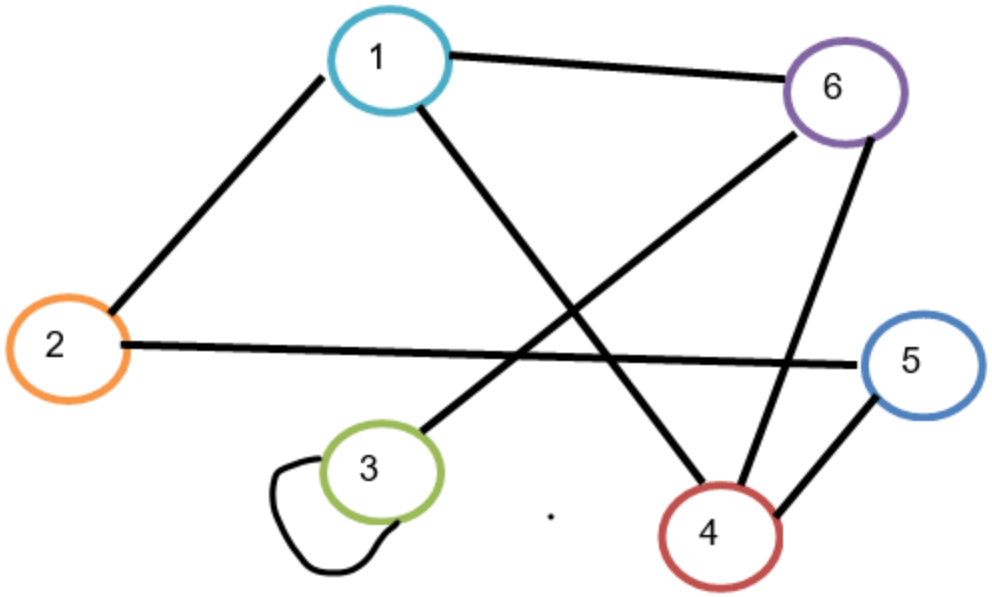


Figure 2.1 A sample undirected network with six nodes and eight edges. The circles are the nodes, while the lines are the edges. The curved line at the node represents a connection between node three and itself.

Table 2.1 A representation of the above sample network in an adjacency matrix.

Node	**1**	**2**	**3**	**4**	**5**	**6**
1	0	1	0	1	0	1
2	1	0	0	0	1	0
3	0	0	1	0	0	1

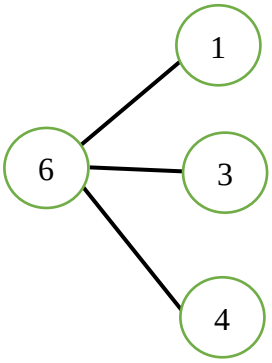
4	1	0	0	0	1	1
5	0	1	0	1	0	0
6	1	0	1	1	0	0

2.2 Message Passing

GNNs are famous for their ability to study structured information. They assume that nodes with similar characteristics are connected to each other; hence, a node's features are derived from the aggregate of the features of the neighboring nodes. Therefore, the neighborhood of a node N_i refers to the set of all the other nodes j connected to i by an edge.

$$N_i = \{j : e_{ij} \sum E\}$$

Message passing refers to taking the node features in a neighborhood and passing them to the source node. GNNs repeat this process in a parallel manner for all the nodes in a graph, examining all the neighborhoods and updating the source nodes with results collected from its neighbors.



