Visual Analytics for Phishing Scam Identification in Blockchain Transactions with Multiple Model Comparison

Jiaqi Dong
Fudan University
Shanghai, China
dexter_0511@163.com

Zishu Qin
Fudan University
Shanghai, China
zsqin18@fudan.edu.cn

Zhou Fang
Harvard University
Cambridge, MA, USA
fangzhou0407@163.com

Xuan Chen
Fudan University
Shanghai, China
chenxuan@fudan.edu.cn

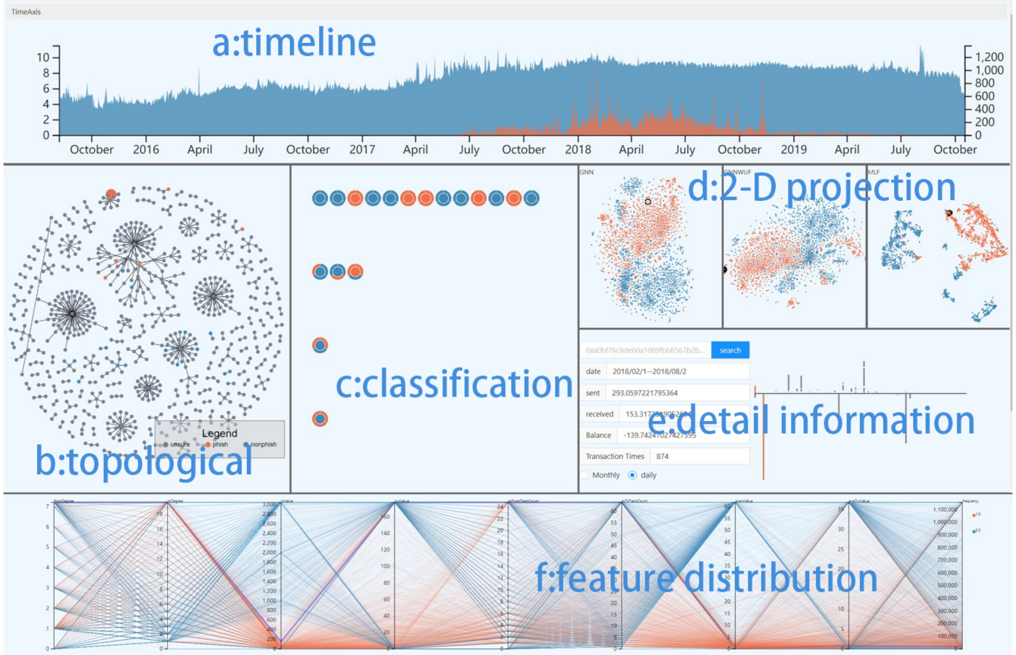
Zengfeng Huang
Fudan University
Shanghai, China
huangzf@fudan.edu.cn

Haoyun Guo
Independent Researcher
China
s28@msn.com

Richen Liu
Nanjing Normal University
Nanjing, China
richen@pku.edu.cn

Cagatay Turkay
Warwick University
Warwicks, UK
CagatayTurkay@warwick.ac.uk

Siming Chen*
Fudan University
Shanghai, China
simingchen@fudan.edu.cn



0xa0bf76c8de60a1089fbb8567b2b...
search

date 2018/02/1--2018/08/2

sent 293.0597221795364

received 153.317

Balance -139.74247027427595

Transaction Times 874

○ Monthly ● daily

Figure 1: Our visual analytical system facilitating a multi-model analysis of dynamic blockchain networks. It includes the (a) timeline: a histogram displaying transaction volume over a period; (b) topological view: a force-directed graph demonstrating network structures; (c) classification view: prediction results based on several feature extraction methods; (d) 2-D projection view: 2-D projections of sample embeddings; (e) detailed information view: detailed transaction information of a node; (f) feature distribution view: a parallel coordinate plot showing samples' features.

