Heterogeneous Network Representation Learning Approach for Ethereum Identity Identification

Yixian Wang, Zhaowei Liu, Jindong Xu, and Weiqing Yan

***Abstract*— Recently, network representation learning has been widely used to mine and analyze network characteristics, and it is also applied to blockchain, but most of the embedding methods in blockchain ignore the heterogeneity of network, so it is difficult to accurately describe the characteristics of the transaction. As smart society evolves, Ethereum makes smart contracts reality, while the mine of transaction characteristics appearing on the Ethereum platform is scarce; thus, there is an urgent need to mine Ethereum from contract and transfer. In this article, we propose a heterogeneous network representation learning method to mine implicit information inside Ethereum transactions. Specifically, we construct an Ethereum transaction network by collecting transaction data from normal and phishing Ethereum accounts. Then, we propose a walk strategy that combines timestamps and transaction amounts to represent the information that occurs at the time of a transaction. To mine the types of nodes and edges, we use a heterogeneous network representation learning method to map the transaction network to a low-dimensional space. Finally, we improve the accuracy of the embedding results in the node classification task, which has important implications for Ethereum mining as well as identity recognition.**

***Index Terms*— Ethereum, heterogeneous network representation learning, node classification, transactions network.**

I. Introduction

BLOCKCHAIN [1] is an open distributed database and is maintained through the consensus mechanism. Due to its openness, decentralization, and anonymity, blockchain technology is spreading rapidly in academia and industry. With the progress of smart society, cryptocurrencies [2], unlike traditional currencies commonly issued by authorized financial institutions, are managed by consensus among users in the network, thus giving an opportunity for unscrupulous elements to take advantage of it. Meanwhile, the occurrence of cybercrime incidents related to cryptocurrencies increases the price volatility of target cryptocurrencies and the correlation of a wide range of cross-cryptocurrencies, and it led to a negative impact on the price of cryptocurrencies in society.

Bitcoin [3] is a typical application of cryptocurrency and a public transaction ledger applying blockchain technology [4], and it has gained much attention since 2009. However, Bitcoin does not support smart contracts [5], and Ethereum turns "smart contracts" from theory to reality and created the blockchain 2.0 phase. With the development of smart society, Ethereum [6] designed to be computationally generic and Turing-complete, and it has recently attracted a lot of attention. To facilitate the implementation of smart contracts, Ethereum introduced the concept of account, which exists formally through an address. Currently, research on Ethereum focuses on security and performance issues based on blockchain technology, and there is an urgent need for research on the interaction between account and smart contracts. However, as the use of Ethereum becomes larger, various cybercrimes occurring on Ethereum appeared [7].

In fact, illegal practices are frequently occurring on the Ethereum transaction platform, and over 10% of Ethereum accounts have reportedly been subjected to a variety of scams, including phishing [8], money laundering [9], Ponzi schemes [10], and other scams [11].

According to a report by Chainalysis [12], crime related to cryptocurrencies declines significantly in 2020. While cryptocurrencies are designed to be transparent and traceable, allowing users to send funds instantly from anywhere, it still gives criminals an opportunity to take advantage of them due to anonymity of it. Although the share of illegal transactions in all cryptocurrencies decreased to 0.34% in 2020, the total amount of illegal transactions still reaches 10 billion. This is a significant decrease from 2019 when the total amount of illegal cryptocurrency transactions was 20 billion and illegal activity accounted for approximately 2.1% in all cryptocurrency transactions, but still thousands of individual users lost a total of up to 7.3 million.

Moreover, among the various security issues of blockchain cryptocurrencies, although the total number of cryptocurrencies received by crimes represented by darknet markets [13] and ransomware [14] has gradually increased, the number of phishing, Ponzi scheme scams had still accounted for more than 50% of all cybercrimes in Ethereum since 2017. The report illustrated that fraud constituted the majority of all cryptocurrency-related crimes, indicating that the issue of transaction security had become an important issue in the blockchain ecosystem.

Traditional methods of mining Ethereum transactions aim to identify frauds by transaction characteristics. However,

Manuscript received 30 November 2021; revised 2 March 2022; accepted 30 March 2022. Date of publication 15 April 2022; date of current version 31 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62072391 and Grant 62066013, in part by the Key Research and Development Program of Yantai under Grant 2020XDRH092, in part by the Doctoral Startup Fund of Yantai University under Grant JS19B77, and in part by the Shandong Key Laboratory of Marine Ecological Restoration under Grant 201920. *(Corresponding author: Zhaowei Liu.)*

The authors are with the School of Computer Science and Engineering, Yantai University, Yantai 264005, China (e-mail: wangyixian@s.ytu.edu.cn; lzw@ytu.edu.cn; jindong.xu@nlpr.ia.ac.cn; wqyan@tju.edu.cn).

Digital Object Identifier 10.1109/TCSS.2022.3164719

these approaches can only mine the simple transaction, and the mining of the behavior is mostly done by the experts' detailed overview of the characteristics of the transaction, which cannot detect unpredictable and complex characteristics of the frauds, and this method cannot ensure the real-time mining for the increasing transactions.

The current analysis methods for Ethereum transactions are similar to most of the transaction analysis methods, where the information of transactions is generalized to the network structure and the characteristics of the transactions are analyzed based on the characteristics of the network structure. In addition, machine learning has attracted a lot of attention in the field of large-scale data analysis due to its high accuracy, speed, automation, and scale when dealing with large-scale datasets and has been applied to wearable systems [15], as well as gesture recognition other cross-cutting fields [16]. Meanwhile, there has been emerged [17] an approach-based machine learning to analyze the identity of Ethereum account and smart contracts. Compared with the traditional identification of Ethereum account addresses by manual annotation or code analysis, although it has better usability for the huge number of Ethereum, it presents further challenges on how to make better use of the structure and attributes in the network structure.

In this article, we propose a feature representation-based heterogeneous network embedding method to identify Ethereum accounts by classifying the mined embedding results as phishing nodes and normal nodes through a node classification task, which is significance for the identification of fraudulent account and the avoidance of fraudulent transactions in Ethereum in the future. We represent attribute information in the network and use it as an attribute in the heterogeneous network after building the transaction network from the Ethereum data. Finally, we embed the whole heterogeneous network into a low-dimensional space, obtain the embedding vector, and use the node classification task to classify the nodes in the heterogeneous network into fraud accounts and normal accounts for the purpose of Ethereum identity identification. The main contributions of this article are given as follows.