In this example;
Node $\boldsymbol{N_6 = \{N_1, N_3, N_4\}}$

The node features 1, 3, and 4 are transformed into functions; $\boldsymbol{f(x_1), f(x_3), f(x_4)}$

The function f can be either a simple Multilayer perceptron (MLP), a Recurrent Neural Network (RNN) or affine transformation function where $\boldsymbol{f(x_j) = W_j x_j + b}$

Figure 2.2 An illustration of message passing in Graph Neural Networks

2.3 Aggregation

After collecting information from the neighbors of a source node, the message passed from all the neighboring nodes have to be combined in a particular manner. Popular aggregation functions in GNNs include sum, mean, maximum, and minimum. The choice of the aggregation method to be used depends on the problem to be solved.

$$Sum = \sum_{j \in N_i} W_j x_j$$

$$Mean = \frac{\sum j \epsilon N_i \; W j x j}{|N_i|}$$

$$Max = max(\{W_j x_j\}) \; for \; j \in N_i$$

$$Min = (\{W_j x_j\}) \, for \, j \in N_i$$

The final aggregation function is represented as

$$m_i = G(\{W_j x_j\}) \, for \, j \in N_i\}$$

where *G* can either be sum, mean, maximum, or minimum.

2.3 Update

Under this stage, the source node is updated with an aggregate of messages collected from its neighbors. By the end of the update stage, the algorithm knows much about the node and its neighbors. The update method is carried out by combining the node features with the aggregate messages. The combination procedure can be carried out either through addition or concatenation.

Using addition:

$$h_i = \sigma(K\big(H(x_i)\big) + m_i)))$$

Using concatenation:

$$h_i = \sigma(K(H(x_i)) \oplus m_i)))$$

where σ is an activation function such as Tanh, RELU, or ELU, *H* is a simple neural network such as MLP or affine transform, and *K* represents another MLP to project the added vectors to another dimension. After collecting the messages, aggregating them together, and updating the node, a single GNN layer on a single node say *i* can be summarized as

$$h_i = \sigma(W_1 h_i + \sum_{j \epsilon N_i} W_2 h_j\ ).$$

Chapter 3. Experiment and Results

This section sheds more light on the data preparation process, the data sources, the model comparison, and the classification of the experimental results. The first section elaborates more on the data sources and configuration, and the subsequent sections explicate the comparative model approach applied in the experimental process. At the final stage of the topic, the classification of the node embedding from the neural network and analysis of the classification results are explained in detail.

3.1 Dataset

The data was acquired from Al-E'mari *et al.* (2021), and it contains 71,251 transactions. Out of these transactions, 2,558 are labelled as fraudulent transactions. The proposed dataset used in this example contains data from Ethereum users from 2017 to 2019. The data is distributed through the three years, as shown in Figure 3.1.

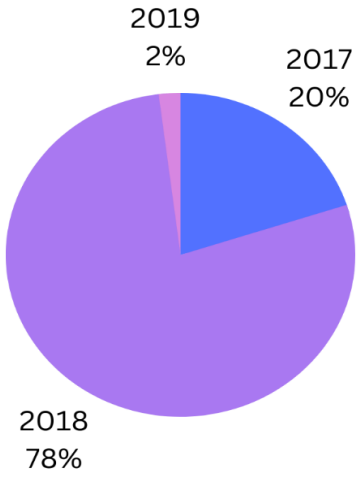


Figure 3.1 A pie chart representation of observations for the years 2017, 2018, and 2019.

Over the three years this dataset was obtained, the rate of fraudulent transactions increased from one year to another. According to the graphs below, fraudulent transaction began in 2017, and they have been increasing ever since.

This dataset contains 15 columns that give more information about the node (the users of the Ethereum platform) and the edges (transaction information). These columns are described in Table 3.1 below, and an illustration of a graph network is shown in Figure 3.2.

Table 3.1 Columns of the dataset and their definitions.

Column	Description
Hash	Hash of the transaction, a figure generated by the Ethereum platform
Nonce	Total amount of transactions executed by the senders
Transaction_index	Transaction index for a block
From_address	Sender's wallet address
To_address	Recipient’s wallet address
Value	Value of the transaction amount in Wei
Gas	Total gas used by the sender

Gas_price	Gas price in Wei
Input	The data transmitted by the sender in this transaction
Receipt_cumulative_gas_used	Total gas used in a transaction when executed as a block
Receipt_gas_used	Total gas used for a transaction
Block_timestamp	Timestamp for the block used during a transaction
Block_number	Transaction's block number
Block_hash	Transaction block's hash
From_scam	The value one denotes if the sender is fraudulent, while zero denotes if the sender is not fraudulent
To_scam	The value one denotes if the recipient is fraudulent, while zero denotes if the recipient is not fraudulent
From_category	Category of activity done by the sender can be phishing, Ponzi scheme, or scamming. Null represents a normal transaction.
To_category	Category of activity done by the recipient can be phishing, Ponzi scheme, or scamming. Null represents a normal transaction.