*Corresponding Author

VINCI 2023, September 22–24, 2023, Guangzhou, China

ACM ISBN 979-8-4007-0751-3/23/09…$15.00
https://doi.org/10.1145/3615522.3615540

ABSTRACT

The phishing scam is a major kind of fraudulence in blockchain. And it has become an urgent issue to discern and prevent the fraudulent behaviors. However, the large-scale and dynamic nature of transaction network imposes great challenges on the identification and analysis. While there have been many sophisticated machine learning approaches providing predictive capability in terms of detecting such cases, they usually offer little insight into the essence of those behaviors and the occasion when phishing scam activities happen. Motivated by these shortcomings and bottlenecks, this paper proposes a suite of visual analytical methods for interpretable and explorable fraudulence identification in large-scale blockchain transaction networks, incorporating an anomaly detection model based on multiple feature extraction manners. In this paper, we adopt two types of graph embedding methods and variable derivation to generate features from transaction data. Then we use machine learning classification approaches to fit the three sets of features. Evaluations show that all kinds of features perform well in classification. Besides, we design an interactive visualization system displaying the transaction networks and classification models, which allows users better explore the data and understand the models. Furthermore, we demonstrate two cases through the visualization system to unearth fraudulent patterns and interpret classification results. Finally, we close with discussions for further improvements of our models and system.

KEYWORDS

visual analytics, graph presentation learning, anomaly detection

ACM Reference Format:

Jiaqi Dong, Zishu Qin, Zhou Fang, Xuan Chen, Zengfeng Huang, Haoyun Guo, Richen Liu, Cagatay Turkay, and Siming Chen. 2023. Visual Analytics for Phishing Scam Identification in Blockchain Transactions with Multiple Model Comparison. In *The 16th International Symposium on Visual Information Communication and Interaction (VINCI 2023), September 22–24, 2023, Guangzhou, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3615522.3615540

1 INTRODUCTION

Recent years has witnessed a surge of interest in blockchain technology especially thanks to the popularity of the cryptocurrency Bitcoin [21]. Blockchain acts as an open ledger that stores transaction records in a decentralized, persistent and anonymous way [22] and has offered promising application prospects such as product traceability and anti-counterfeiting [10]. This trend, however, is also matched with an increasing number of security incidents on blockchain, including digital currency scams and hacking attacks. Among these incidents, phishing scams have been cited as the main threat to the security blockchain ecosystem [20]. With phishing scams causing significant financial losses on blockchain [4], it has become crucial to devise methods to identify them on blockchain.

The temporal, high-frequency nature of the blockchain transactions as well as the topological complexities of blockchain networks, make the fraudulence identification a technically challenging task. Thus there has been a growing interest in deep learning approaches [12, 18] in the past decade. More specifically, graph neural networks (GNNs) [17] have attracted attention. However, they usually fail to offer explanations referring to node features or transaction patterns indicating fraudulent behavior. As a result, although various machine learning methods are competent in classification, they are limited in terms of diagnostic and descriptive capabilities and offer limited insight into understanding and exploring the characteristics and time-varying transaction patterns of phishing nodes in a blockchain network.

To address those challenges, we design an interactive system providing multiple and complementary representations of blockhain networks, and a suite of visual analytical methods for the informative analytics of fraudulent behaviors among large-volume transaction data. In our approach, GNNs and variable derivation are utilized to generate several features of each node. To investigate nodes' relationships, we provide a force-directed graph to display transaction information in a topological space. To demonstrate the impact of different features on label prediction, we provide a classification view. We also use parallel coordinate plots to explore the transaction characteristics. To examine the evolution of transactions of a node, we provide a local network view to explore the time-varying characteristics of the selected nodes as well as a timeline to display and select time span for analysis. Our approaches offer dynamic network visualization methods for analysts to examine explicit and implicit information of the blockchain network from multiple perspectives concurrently.

We evaluate our approach on a data set emerging from a blockchain network with known phishing activities. We provide two case studies to demonstrate the efficacy of our approach in supporting phishing activity detection and analytics. Using our visual analytical approach, we identified several behaviors and features that characterise fraudulent blockchain activity.