1) We applied a heterogeneous network representation learning approach to mining Ethereum identity information.
2) We present attributes in the Ethereum transactions and map it to low-dimensional representations through representation strategies based on transaction time and transaction amount.
3) We utilize the information that exists in the nodes and edges in the heterogeneous network of Ethereum transactions and add attribute information while retaining the characteristics of the heterogeneous network.

The other part of this article is given as follows. Section II presents the relevant work. Section III describes the framework of this article. Section IV conducts the evaluation and analysis of the experiment. Finally, Section V summarizes the work of this article.

II. Related Work

Recently, Ethereum has become a widely adopted cryptocurrency trading platform due to its own characteristics of centralization, security, and anonymity and has been further developed in various applied works driven by blockchain technology. Compared with other cryptocurrency transactions, Ethereum transactions support smart contracts so that the transactions related to it are open, transparent, immutable, and jointly maintained. At the same time, there are many types of scams in Ethereum, and research on specific types of scams has attracted more and more interest.

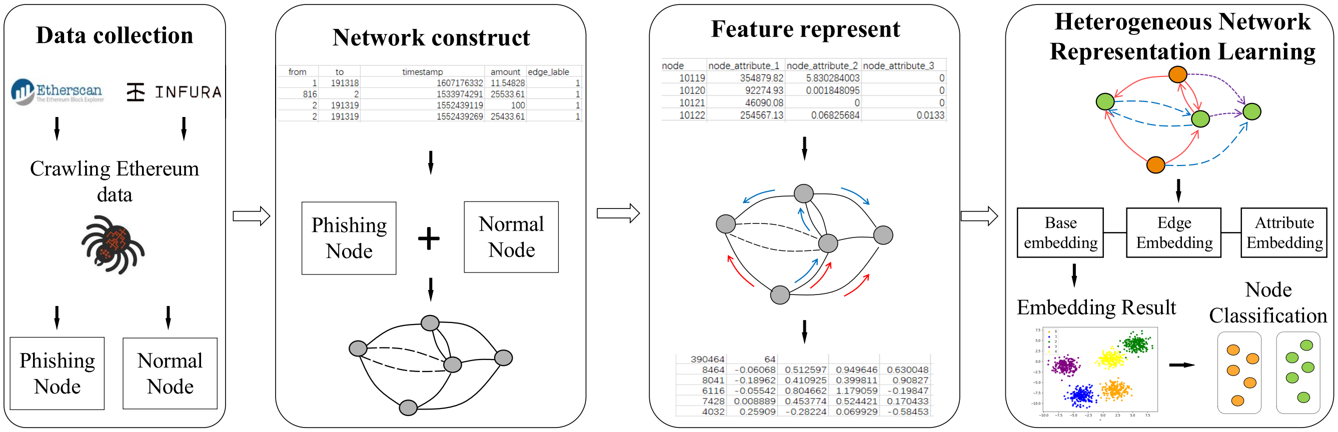
At present, there are two ways to detect scams in Ethereum. One is to detect scams existing in smart contracts [18], and the other is to detect Ethereum through graph learning. Graph learning can be used to analyze the transactions and then achieve the purpose of anomaly detection [8] or identification [19]. Although the detection method of smart contracts has high accuracy, for a large number of contract types in Ethereum, this detection method has certain insufficient in comprehensiveness.

The method of identifying fraudulent accounts in Ethereum using graph learning is more flexible and changeable, so it is a more preferable option. However, due to the huge amount of data in the transaction, how to complete the detection quickly and efficiently is also a major challenge. Network representation learning or network embedding, graph embedding [20], is a relatively mature method to map the information in the network to a low-dimensional space, which can effectively represent the information of the network. It is also applied to various machine learning tasks. Since real networks usually consist of tens of millions of nodes and millions of edges, in order to apply the network embedding method to large networks, LINE [21] uses breadth-first search to first obtain the node sequence- and second-order neighborhoods, and applying the embedding results in a large-scale network with weights. Node2vec [22] uses a biased random walk strategy controlled by tuning hyperparameters to preserve node–neighborhood relationships in the original network as much as possible in a low-dimensional space.

However, how to effectively preserve multiple transactions and account types in Ethereum is the problem, and Chen *et al.*. [23] classified transaction data into transfer transactions according to different transaction categories performed in the collected dataset, create smart contracts, and call smart contracts, constructed three graphs of transaction data, and analyzed their security issues.

Heterogeneous network representation learning has attracted extensive attention due to mapping of the rich structural properties and category information contained in complex heterogeneous networks into low-dimensional spaces and has been applied to disease gene prediction [24], disease-associated factor prediction [25], question routing recommendation [26], task identification, illegal transactions, and account identification [27].

In Metapath2vec [28], in order to consider the node types when walking, a meta-path-based method is proposed, which predefined the change of node types, considering the



from	to	timestamp	amount	edge_lable
1	191318	1607176332	11.54828	1
816	2	1533974291	25533.61	1
2	191319	1552439119	100	1
2	191319	1552439269	25433.61	1

node	node_attribute_1	node_attribute_2	node_attribute_3
10119	354879.82	5.830284003	0
10120	92274.93	0.001848095	0
10121	46090.08	0	0
10122	254567.13	0.06825684	0.0133

390464	64			
8464	-0.06068	0.512597	0.949646	0.630048
8041	-0.18962	0.410925	0.399811	0.90827
6116	-0.05542	0.804662	1.179059	-0.19847
7428	0.008889	0.453774	0.524421	0.170433
4032	0.25909	-0.28224	0.069929	-0.58453

Fig. 1. Overall framework for mine Ethereum transaction based on heterogeneous network representation learning method.

change of node or edge types when selecting the walk node. In GATNE [29], the GATNE-T and GATNE-I models are proposed to consider the difference between transductive learning and inductive learning, respectively. GATNE-T is to aggregate neighbors of different edge types to the current node and then generate a different vector representation for each edge type, while GATNE-I takes the initial characteristics of the nodes into account in order to handle unobserved data. Although the above heterogeneous network representation learning methods can map the network to a low-dimensional space, it is difficult to retain the attribute information of the network itself. Yuan *et al.*. [30] built Ethereum transaction records onto subgraphs and used improved Graph2Vec to extract latent features of subgraphs as address features for subsequent phishing classification.

The above heterogeneous or homogeneous network representation learning methods can significantly improve the representation effect, but there are difficulties in the feature representation of Ethereum transaction data, especially the analysis of Ethereum transaction data and fraud detection. According to the types of downstream tasks currently applying graph embedding, including node classification, link prediction, and node visualization, the application of tasks using the above graph embedding in Ethereum data has been appeared. Trans2vec [31] proposed a network embedding method based on transactions and timestamps in Ethereum transactions to extract features from transactions and identify phishing accounts in transactions through an unsupervised SVM classifier. Then, Liu *et al.*. [32] classified account into miners, transfer account, ICO account, and phishing account according to the characteristics of Ethereum in the node classification task and used the visualization of nodes to classify the classified account.