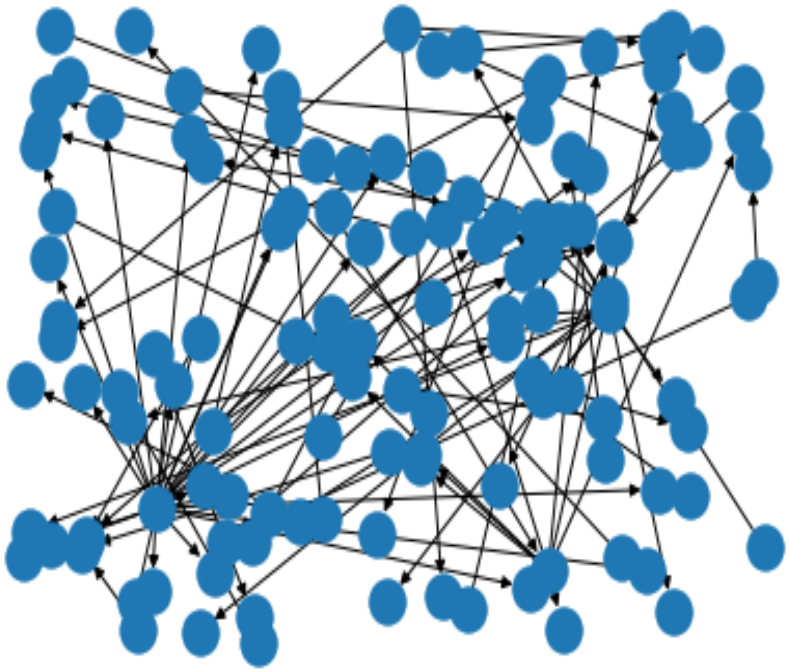


Figure 3.2 A visual representation of the first 100 observations network in the dataset.

3.2 Data Preprocessing

After loading the dataset into Google's Colab, data scrubbing was done by checking the accuracy, consistency, missing values, and repetition in the dataset. This data had no missing values, no repetition, and all the rows were accurate and consistent. This made it easier for the graph neural network algorithm to study the dataset. To improve the algorithm's performance in studying the node features, feature engineering techniques such as adding new columns and feature selection methods were applied. The unimportant columns in this dataset were removed, and the remaining columns were separated into node feature columns and edge feature columns as displayed in Table 3.2.

Table 3.2 A table of columns that represent the nodes, edges, and the response variables.

Edge Features	Node Features	Response
• Gas • Gas_price • Receipt_cumulative_gas_used • Receipt_gas_used • Block_number • Value_cat • Number of years in operations	• From_address • To_address • Year of registration • Maximum transaction value	• From_scam • To_scam

Directed graph neural networks are used to study this data and extract information from the nodes and the edges to update the nodes for classification purposes. A full batch generator with a self-loop method provided the best results for our analysis. Our neural network architecture included seven layers with 4, 32, 16, 16, 16, and 4 nodes respectively, and an output layer of 2 nodes. We provide an illustration of a graph neural network in Figure 3.3. In the analysis process, Python's StellarGraph proved to be the

best match for our dataset needs, and it provided a clear and concise way of extracting information from the network of transactions in the dataset. Out of the 71,251 observations, half of the data was used as a training set, 0.25 as the validation set, and the rest in the test set. The training, validation, and test dataset split was randomly selected from all the observations. After tuning the model parameters, Adam optimizers with the self-loops method for 20 epoch iteration training provided the best results for this process which are discussed below.