The key contributions of this paper are as below:

- **Comparison of multiple models in anomaly detection in the blockchain network** through a comprehensive combination of Graphical Neural Network feature space, graph topology and dynamic network representation,
- **A visual approach for the multi-perspective analytics of dynamic networks** through a visual analytical system that supports the exploration of phishing node networks by providing interpretable visualizations of transaction patterns and behaviors.
- **Two real-world data case studies** demonstrating how our visual analytical approach could facilitate the identification and characterization of fraudulent behaviors on blockchain transaction networks.

2 RELATED WORK

2.1 Graph Data Mining

Matrix networks usually perform well in dense graphs mining[6, 8]. There are also hybrid approaches such as NodeTrix [5] that combine connectivity graphs and matrices. Pienta et al. [15] proposed a feature-based method to query a large graph from local neighborhoods of interest to the user. Heer and Boyd [3, 7] proposed a top-down strategy through which analysts can first have an overview of the network and then filter or query to reach local information from the network. In contrast, Moscovich et al. [13] proposed a bottom-up strategy, which requests to get local details first and then analyzes larger networks through corresponding nodes and edges of interest. Both strategies allow analysts to navigate between overall networks and local nodes to get both structural and informative details of the network.

2.2 Anomaly Detection

There has been a growth in security incidents in the blockchain space, including DeFi security is front, phishing incidents, ransom incidents and digital wallet security incidents. Data scientists have explored various approaches to examine these anomalies. Signorini et al. [16] proposed BAD (blockchain anomaly detection) which leverages the features of blockchain to provide an anomaly detection service and protect the peers in a blockchain network against eclipse attacks. Micha and Kamil [14] apply three supervised learning techniques, Random Forest, Support Vector Machine and XGBoost, to detect fraudulent accounts on the Ethereum blockchain. Damiano et al. [11] analyzed the outliers of Bitcoin user graph in degree distribution and found that these outliers of degree distribution are generated by artificial chains of transaction.

2.3 Blockchain Visual Analytics

The transactions based on blockchain are gowing rapidly in information capacity and effectively visualizing the transaction information has become imminent. Kinkeldey et al. [9] designed an actor-centric view of transactions, where non-technical experts can categorize participants through a workflow based on multiple activity indicators. Dan et al. [2] used force guide graph visualization to help accelerate the exploration of large-scale Bitcoin transaction data and provide collaborative models to detect unexpected but frequent transaction patterns, such as money laundering. Michele et al. [19] proposed a modular framework for parsing Bitcoin blockchain, which clusters addresses that may belong to the same entity, classifies and marks this entity and visualizes the complex information extracted from the Bitcoin network.

3 OVERVIEW

3.1 Data Description

The data was crawled from Etherscan [1]. There are 4,161,444 transaction records and 944,705 nodes, 3,360 labeled and 941,345 unlabeled. The labeled nodes consist of 1,660 phishing and 1,700 non-phishing (legitimate) nodes. Each node is represented by an address, identification for a specific Ethereum account. Each transaction contains following six variables:

Table 1: Variables of transactions and their meanings

Variable	Type	Meaning
TxHash	string	Hash value of the transaction
BlockHeight	integer	height of block cotaining the transaction
TimeStamp	integer	timestamp of the transaction
From	string	address of the transaction launcher
To	string	address of the transaction receiver
Value	float	transaction amount

Our approach will focus on the last four variables. We examine the relationships between transaction node addresses, weighing bias of transaction timestamp and transaction amount.

3.2 Analysis Tasks

There are various models for fraudulence detection. But the exploration for misclassification and dynamic methods to interpret their behavioral patterns are still inadequate. Therefore, we propose a visual analysis framework to address the following tasks:

T1: Temporal analysis. The data used in our work are transaction data, and the most conspicuous feature of such kind is large-volume. For a more efficient exploration of the node network, we need to control the volume of data in time domain. This can not only help us to enhance the front-end stability and management of data content, but also to analyze the evolution of the phishing nodes' transaction network dynamically in the time dimension.

T2: Transaction pattern analysis. Reading the literature reveals that from the regulator's perspective, it is important to identify fraudulent accounts, but it is equally important to determine whether an account directly or indirectly associated with a fraudulent account is a also fraudulent account. Therefore it is crucial to explore nodes' transaction pattern, or namely, structural characteristics in transaction network. On the one hand it can provide some insight to determine whether a node is a phishing node or not, and on the other hand it can show the fraudulent pattern of the node to some extent.