Subsequently, TWMDG [33] constructed transaction records as a temporal weighted multidigraph and used it to analyze and understand the dynamic transactions between Ethereum accounts through the task of link prediction. In addition, the same properties of TWMDG are applied to the task of node classification [34]. Furthermore, Lin *et al.*. [35] proposed a framework based on link prediction and quantified the influence of network features on the evolution of Ethereum from a microscopic perspective, and Bai *et al.*. [36] used a time window to construct the Ethereum transaction network as a temporal subgraph and mined the evolutionary behavior between the size of the Ethereum account and the transaction network and the transaction price of Ethereum.

In summary, network embedding-based data mining of Ethereum transactions is pioneering research in recent years, and these Ethereum transaction mining methods are classified into the following categories according to the task classification of network embedding: classification-based task [31], [34], link prediction-based task [33], [35], visualization-based task [32], and dynamic network evolution-based task [36], while most of these networks are homogeneous, some of them are using existing embedding strategies, and others are embedding strategies designed based on Ethereum transactions, offering the possibility of complex network mining approach [30].

III. Framework

In order to mine Ethereum transaction records sufficiently, we propose a four-part framework, which is shown in Fig. 1: 1) data collection; 2) network construction; 3) random walk-based feature representation; and 4) heterogeneous network representation learning.

A. Data Collection

Due to the openness of Ethereum, we can independently access the transaction records of it. Besides, we collect Ethereum transaction records by running the Ethereum client, which synchronizes all historical transaction records in Ethereum. Since each Ethereum client contains the history of all transactions, we query the transaction information of each account according to the API provided by Etherscan (etherscan.io) and extract the "time difference between the first and last transaction," "account balance," and "minimum amount of Ether received by the account" of each account as the node characteristics. Then, the first-order transaction data of each account are obtained, and 1 048 576 transaction

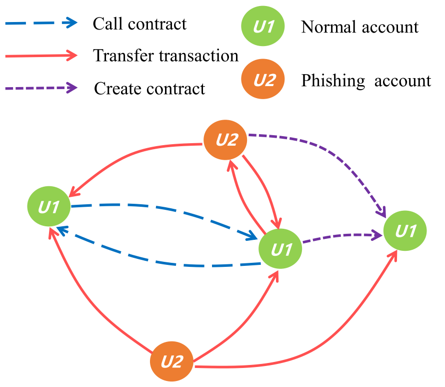


Fig. 2. Attributed heterogeneous network built on Ethereum transactions.

record datasets are finally obtained. Each account corresponds to a unique address, and each transaction represents a flow of behavior between a pair of addresses. In the end, the information we collected was used for network construction.

According to the different transaction characteristics of Ethereum accounts, the accounts in the data collection are divided into two parts: phishing accounts and normal accounts. For the collection of phishing accounts, we collected data on transactions marked as phishing frauds from an authoritative website EtherScamDB (https://etherscamdb.info/scams) that reported various illegal activities on Ethereum, and fraudulent information guides Ethereum investors away from possible fraud. In addition, the various fraud reports on the website not only show the content of the fraud but also address the suspect of fraud. We use a GET request to parse the available addresses in the returned JSON object to obtain the addresses of phishing accounts that have transacted with other accounts.

All normal nodes are also obtained through Etherscan, but considering that the time spent collecting a large number of externally owned accounts (EOAs) is huge, only successful transactions with a nonzero value are included in the dataset, and set the block height range from 10 876 500 to 10 877 500 for all nodes. The block height is the identifier of the block, which refers to the position of the block in the blockchain. After filtering out nonunique accounts, 5366 untagged accounts were randomly selected as normal accounts.

B. Network Construction

After collecting the transaction dataset, due to the different sources of data collection, the accounts are divided into normal and phishing accounts. When establishing the network, according to the account type, the corresponding node type is also normal or phishing node. The transaction network behavior established based on the collected dataset is shown in Fig. 2.

According to the data collection part, most of the main behaviors in Ethereum are related to transactions, and the attribute of each transaction is also particularly important. Therefore, in order to fully express transaction information clearly, we adopt an attributed heterogeneous network to represent the collected transaction data. The specific network is defined as $G = (V, E, X, M)$, which is similar to the heterogeneous network, in which each node v has a unique node type $z \in Z$, corresponding to the account in the dataset type. In addition, each edge e has a unique edge type $r \in R$ corresponding to the transaction types in the dataset. Also, if and only if $|Z| + |R| > 2$ is regarded as an attributed heterogeneous network, X/M has similar characteristics to the attributed network, $X = x_i | v_i \in V$ is the attribute set of all nodes, where x_i is the attribute associated with node v_i, and $M = m_i | e_i \in E$ is the attribute set of all edges, where m_i is the attribute associated with a certain edge e_i.

C. Random Walk-Based Feature Representation

In recent years, network representation learning methods based on random walks have been widely proposed and applied to network feature extraction. Taking DeepWalk and Node2vec as examples of the network representation learning method based on random walk, the nodes in the network are mapped out through the mapping function $f : V \to \mathbb{R}^{|V| \times d}$, in which the structural information is retained while maximizing the probability of neighbor nodes appearing in the d-dimensional feature space.

The process of feature representation consists of three parts. The first part is to generate random walks, which is used to capture the structural relationship between nodes. The second part is different walk strategy, which is used to capture different information between nodes. The final part is the skip-gram architecture, which is used to learn node embeddings by solving maximum likelihood optimization problems.

1) Random Walk: Given the source node u, we obtain the walk sequence l between vertices by using a random walk. Assuming that the first node of the sequence is c_0, then the nodes between the walk sequence l are represented by c_i. The probability that the next node of c_{i-1} is x_i is

$$\Pr(c_i = x \mid c_{i-1} = u) = \begin{cases} \frac{\pi_{ux}}{Z}, & \text{if } (u, x) \in E \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where Z represents the normalization constant.

2) Sample Strategy: We still use a random walk method to obtain the neighbor sequence of a node. The difference is that in the Ethereum transaction network, the edge between a pair of nodes usually includes the transaction amount or the timestamp of the occurrence. In order to express the indispensability of information in the transaction network more pertinently, we propose three biased random walk strategies.