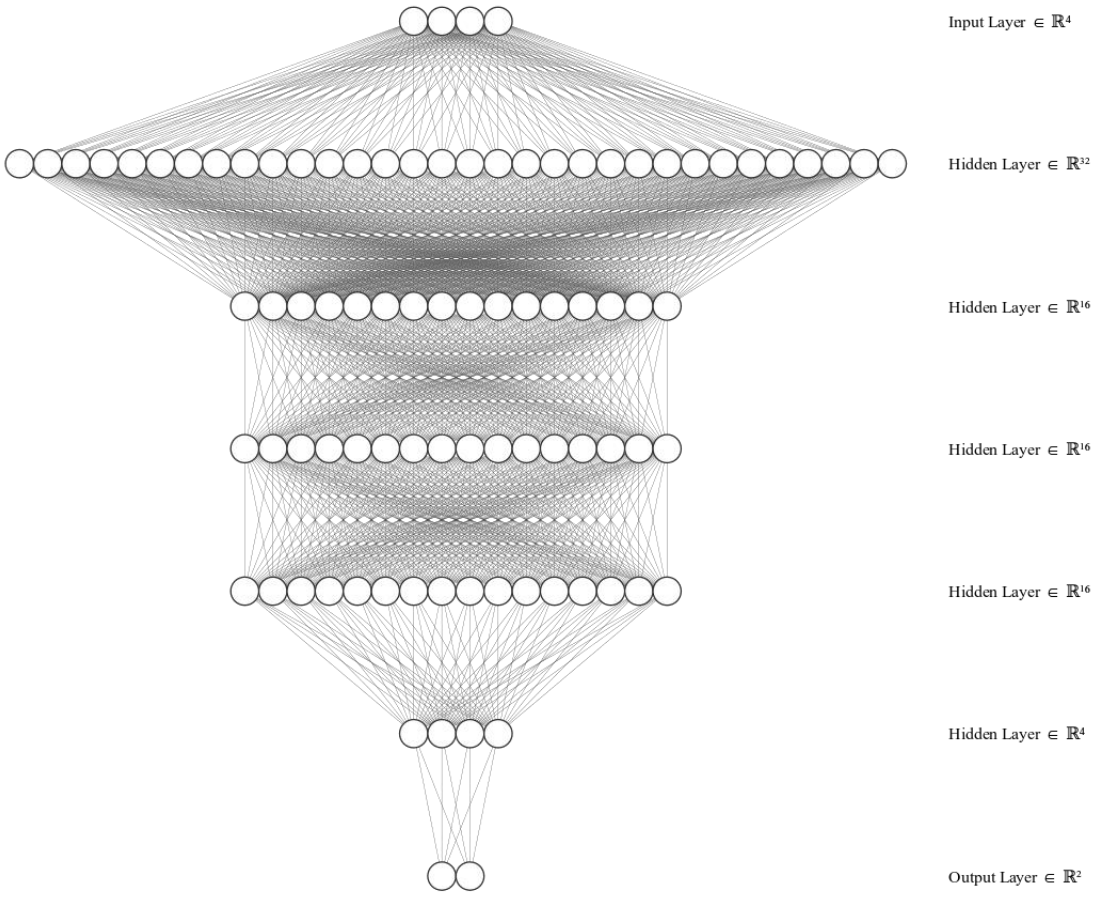


Figure 3.3 A graph neural network with input layers of 4, 5 hidden layers of 32, 16, 16, 16, and 4 respectively, and an output layer of 2.

3.3 Results

To verify the effectiveness of our model, we compared our results with those of traditional non-graph-based models *i.e.*, Random forest, Decision tree, and *K* Nearest neighbor classifier as the baselines. These traditional non-graph-based models have been applied to different datasets over the years and have provided promising results. Random forest algorithm, also known as random decision forest, is an ensemble learning method that constructs numerous trees of the various subsets of the entire dataset and takes the average of the decision trees created for classification. On the other hand, the decision tree classification algorithm uses multiple mathematical computations to split nodes into sub-nodes using defined criteria, creating more decision trees that are later used for the classification of observations in the dataset. Lastly, K nearest neighbors is a supervised, machine learning model that uses the proximity of observations in a dataset to make classifications.