T3: Trace for fund flows. In real-world transactions, once fraud occurs, as is important to sanction fraudulent individuals in a timely manner. it is also necessary to monitor the flow of funds. This not only helps us to identify the associated accounts of the phishing node, but also provides the possibility to recover the lost funds. Therefore, the ability to trace the flow of funds is very essential.

3.3 Design Requirements

To accomplish the goals and analytical tasks discussed before, we summarize our four main design requirements as below (R1, R2 and R3 correspond to T1, T2 and T3 discussed above):

R1: Interactive temporal analysis and linked views. To efficiently control the data volume and explore the transaction data in time domain, there need to be a key view where we can conveniently select any time interval. The interrelationships between this view and others will be also necessary. Once we select a time interval, data displaying in other views must be updated subsequently for a coherent demonstration. Furthermore, this view may depict an overview of the transaction data within the selected interval.

R2: Topological representation of transaction network in global and local perspectives. The transaction pattern of an account means how the account owner makes transactions with others. This corresponds to topological structure of a node in the transaction network. Visualizing the network makes it easier and plainer to observe a node's neighbors and their interactions so that users can infer a node's label from this information. Two different scales of perspective will fulfill us both to seize an overview of the whole transaction network and check a node's topological structure more thoroughly.

R3: Query and display for detailed information. To trace the flow of a fund, it is necessary to obtain the detailed information of transactions itself and the two subjects of a transaction. Therefore, we need a query and browse tool for detailed information of nodes and transactions. After accessing the interested data, it is also helpful to efficiently display and visualize it in order to have a clear and straightforward grasp of the needed information.

4 DATA PROCESSING AND MODELING

Three data processing methods are taken to extract the structural and transactional features of nodes and to perform the node classifying mission. In this section, we will introduce them and compare their discriminative capacity.

4.1 Feature Derivation

In Blockchain transactions, every transaction contains two pieces of important information: timestamp and value. To explore the difference between phishing nodes and normal nodes, we use information from the two features to generate some new features. The derived features are shown in the following table:

Table 2: Derived variables and their meanings

Variable	Type	Meaning
node	string	node address
label	string	node class
fromDegree	integer	how many people it has transacted with as the initiator
outDegree	integer	how many people it has transacted with as the receiver
outValue	float	cumulative transfer-out amount
inValue	float	cumulative transfer-in amount
asFromTransCount	integer	cumulative number of transactions as the initiator
asToTransCount	integer	cumulative number of transactions as the receiver
frequency	float	(Last transaction time - First transaction time) / Total number of transactions
aveInValue	float	average transfer-in amount
aveOutValue	float	average transfer-out amount

4.2 GNN methods

In our approach, GNNs are used to generate the embedding of nodes. The encoded vectors capture structural characteristics of the graph in a high-dimensional format, thus can be employed as the features to build predictive machine learning models.

The embedding method in our paper is node2vec, a biased random walk to obtain the neighbor sequence of a vertex. Given the current vertex v, the probability of visiting the next vertex x is:

$$\mathbf{P}\left(c_i = x | c_{i-1} = u\right) = \begin{cases} \frac{\pi_{ux}}{z} & \text{if } (u,x) \in E \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

π_{ux} is the unnormalized transfer probability between vertex u and vertex x. z is the normalized constant.

node2vec controls the random walk strategy through two hyperparameters - p and q. Suppose that the current random walk arrives at vertex u through edge (t, u). We set $\pi_{ux} = \alpha_{tx} \cdot w_{ux}$, w_{ux} as the edge between vertex u and x:

$$\alpha_{tx} = \begin{cases} \frac{1}{p} & \text{if} d_{tx} = 0 \\ 1 & \text{if} d_{tx} = 1 \\ \frac{1}{q} & \text{if} d_{tx} = 2 \end{cases} \tag{2}$$

d_{tx} is the shortest path distance between t and x. Parameter p is a return parameter that controls the probability of returning to the source node t. Parameter q is an in-out parameter that controls the probability of going away from the source node t.