1) *Strategy 1 (Walk Strategy Based on Timestamp):* Since each edge between two nodes corresponds to a timestamp, we assume that the later event occurs in the transaction, the closer relationship between two nodes. Therefore, V_u is used to represent the set of nodes directly connected to the node u, and we use mapping functions T to map real timestamps to discrete time steps

$$PT_{ux} = \frac{T(u, x)}{\sum_{x' \in V_u} T(u, x')} \tag{2}$$

where $T(u, x)$ represents the timestamp of the latest transaction between nodes u and x.

2) *Strategy 2 (Walk Strategy Based on Transaction Amount):* In addition to the timestamp of the transaction, each transaction in Ethereum is accompanied by a transfer transaction and GAS consumption, if the transaction amount between two nodes is larger, we believe that the relationship between the two connected nodes is closer. The total transaction amount is calculated by the linear function summation method, and the sampling probability is generated by PA_{ux}

$$PA_{ux} = \frac{A(u,x)}{\sum_{x' \in V_u} A(u,x')} \tag{3}$$

where $A(u,x)$ represents the total amount of transactions between nodes u and x.

3) *Strategy 3 (Walk Strategy Combining Transaction Amount and Timestamp):* Besides, in order to take the timestamp and transaction amount into account, we consider the transaction amount and transaction timestamp to be equally important. Similar to the above two strategies, we also regard the number of transactions and the sequence of timestamps as a walk strategy. V_u is used to represent the set of nodes directly connected to node u, $A(u,x)$ represents the total amount of transactions between nodes u and x, and $T(u,x)$ represents the latest transaction between nodes u and x timestamp expressed by the formula as

$$PA_{ux}T_{ux} = \frac{A(u,x)T(u,x)}{\sum_{x' \in V_u} A(u,x')T(u,x')}. \tag{4}$$

Finally, in order to balance the walk strategy, we use the hyperparameter q to control the walk preference. The probabilities obtained by different walk strategies are given as follows:

$$\pi_{ux}(q) = \begin{cases} PT_{ux}, & \text{if } 0 < q < 0.5 \\ PA_{ux}T_{ux}, & \text{if } q = 0.5 \\ PA_{ux}, & \text{if } 0.5 < q \leq 1. \end{cases} \tag{5}$$

The time complexity of our feature representation Algorithm 1 is O(nul), where u is the number of nodes, n is the number of walks per node, and l is the walk length.

3) Feature Representation and Optimization: The function $f(u)$ is a mapping function that maps node u to an embedding vector. For each node u in the network, by defining different sampling strategies s, the collection of all neighbors $Ns(u)$ of u of the source node is sampled. The optimization goal of the function $f(u)$ is to maximize the probability of the neighbor node $n \in Ns(u)$ given the source node u. In fact, we use skip-gram to optimize the following objective function:

$$\max_{f} \sum_{u \in V} \log \Pr(Nc(u) \mid f(u)). \tag{6}$$

The optimization method for the function f is further solved by using the stochastic gradient descent method.

D. Representation Learning for Attributed Heterogeneous Network

Through the part of feature representation, we can select a biased random walk strategy and represent representing the connection between nodes in a low-dimensional space. Specifically, we use the mapping function f in Algorithm 1 to map the connectivity of nodes to m_i and finally use M to represent the node information in the transaction network

$$f(u_i) \to m_i, \quad m_i \in M. \tag{7}$$

Algorithm 1 Random Walk-Based Feature Representation

```
Input: Transaction network $G = (V, E)$ search bias parameter $q$, embedding dimension $d$, walk length $l$, number of walks per node $n$, neighborhood size $k$
Output: Mapping function $f$
  Chosen $P_i$ using Eq. (2), Eq. (3), Eq. (4)
  Calculate $P_i$ using Eq. (5)
  $\pi = P_i(G, q)$
  $G' = (V, E, \pi)$
  walks = []
  for $iter = 1$ to $n$ do
    for each node $u \in \mathcal{V}$ do
      walks=strategy($G', u, l$)
      Append walk to walks
    end for
  end for
  $f$ = StochasticGradientDescent($k$, $d$, walks)
  Optimization Eq. (6) by random gradient descent
  return $f$
```

Then, we combined with the representation learning method of the attributed heterogeneous network to conduct further research on the Ethereum transaction network.

More specifically, we proposed a method, in which in order to represent an attributed heterogeneous network, we divide the entire representation learning into three parts: the representation learning of specific nodes v_i, the representation learning of different types of edges r connected to the node v_i, and the representation learning of the attributes of nodes and edges; all in all: base embedding, edge embedding, and attribute embedding. The specific representation structure is shown in Fig. 3.

For the basic embedding, we learn the embedding of the attribute x_i corresponding to node v_i through a neural network or linear transformation by parameterizing the function u_z, that is, $\mathrm{u}_z(x_i)$. u_z is a conversion function and corresponds to the type of node z. Since different nodes have different node types, it also corresponds to attributes x_i of different dimensions, and it can be used to learn nodes that do not appear in the test set. Usually, the parameterized function u_z uses graph neural network (GNN) [37] or multilayer perceptron (MLP), and in our algorithm, we choose GNN as parameterized function.

The basic embedding for a certain node v_i is usually expanded between different types of edges, where $N_{i,r}$ corresponds to all neighbor nodes of the node v_i of the edge type r. For the kth embedding $O_{i,r}^{(k)} \in \mathbb{R}^s, (1 \leq k \leq K)$ of the type r of the edge of the node v_i, it is formed by the edge embedding aggregation of the corresponding neighbor nodes, i.e.,

$$o_{i,r}^{(k)} = \text{aggregator}\left(\left\{o_{j,r}^{(k-1)}, \forall v_j \in N_{i,r}\right\}\right). \tag{8}$$

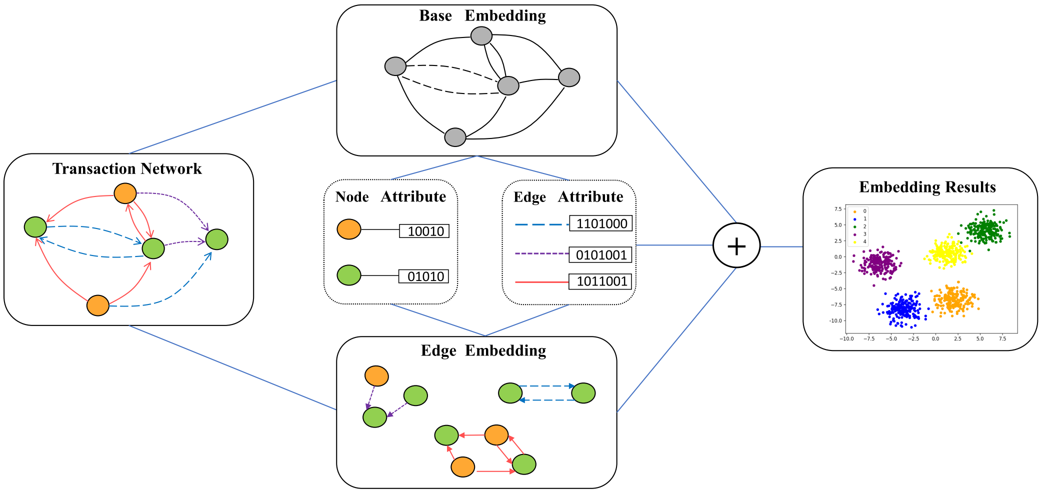


Fig. 3. Representation learning framework for attributed heterogeneous network.