To verify the effectiveness of the graph neural network model in detecting fraudulent activities in the Ethereum network, we compared the graph-based algorithms results with those of the traditional non-graph-based algorithm results.

Recall is the performance metric that was used for the evaluation of the test results. In this problem, it is essential to measure what proportion of actual fraudulent activities was identified correctly, hence the use of recall defined as the sum of true positives divided by the cumulative sum of true positives and false negatives. The formula for recall is acquired from the below confusion matrix below in Table 3.3.

Table 3.3 An illustration of the confusion matrix where performance metrics are calculated.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Comparing the results from the different classification models tested, graph neural networks had the best recall of 0.69 in the test set before parameter tuning. The results are displayed in Table 3.4.

Table 3.4 Comparison of the performance of different models

Algorithm	**Validation set recall**	**Test set recall**
Random Forest	0.43	0.47
Decision Tree	0.65	0.39
K-Nearest Neighbors	0.28	0.34
Graph Neural Networks	0.70	0.69

The parameter tuning results of the graph neural network increased the test set recall to 0.8237 with a loss of 0.06. Figure 3.4 and Table 3.5 summarizes these findings.

Table 3.5 Parameter tuning results of Graph Neural Networks

Parameter	Minimum Loss	Maximum Recall
Optimizers using GNC method		
Adam optimizer	123465	0.78
SGD Optimizer	11234512	0.25
RMSprop	95423	0.21
Optimizers using SGC method		
Adam optimizer	15263	0.35
SGD Optimizer	2530	0.13
RMSprop	8965	0
Optimizers using self-loop method		
Adam optimizer	0.6612	0.8237
SGD Optimizer	15.38	0.36
RMSprop	126	0.17

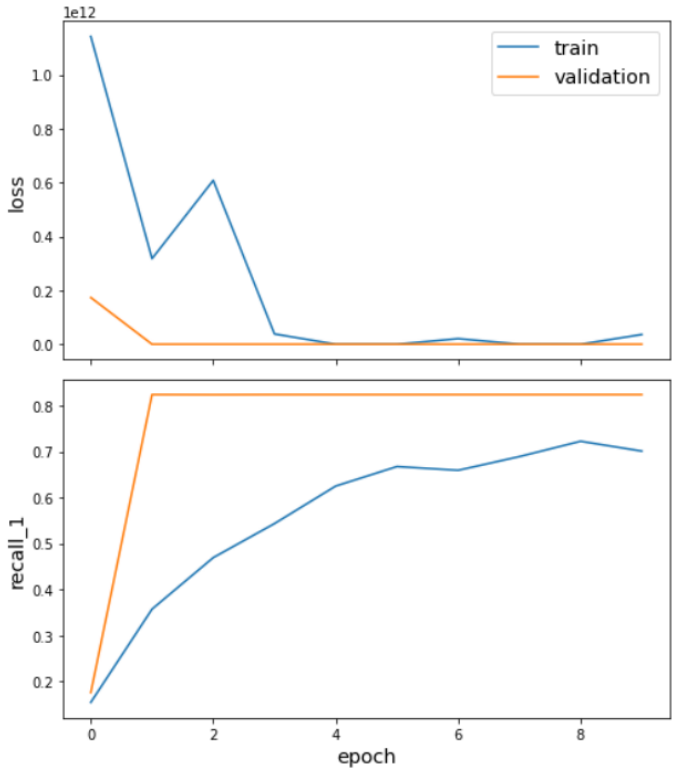


Figure 3.4 A graphical visualization of recall and loss value change within the 10 epochs ran in this model.