Referring to Wu et al. [20], we define PA_{ux} as the transition probability from u to a neighbor x based on the transaction amount:

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

$A(u,x)$ denotes the total amount of transaction value between u and x. V_u represents the set of nodes directly connected to u.

Meanwhile, we define PT_{ux} as the transition probability based on transaction time:

$$PT_{ux} = \frac{T(u,x)}{\sum_{x' \in V_u} T(u,x')} \tag{4}$$

$T(u,x)$ denotes the timestamp of the last transaction between u and x, V_u represents the set of nodes directly connected to u.

In order to take both transaction time and transaction amount into account, a parameter α is used to balance their effects and a product act as the edge weight w_{ux}:

$$w_{ux} = PA_{ux}^{\alpha} \cdot PT_{ux}^{1-\alpha} \tag{5}$$

Then the transition probability from u to x can be derived as :

$$\pi_{ux} = \alpha_{tx} \cdot PA_{ux}^{\alpha} \cdot PT_{ux}^{1-\alpha} \tag{6}$$

After performing the biased random walk, we obtain node sequences. Then we use skipgram to learn the embeddings.

In our case, we set $\alpha = 0.5$, $p = 0.25$, $d = 64$, $r = 20$, $l = 5$, and $k = 10$. Then we get a 64-character embedding. When we disregard the weights of edges and consider only the connection relations between nodes, another embedding can be obstained. The weighted and unweighted node2vec methods are denoted as GNN and GNNWUF respectively.

In order to test whether node2vec along with variable derivation can identify nodes' characteristics, we utilize them to gain the features and then perform classification. The ROC results of different models are shown in the following table:

Table 3: ROC of each classification method

	Logistic regression	SVM
GNN	0.95	0.97
GNNWUF	0.94	0.96
Derived variables	0.94	0.93

5 VISUALIZATION SYSTEM

After data modeling and processing, we get the several sets of features generated by transaction data. In this section, we will use this information to design a visual analytic framework.

The whole framework of our visualization system consists of 7 views: (1) timeline; (2) topological view; (3) classification view; (4) 2-D projection view; (5) detailed information view (6) feature distribution view (7) node transaction network view (Fig.1)

5.1 Timeline

The timeline (Fig.1 a) plays a vital role in the whole design because it provides interactions in terms of time and acts as the prerequisite for several other views' generations. Above the time axis is a bicolor histogram showing the volume of transactions during a specific period. The height of blue area represents the total amount of transactions at this timestamp in reference to the left y-axis. The height of orange area figures out the transaction amount involving phishing nodes, with the right y-axis for reference. We take a logarithm for transaction volume to obtain an easier view.

The timeline facilitates us to observe the amount of transactions at each timestamp, and demonstrates when the phishing nodes show up for fraudulence detection and exploration. Besides, we can use it to decide how many transactions for display. Users can brush the time axis to select a specific time interval, and the data in it will be transferred to the topological view and classification view.

5.2 Topological View

In order to observe global transaction relationships, we visualize a topological space to demonstrate the network information. A force-directed graph is utilized for our topological view (Fig.1 b). The features of our force-directed graph are listed as following:

- A *Node* represents an address.
- The *color* of a node represents the node type: orange for fraudulent, blue for normal and yellow for unknown.
- The *edge* between two nodes indicates that they conduct transaction at least once with each other in the time range.

Several interactive functions are incorporated. First, the user can drag to draw a specific area on the graph to zoom in. Information in other views will also be updated according to the selected nodes. Another function is click. If the user clicks on a specific node, the local transaction network involving this node will be highlighted. In addition, when hovering on a node, its address will be displayed.

With all the features and functions, the topological view provides an interactive and dynamic interface for users to examine the global and local information of the blockchain transaction network.

5.3 Classification View

In data processing and modeling, three sets of features are generated by GNN, GNNWUF and variable derivation and adopted for label prediction. In the classification view (Fig.1 c), we can have a plain sight of the prediction results and the true label of nodes.

The prediction results are demonstrated by three sectors around each node in this view. The colors of the top-right, top-left and bottom sectors of each node respectively represent the prediction results using features generated by GNN, GNNWUF and derived variavles, orange for fraudulent and blue for normal. And the color in the center of each node represents its true label.