For the initialization edge embedding $o_{\mathrm{i,r}}^{(0)}$ of the node v_i, a parameterized function $h_{z,r}$ is used for calculation, which is specifically expressed as $o_{i,r}^{(0)} = h_{z,r}(x_i)$, where x_i represents the v_i attribute of the node. More specifically, $h_{z,r}$ represents a conversion function that transforms the feature of node v_i into the edge embedding of node v_i corresponding to the edge type r. Besides, z also corresponds to the type of node v_i. In order to represent the collection of all edge embeddings of node v_i, we use O_i to represent

$$O_i = \left(o_{i,1}, o_{i,2}, \ldots, o_{i,m}\right). \tag{9}$$

The connection is made by m edge embeddings of s dimension, where m represents the number of edge types and s represents the dimension of each type of edge embedding. The parameter $a_{i,r} \in \mathbb{R}^m$ is calculated by using the self-attention mechanism [38], which represents the linear combination in O_i

$$a_{i,r} = \operatorname{softmax}\left(w_r^T \tanh \cdot (W_r O_i)\right). \tag{10}$$

In fact, w_r and W_r represent the learnable parameters of size d and $d \times s$, respectively, which are converted into matrix transformation by the superscript T

$$V_{i,r} = u_z(x_i) + \alpha C_r^T O_i a_{i,r} + \beta D_z^T x_i + \gamma D_r^T y_i. \tag{11}$$

α, β, and γ are hyperparameters, which, respectively, represent the proportion of basic embedding, edge embedding, and attribute embedding in the overall embedding. D_z^T is the feature transformation matrix on node v_i corresponding to node type z, D_r^T is the feature transformation matrix on edge e_i corresponding to edge type r, where $u_z(x_i)$ is the corresponding basic embedding of node v_i, and $C_r^T \in \mathbb{R}^{s\times d}$ is a trainable transformation matrix, x_i corresponding to the attribute of node v_i and y_i corresponding to the attribute of the edge e_i.

We will further study and optimize the proposed model. We learn the embedding result according to the sequence of generating nodes through random walk and obtaining the walking sequence of nodes through skip-gram. Since our input network is heterogeneous, we use meta-path-based random walks.

Random walk is also used to generate node sequences and then perform skip-gram over the node sequences to learn embeddings. Since each view of the input network is heterogeneous, we use meta-path-based random walks. The random walk with length l on edge type r follows a path $P = (v_{p1}, \ldots, v_{pl})$ such that $(v_{pt-1}, v_{pt}) \in E_r (t = 2, \ldots, l)$ denote v_{pt} context as $C = \{v_{pk} | v_{pk} \in P, |k - t \,| \leq c, t \neq k\}$, where c is the radius of the window size. Thus, given a node v_i with its context C of a path, our objective is to minimize the following negative log likelihood:

$$-\log P_\theta\left(\left\{v_j \mid v_j \in C\right\} \mid v_i\right) = \sum_{v_j \in C} -\log P_\theta\left(v_j \mid v_i\right). \tag{12}$$

Then, following metapath2vec and GATNE, we use the heterogeneous softmax function, which is normalized with respect to the node type of node v_j. Specifically, the probability of v_j given v_i is defined as

$$P_\theta\left(v_j \mid v_i\right) = \frac{\exp\left(c_j^T \cdot v_{i,r}\right)}{\sum_{k\in V_t}\left(c_k^T \cdot v_{i,r}\right)}. \tag{13}$$

Finally, we use heterogeneous negative sampling to approximate the objective function $-\log P_\theta(v_j|v_i)$ for each node pair (v_i, v_j)

$$E = -\log \sigma\left(c_j^T \cdot v_{i,r}\right) - \sum_{l=1}^{L} E_{v_k}\left[\log \sigma\left(-c_k^T \cdot v_{i,r}\right)\right] \tag{14}$$

where $\sigma(x)$ is the sigmoid function, L is the number of negative samples corresponding to a positive training sample, and v_k is randomly drawn from a noise distribution $P_t(v)$ defined on node v_j's corresponding node set V_t.

We summarize our algorithm in Algorithm 2. The time complexity of our random walk-based algorithm is $O(nmdL)$, where n is the number of nodes, m is the number of edge types, d is the overall embedding size, and L is the number of negative samples per training sample $(L \geq 1)$.

Algorithm 2 HNRL

```
Input: Transaction network $G = (V, E, X, M)$, embedding dimension $d$, edge embedding dimension $s$, attribute dimension $a$, learning rate $\eta$, coefficient $\alpha$, $\beta$, $\gamma$ , parameters $\theta$
Output: overall embeddings $v_{i,r}$ for all nodes on every edge type $r$
  Initialize all the model parameters $\theta$
  Generate random walks on each edge type $r$ as $P_r$
  Generate training samples $(v_i, v_j, r)$ from random walks $P_r$ on each edge $(v_i, v_j)$
  While not converged do
    for each $(v_i, v_j, r) \in$ training samples do
      Calculate $v_{i,r}$ using Eq. (11)
      Sample $L$ negative samples and calculate objective function E using Eq. (14)
      Update model parameters $\theta$ by $\frac{\partial E}{\partial \theta}$
  return embedding $v_{i,r}$
```

IV. Experiments

In this section, we first introduce the details of transaction dataset and the evaluation criteria. Then, we explain the comparison baseline method in detail. Finally, we perform the node classification task on the Ethereum transaction dataset to demonstrate the effectiveness of our proposed algorithm and give an analysis of parameter sensitivity experimental results.

A. Datasets

In our collected Ethereum transaction dataset, there are 10 122 nodes and 1 048 576 edges. Each node is represented by two types of accounts, namely, phishing accounts or normal accounts. Each edge is represented as three types of transactions: create contract, call contract, and transfer transaction.