From the above results, it is clear that graph neural networks provide better results than other models since they gather a significant amount of information from the nodes, which in this case represent the specific users of the Ethereum platform, and the edges, which represent the specific features of a particular transaction. This research shows the best results when the Adam optimizer is used with the self-loop method.

Chapter 4. Conclusions

From this work, we notice that indeed, graph neural networks are able to accurately classify fraudulent users based on their transaction history on the Ethereum blockchain. Our results demonstrate that GNNs are superior to traditional machine learning methods such as random forests and support vector machines. One key reason is the inherent ability of the GNNs to make use of the natural network structure of our transaction dataset as opposed to the tabular form that other algorithms use. By learning directly from the network structure, more information can be inferred and not lost due to data translations.

From a practical perspective, being able to identify fraudulent transactions before they occur is of vital importance to the cryptocurrency community. Once a transaction is complete, it is near impossible to reacquire the funds sent in error. Therefore, centralized exchanges such as Binance can hold funds in escrows for a few minutes to algorithmically verify the validity of a transaction. Once the transaction is deemed to be legitimate, these centralized exchanges can send the funds to the blockchain.

As fraudsters advance their capabilities of stealing from innocent investors, families, communities, and economies count more losses as time goes by. This calls for immediate and urgent measures to combat fraudsters; hence, identifying fraudulent activities accurately and promptly is essential. From this paper, it is clear that through graph neural networks, cryptocurrency companies can use the representation of their

platform users in graphs to gather more information about them and identify their intentions of transacting in their platforms. This can predict whether certain users are likely to carry out fraudulent activities and help the management of these platforms to create measures to prevent fraud.

To build upon this work, the next steps are to combine our GNN with an explainable artificial intelligence approach such as GraphLIME to further understand the behavior of users and transactions (Huang *et al.* 2022). Once that is complete, one should explore custom architectures that maximize accuracy as well as computational time. If this is to be implemented for wide adoption, then designing an algorithm for real-time detection is a necessity.

References

Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alsafari, S., & Hameed, I. A. (2022). A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism. *Sensors*, *22*(19), 7162.

Bartoletti, M., Carta, S., Cimoli, T., & Saia, R. (2020). Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, *102*, 259-277.

Bistarelli, S., Mazzante, G., Micheletti, M., Mostarda, L., Sestili, D., & Tiezzi, F. (2020). Ethereum smart contracts: Analysis and statistics of their source code and opcodes. *Internet of Things*, *11*, 100198.

Chainalysis(2022) Crypto Crime Trends for 2022: Illicit Transaction Activity reaches All-Time High in value, All-Time Low in Share of All Cryptocurrency Activity. Retrieved from URL https://blog.chainalysis.com/reports/2022-crypto-crime-report-introduction/

Chen, W., Guo, X., Chen, Z., Zheng, Z., & Lu, Y. (2020). Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem. *IJCAI*, *7*, 4456-4462.

FBI Report, (2022). Spoofing and Phishing. Retrieved from URL https://www.fbi.gov/contact-us/field-offices/boston/news/press-releases/fbi-warns-public-to-beware-of-tech-support-scammers-targeting-financial-accounts-using-remote-desktop-software

Huang, Q., Yamada, M., Tian, Y., Singh, D., & Chang, Y. (2022). Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*.

Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., & Lee, H. N. (2022). Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access*, *10*, 6605-6621.

Liu, L., Tsai, W. T., Bhuiyan, M. Z. A., Peng, H., & Liu, M. (2022). Blockchain-enabled fraud discovery through abnormal smart contract detection on Ethereum. *Future Generation Computer Systems*, *128*, 158-166.

Al-E'mari, S., Anbar, M., Sanjalawe, Y., & Manickam, S. (2021). A labeled transactions-based dataset on the ethereum network. In Advances in Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December 8-9, 2020, Revised Selected Papers 2 (pp. 61-79). Springer Singapore.

Yuan, Q., Huang, B., Zhang, J., Wu, J., Zhang, H., & Zhang, X. (2020, October). Detecting phishing scams on ethereum based on transaction records. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). IEEE.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57-81.