Nodes are staggered into four rows from top to bottom. The first row represents that all three feature generation methods lead to correct prediction results. The second and third rows respectively indicates two and one methods give rise to right predictions for this node. And the last row consists of the nodes whose true label is inconsistent with all the predictions induced by three methods.

In terms of interactive function, one can click on a node. Then the corresponding node in other views will receive a concentration effect. For instance, the node in 2-D projection view will be circled and the area around it in topological view will be enlarged. More parts of the interactions between the classification view and other views will be illustrated below.

5.4 2-D Projection View

The 2-D projection view (Fig.1 d) includes 2-D points projected from high-dimensional vectors produced by GNN, GNNWUF and variable derivation, using t-SNE algorithm. t-SNE is a nonlinear dimension reduction algorithm, and the basic idea is that that are similar points in high-dimensional space are mapped to be similar in low-dimensional space as well. As can be seen from the figure, the feature projections are roughly divided by the color in each subview, conceivably showing the features' capacity in classification.

5.5 Detailed Information View

The detailed information view (Fig.1 e) enables users to have a better grasp of the phishing nodes' features and actions by providing both thorough data and plain visualization for the account state.

This view can be divided into two parts. The left part is an account information query and display window. Users can attain a general understanding of a node's transaction information through this window. At the top is an input box to enter an address. Clicking on the search button, the information in display includes the time span, cumulative amount launched, cumulative amount received, account balance and number of transactions.

The right part shows the receiver of the transaction. Bars above and below the horizontal line represent the transfer-in and transfer-out cash flow. Each column denotes one day by default. The height of each bar indicates the volume of the transaction, and the color symbolizes the label of the transaction receiver. When hovering on a bar, the address of the transaction receiver will be shown. Furthermore, that address will be assigned to the clipboard once clicking on the bar. From this view we can trace the flow of funds and it also helps us to detect phishing nodes.

5.6 Feature Distribution View

The feature distribution view (Fig.1 f) is a parallel coordinate diagram. It shows the distribution of the normal and phishing nodes' derived variables after median depolarization. Each axis represents a feature and each consecutive line from left to right denotes a node. From this view, we can see what kind of characteristics the distribution of each feature has for fraudulent and normal nodes. For example, the total and average transfer-in amount of phishing nodes are lower compared to non-phishing nodes.

When we click on a node in the classification view, the line corresponding to that node will be highlighted as a bold purple line. Consequently, we may get an understanding of the mechanism of why it is misclassified or correctly classified by observing and comparing the feature distribution.

5.7 Node Transaction Network View

The node transaction network view (Fig.5) demonstrates the second-order transaction network of a selected node. After clicking on the node we are concerned about in the classification view, the front-end page will send a request to the database to retrieve the second-order transaction network data that exists in the transaction time span of the node and then utilize the data for plotting. The view will show up as a pop-up window where we can observe the dynamic transaction process of the node by dragging the progress bar at the top of the window to select a different time range.

The view is designed to help us better discover the temporal characteristics of node transactions and gain some insight into fraudulent transactions from the transaction network's evolution.

6 CASE STUDY

In this section, we present two case analyses based on real-world data from Etherscan and demonstrate how our system can provide visual analytical patterns for a blockchain transaction network. Each of our explorations starts with a brush for a selected time interval in the timeline.

Case 1: Transaction Network Overview and Trace for Fund Flow

In this case, the time interval from November 2017 to December 2017 is chosen. As can be seen from the timeline, there is a large scale of transactions involving phishing nodes over this period, while the total transaction volume is still within a manageable range.

First, we will have an overview of the initial topological view (Fig.2), from which we can find some nodes surrounded by a large number of other nodes. Exploring the other spiky regions of the timeline, we find that such super nodes are basically present in regions with high transaction volumes. In order to unearth the label of those special nodes, we consulted with experts and scholars in the field, who explained that there are two types of nodes that exhibit the characteristics of large transactions with multiple nodes. One of the types is ICO (Initial Coin Offering) nodes. They will raise a large amount of money during the initial token issuance, attracting many transactions. Additionally, this kind of nodes possess an obvious feature that they will make a lot of high-volume transactions in a short period, and never appear again. The other type is the transaction intermediary. They usually belong to an exchange and are used to transfer funds between different exchanges.