In order to facilitate the understanding of the behavior differences of different node types, as shown in Fig. 4, we analyzed the proportion of all node degrees in the graph. Compared with the phishing nodes, the normal nodes generally have a small number of degrees, which means that the phishing nodes send phishing multiple times. Message to complete the fraud has important implications for fraud analysis.

B. Evaluation Criteria for Node Classification

In academia, link prediction and node classification are two common downstream tasks being widely used to evaluate the quality of network embeddings obtained by different methods. Meanwhile, classification tasks based on complex networks are also applied in the field of DNA classification [39]. In fact, node classification is also a valuable issue in the blockchain platform. A series of studies on Ethereum has witnessed various illegal behaviors or scams, such as phishing, Ponzi, and money laundry, and the identity information of the Ethereum accounts can be effectively identified by classifying them in a node classification task. Besides, due to the unavailability of ground truth of Ethereum accounts, Lin *et al.* [34] evaluated temporal random walk-based graph embedding by node classification on realistic Ethereum data.

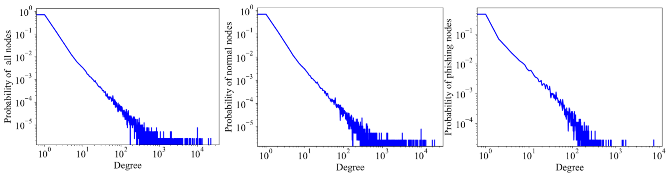


Fig. 4. Variation of degree under different proportions of ordinary nodes and phishing nodes.

In order to evaluate the effectiveness of the node classification experiment, we use some commonly used evaluation criteria, i.e., the area under the ROC curve (ROC-AUC) [40] and the PR curve (PR-AUC) [41] in our experiments, and also use F1 score as the third metric for evaluation.

Besides, in the task of node classification, it can classify the nodes in the network based on the observed information, and in this article, we classify the nodes present in the trading network as phishing nodes or normal nodes. Since the nodes we obtain are labeled, it makes no sense to classify the data with labels. To test the effectiveness of our proposed method, we will randomly select 70% of the nodes as the training set and 30% of the nodes as the test set and the validator. It is important to note that the nodes in the training set are labeled with the type of node, i.e., normal node or phishing node, while the type of node in the test set and validation set is infeasible.

C. Baseline Method

In the experiment, our proposed method is compared with two typical network embedding approaches based on the random walk, i.e., DeepWalk and Node2vec, and one popular heterogeneous network embedding approach based on the meta-path random walk, i.e., GATNE.

The GATNE model is proposed to learn the embedding representation of each node under different types of edges. The model supports both transductive (GATNE-T) and inductive learning (GATNE-I), and we choose GATNE-I as a baseline method.

To implement the abovementioned random walk based representation methods, we have several hyperparameters: the representation dimension d, the length of walk l, and walks per node n. In our comparative experiment, the parameter settings are $d = 64$, $n = 200$, and $l = 20$. For Node2vec, we set $q = 0.5$. Similarly, for heterogeneous network representation method GATNE, we set several hyperparameters: the base embedding dimension b and the edge embedding dimension s; in general, we set $b = 100$ and $s = 5$. For our proposed HNRL, we need to set two parts of hyperparameters. For random walk-based information representation, we set $d = 64$, $n = 200$, and $l = 20$. For heterogeneous network embedding, we set $b = 200$, $e = 15$, and $a = 15$.

In order to reflect the influence of different transaction characteristics on the overall effect of the model, we select $q = 0.25$ when the transaction amount is selected and $q = 0.75$ when the transaction time is selected and choose when the two are combined, $q = 0.5$.

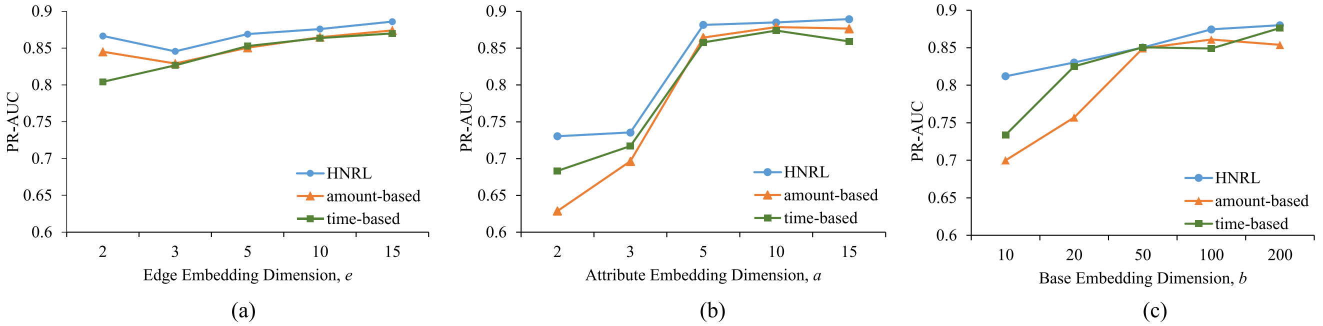


Fig. 5. Performance comparisons of different methods with various (a) base embedding b, (b) edge embedding e, and (c) attribute embedding a dimensions in PR-AUC.

TABLE I
PERFORMANCE COMPARISONS OF DIFFERENT EMBEDDING METHODS

Method	PR-AUC	ROC-AUC	F1 score
DeepWalk	0.682	0.695	0.644
Node2vec	0.723	0.746	0.687
GATNE-I	0.866	0.873	0.784
Time-based	0.845	0.881	0.863
Amount-based	0.866	0.883	0.879
HNRL	**0.889**	**0.959**	**0.957**

TABLE II
PERFORMANCE COMPARISONS OF DIFFERENT CLASSIFIERS

Method	PR-AUC	ROC-AUC	F1 score
Naive Bayes	0.739	0.753	0.775
Logistic regression	0.731	0.792	0.791
Decision tree	0.815	0.868	0.871
One-class SVM	**0.889**	**0.959**	**0.957**

D. Node Classification Performance

Based on the above given parameter settings, Table I compares the experimental results of each embedding method from three aspects: PR-AUC, ROC-AUC, and F1 score. Our proposed HNRL method is superior to other embedding results in all evaluation indicators with node classification task. In addition, it is obvious that due to the characteristics of the transaction network itself and the heterogeneous network representation method GATNE algorithm itself, the performance of the homogeneous network representation method DeepWalk and Node2vec is better. At the same time, the representation method based on transaction amount and transaction time is also better than DeepWalk, which is unbiased sampling. Two biased representation methods show that the embedding method based on transaction time has a better effect than the embedding method based on transaction amount in the node classification task. Also, it is illustrated in Table I that only extracting single transaction information does not ensure the best effect. The embedded method that combines transaction time and transaction amount shows the best performance.