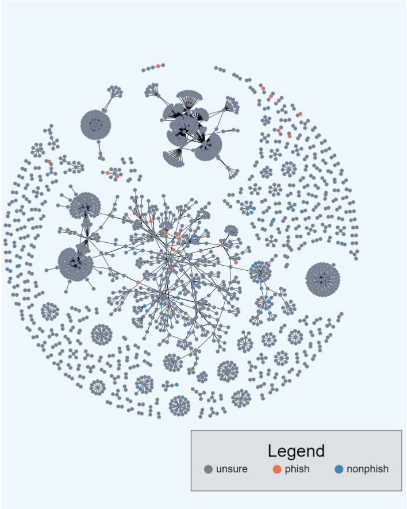
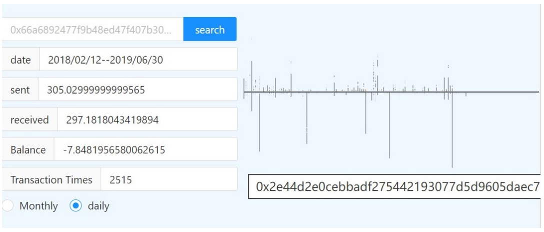


Figure 2: Topological view from November to December in 2017. There are some super nodes surrounded by lots of nodes, and such nodes basically are label-unknown. These nodes tend to belong to ICO or exchange nodes.

Hiding the unlabeled nodes, we observe that many phishing nodes are in the same community and are second-orderly connected through an intermediate node. From this phenomenon, it can be discerned that there is a collaborative fraudulent relationship among the phishing nodes.

In the classification view, we randomly choose a phishing node from the second row, whose address is 0x66a6892477f9b48ed47f407b30dde754405e1910 (Node A). Its label is correctly predicted by two feature extraction methods. After clicking, its second-order transaction network is displayed as a pop-up window. We can find a lot of phishing nodes in the local network, which proves the relationship of collaborative fraudulence.



0x66a6892477f9b48ed47f407b30...

date 2018/02/12--2019/06/30

sent 305.02999999999565

received 297.1818043419894

Balance -7.8481956580062615

Transaction Times 2515

○ Monthly ● daily

0x2e44d2e0cebbadf275442193077d5d9605daec7

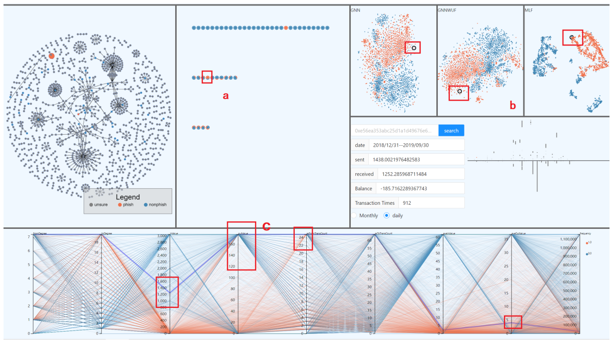
Figure 3: Detailed transaction information of a phishing node. It receives many small-volume transactions and launches a few large-volume transactions.

From the detailed information view (Fig.3), the selected phishing node made many small-volume transactions as the receiver, while the transactions launched by Node A were less but with larger value. Moreover, We note that Node A transferred most of the funds to 0x2e44d2e0cebbadf275442193077d5d9605daec7b (Node B), so we can assume that this node is important to the phishing node. Hence we conduct a search for the transaction details of this node. From the retrieved transaction details, we observe that most of its funds originate from the phishing Node A, with the total amount being 58.8. While the main transferee to that node is 0xd64f9a7f4ff1e6674a72a2a0f7e1c0f0aeedf6fe (Node C). So we make a further exploration to Node C, discovering that it has received only one sum of fund and the source is Node B. Therefore, we have an adequate motivation to suspect Node B and Node C to be the associated nodes of the initial phishing node, A, or the members of an identical fraudulent collection. By tracing the flow of funds, we can dig out some key suspicious nodes, which also provides the possibility of recovering the fraudulent funds.