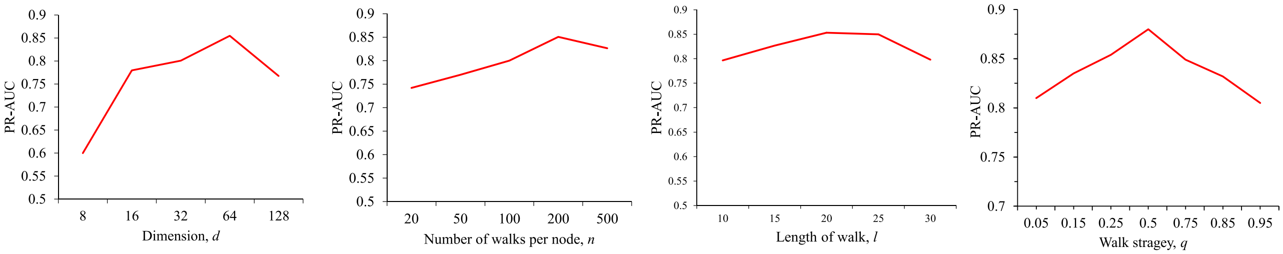
As shown in Fig. 5, based on the feature representation of three different walking strategies, we have continuously improved the embedding dimensions of base embedding, edge embedding, and attribute embedding, and the performance of three is improved with the dimension, which shows a better effect. For base embedding, the larger the dimension, the richer node information, and network structure can be retained. At the same time, the larger the dimension for edge embedding, the greater the information between different transactions. For attribute embedding, the larger the dimension, the retained the more abundant the attributes of nodes and edges. In addition, when the base embedding dimension is 200, the effect of HNRL is close to the effect of the embedding method based on transaction amount, and when the embedding dimension is 500, the difference between the two in the PR-AUC effect is obvious, which means with the base embedding. As the dimension increases, the embedding of multiple information is richer, so we choose 500 as the base embedding dimension.

Then, we use the classifier to classify the nodes into fraud and normal nodes, and the accuracy of the classified fraud nodes is equal to the accuracy of identification or illegal account detection. In addition, the choice of the classifier will also affect the recognition or detection results. In Table II, we compare four types of mainstream machine learning classification methods as baselines, select the embedding results of HNRL as the input, and choose the embedding dimension $d = 64$. The experimental results show that one-class SVM is better for detecting illegal accounts.

E. Parameter Sensitivity Analysis

For the proposed method, there exist a number of parameters, which may influence the results. In Fig. 6(a)–(c), we evaluate the effects of a series of parameters on the performance of HNRL on the node classification task on the Ethereum transaction network. When a particular parameter is under evaluating, all other parameters are set as default values. In this section, we also only consider PR-AUC for performance comparison.

We explore the effect of parameter embedding dimensions d of random walk-based information representation, as shown



(a) (b) (c) (d)

Fig. 6. (a)–(d) Results of parameters sensitivity analysis.

in Fig. 6(a). As the dimensions of feature representation increase, the algorithm can achieve better results. When the dimension reaches 64, the effect reaches its peak, but when the dimension is 128, the effect decreases because of the large difference between the dimension of the attribute embedding and the attribute.

We also examine the influence of the node's neighborhood parameters, including the number of walk per node n and walk length l. As shown in Fig. 6(b), with the increase of the number of per walk, it has been on an upward trend when n is less than 200, but it decreases when n is in the range of 200–500. As shown in Fig. 6(c), as the length of the walk length l increases, it reaches the maximum when $l = 20$, but since the walk length reaches saturation, there is no upward trend in the increase of l afterward. As shown in Fig. 6(d), when the walk strategy parameter q is varied, the chosen walk strategy differs and therefore varies in the performance of node classification. When $q = 0.5$ is a combined time and amount strategy, the performance reaches the peak, while when $0 < q < 0.5$ and $0.5 < q < 1$ due to the limitations of the walk strategy, it is always worse than the high performance when $q = 0.5$.

V. Conclusion

In this article, we propose a heterogeneous network representation learning method to characterize implicitly inside Ethereum transactions. Specifically, we build an Ethereum transaction network by collecting transaction data from normal and phishing Ethereum accounts. Then, we propose a walk strategy that combines the timestamps and amounts of transactions to represent the characteristics, and then to mine the types of nodes and edges, we use a method of representation learning for attributed heterogeneous network to map the transaction network to a low-dimensional space. Finally, we verify the validity of the results in the task of node classification, which has important implications for Ethereum identity identification. The experimental results show that our heterogeneous network representation learning method outperforms existing algorithms for analysis on the Ethereum transaction dataset.

The limitation of the proposed method is that although we construct the collected Ethereum transaction data in the form of a heterogeneous network and mine the implicit information in it, the network structures that appear in life such as social networks and knowledge networks are dynamic, the nodes or relationships that appear or disappear over time. In addition, the task of classifying nodes as a rubric for experiments can classify Ethereum accounts as either anomalous or normal, but the classification method can only classify abnormal behavior that already exists and is unknown for the first appearance of the category.

In the future, we consider building a dynamic heterogeneous network by weighting the time of Ethereum transactions and explore more types of Ethereum account identities.

References

[1] M. Iansiti and K. R. Lakhani, "The truth about blockchain," *Harvard Bus. Rev.*, vol. 95, no. 1, pp. 118–127, Jan. 2017.

[2] Y. Yuan and F.-Y. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.

[3] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] M. H. Joo, Y. Nishikawa, and K. Dandapani, "Cryptocurrency, a successful application of blockchain technology," *Managerial Finance*, vol. 46, no. 6, pp. 715–733, Aug. 2019.

[5] V. Buterin, "A next-generation smart contract and decentralized application platform," *Ethereum Project White Paper*, vol. 3, no. 37, pp. 1–36, 2014.

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2017.

[7] S. Lee *et al.*, "Cybercriminal minds: An investigative study of cryptocurrency abuses in the dark web," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.

[8] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, and Z. Zheng, "Phishing scams detection in ethereum transaction network," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–16, Feb. 2021.

[9] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, "Detecting mixing services via mining bitcoin transaction network with hybrid motifs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2237–2249, Apr. 2022, doi: 10.1109/TSMC.2021.3049278.

[10] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting Ponzi schemes on ethereum: Towards healthier blockchain technology," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 1409–1418.

[11] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–43, 2020.

[12] C. Team. (2021). *Crypto Crime Summarized: Scams and Darknet Markets Dominated 2020 by Revenue, but Ransomware is the Bigger Story*. [Online]. Available: https://blog.chainalysis.com/reports/2021-crypto-crime-report-intro-ransomware-scams-darknet-markets

[13] S. Kethineni, Y. Cao, and C. Dodge, "Use of bitcoin in darknet markets: Examining facilitative factors on bitcoin-related crimes," *Amer. J. Criminal Justice*, vol. 43, no. 2, pp. 141–157, Jun. 2018.

[14] M. S. Rana, C. Gudla, and A. H. Sung, "Evaluating machine learning models on the ethereum blockchain for Android malware detection," in *Proc. Intell. Comput. Comput. Conf.*, Cham, Switzerland: Springer, 2019, pp. 446–461.

[15] W. Qi and A. Aliverti, “A multimodal wearable system for continuous and real-time breathing pattern monitoring during daily activity,” *IEEE J. Biomed. Health Informat.*, vol. 24, no. 8, pp. 2199–2207, Aug. 2020.

[16] W. Qi, S. E. Ovur, Z. Li, A. Marzullo, and R. Song, “Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 6039–6045, Jul. 2021.

[17] H. Sun, N. Ruan, and H. Liu, “Ethereum analysis via node clustering,” in *Proc. Int. Conf. Netw. Syst. Secur.*, Cham, Switzerland: Springer, 2019, pp. 114–129.

[18] Y. Zhang, W. Yu, Z. Li, S. Raza, and H. Cao, “Detecting ethereum Ponzi schemes based on improved LightGBM algorithm,” *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 2, pp. 624–637, Apr. 2022, doi: 10.1109/TCSS.2021.3088145.

[19] S. Farrugia, J. Ellul, and G. Azzopardi, “Detection of illicit accounts over the ethereum blockchain,” *Expert Syst. Appl.*, vol. 150, pp. 1–11, Jul. 2020.

[20] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: Large-scale information network embedding,” in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[22] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[23] T. Chen *et al.*, “Understanding ethereum via graph analysis,” *ACM Trans. Internet Technol.*, vol. 20, no. 2, pp. 1–32, May 2020.

[24] K. Yang *et al.*, “HerGePred: Heterogeneous network embedding representation for disease gene prediction,” *IEEE J. Biomed. Health Informat.*, vol. 23, no. 4, pp. 1805–1815, Jul. 2019.

[25] Y. Xiong *et al.*, “Heterogeneous network embedding enabling accurate disease association predictions,” *BMC Med. Genomics*, vol. 12, no. S10, pp. 1–17, Dec. 2019.

[26] Z. Li, J.-Y. Jiang, Y. Sun, and W. Wang, “Personalized question routing via heterogeneous network embedding,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 192–199.

[27] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, “Key player identification in underground forums over attributed heterogeneous information network embedding framework,” in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 549–558.

[28] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable representation learning for heterogeneous networks,” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.

[29] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, “Representation learning for attributed multiplex heterogeneous network,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1358–1368.

[30] Z. Yuan, Q. Yuan, and J. Wu, “Phishing detection on ethereum via learning representation of transaction subgraphs,” in *Proc. Int. Conf. Blockchain Trustworthy Syst.*, Cham, Switzerland: Springer, 2020, pp. 178–191.

[31] J. Wu *et al.*, “Who are the phishers? Phishing scam detection on ethereum via network embedding,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 2, pp. 1156–1166, Feb. 2022, doi: 10.1109/TSMC.2020.3016821.

[32] X. Liu, Z. Tang, P. Li, S. Guo, X. Fan, and J. Zhang, “A graph learning based approach for identity inference in DApp platform blockchain,” *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 438–449, Mar. 2022, doi: 10.1109/TETC.2020.3027309.

[33] D. Lin, J. Wu, Q. Yuan, and Z. Zheng, “Modeling and understanding ethereum transaction records via a complex network approach,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 11, pp. 2737–2741, Nov. 2020.

[34] D. Lin, J. Wu, Q. Yuan, and Z. Zheng, “T-EDGE: Temporal WEighted MultiDiGraph embedding for ethereum transaction network analysis,” *Frontiers Phys.*, vol. 8, pp. 204–213, Jun. 2020.

[35] D. Lin, J. Chen, J. Wu, and Z. Zheng, “Evolution of ethereum transaction relationships: Toward understanding global driving factors from microscopic patterns,” *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 2, pp. 559–570, Apr. 2022, doi: 10.1109/TCSS.2021.3093384.

[36] Q. Bai, C. Zhang, N. Liu, X. Chen, Y. Xu, and X. Wang, “Evolution of transaction pattern in ethereum: A temporal graph perspective,” *IEEE Trans. Computat. Social Syst.*, early access, Sep. 20, 2021, doi: 10.1109/TCSS.2021.3108788.

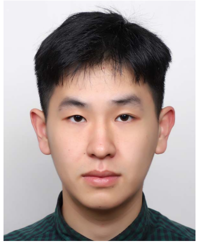
[37] J. Zhou *et al.*, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[38] Z. Lin *et al.*, “A structured self-attentive sentence embedding,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–15.

[39] P. Wu and D. Wang, “Classification of a DNA microarray for diagnosing cancer using a complex network based method,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 3, pp. 801–808, May 2019.

[40] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve,” *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[41] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.



Yixian Wang is currently pursuing the M.Sc. degree with the School of Computer and Control Engineering, Yantai University, Yantai, China.

His current research interests include blockchain and machine learning with graphs.

Zhaowei Liu received the Ph.D. degree from Shandong University, Jinan, China, in 2018.

He is currently an Associate Professor with Yantai University, Yantai, China. His research interests include blockchain and machine learning with graphs.

Jindong Xu was born in Zhaoyuan, Shandong, China, in 1980. He received the B.S. degree from Shandong University, Jinan, China, in 2003, and the M.S. and Ph.D. degrees from the College of Information Science and Technology, Beijing Normal University, Beijing, China, in 2008 and 2014, respectively.

His research interests include image processing, pattern recognition, and information fusion.

Dr. Xu was is a member of the China Computer Federation (CCF) and International Association for Mathematical Geosciences (IAMG). He received the Best Poster Paper Award of 15th IAMG Conference, Madrid, September 2013.

Weiqing Yan was born in 1988. She received the Ph.D. degree from Tianjin University, Tianjin, China, in 2017.

She was with the Visual Spatial Perceived Laboratory, University of California at Berkeley, Berkeley, CA, USA, from September 2015 to September 2016. She is currently an Associate Professor with the School of Computer and Control Engineering, Yantai University, Yantai, China. Her research interests include 3-D image editing, machine learning, and computer vision.