To verify whether other nodes would exhibit similar characteristics of the above two associated nodes, we search for the node with the second-highest transfer amount of this initial phishing node, A. The result is 0x8d212fe863dcf6c27bb77393592e13cd00c80bc9 (Node D). This node received funds of 51.4 from the Node A. Then we examine all its transaction receiver and find no such node with characteristics similar to the key associated node mentioned above. Consequently, the two associated nodes are very possible to be key nodes in collective fraudulent patterns.

Case 2: Exploratory Analysis for the Misclassified Nodes

Another purpose of our system design is to explore the reason for misclassification. We randomly selected nodes which are wrongly predicted by one, two or three sets of features for our exploration. First, for the nodes misclassified by one set of features, we randomly select one node of this type with the data from November 2018 to December 2018, and obtain the views shown below. (Fig.4)



0xe56ea353abc25d1a1d49676e6...

date 2018/12/31--2019/09/30

sent 1438.0021976482583

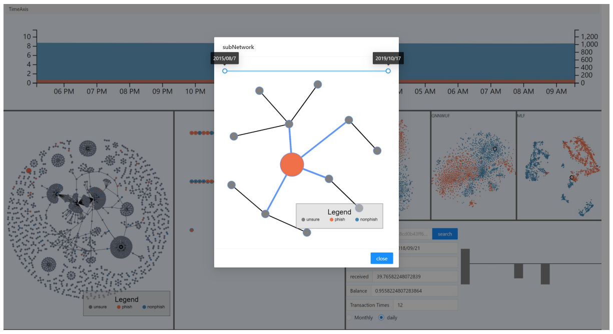
received 1252.285968711484

Balance -185.7162289367743

Transaction Times 912

Figure 4: Several views based on the data from November to December in 2018. The phishing node wrongly predicted by one feature extraction method is selected for exploration

From the classification view (Fig.4 a), we can learn that the classification based on derived variables is wrong. The node has been highlightened in the 2-D projection view (Fig.4 b). However, we cannot find out the reason for misclassification from this view for it is close to most phishing nodes. Though it locates in the periphery of the phishing nodes' aggregation area, this is not sufficient to account for its misclassification. Therefore, we turn to feature distribution view (Fig.4 c). We can discover that the phishing nodes and normal nodes appear strong differences in $inValue$, $outValue$, $aveInValue$, $aveOutValue$ and $frequency$. They also partly distinguish in $asFromTransCount$ and $asToTransCount$, but not very obviously. The selected node is demonstrated as a purple line in the feature distribution view. We can tell that this node is different from most of the other phishing nodes in the high-differentiation feature $inValue$, and shows a larger bias from other phishing nodes in $asFromTransCount$ and $asToTransCount$. From these aspects, it is reasonable for the misclassification induced by derived variables.



received 39.76582248072839

Balance 0.9558224807283864

Transaction Times 12

Figure 5: The second-order transaction network of a selected node. It exhibits low complexity.

To explore the nodes misclassified by two feature extraction methods, we brush the interval from August to September in 2018 and observe such a node. From the classification view, the features generated by GNN and GNNWUF lead to wrong predictions. We click on it and the second-order transaction network pops up. Unlike other phishing nodes, its second-order transaction network has a low complexity (Fig.5). From the 2-D projection view, this node is distant from other phishing nodes in GNN's and GNNWUF's feature space, but much closer in terms of derived variables. It is considered that GNN's and GNNWUF's sensitivity in network structures lead to their misclassification. For derived variables, despite its right prediction, this node is at the periphery of phishing nodes' cluster. Observing the purple line in feature distribution view, this may result from the heavy deviation in its $frequency$.

When it comes to the nodes misclassified by all three sets of features, we select the non-phishing node 0x93e4599b1ab3a336eb23f21689c0adc6c957f31a from March 2018 to April 2018. This node's local transaction network appears over-complex, where there are a large number of phishing and normal nodes. Conceivably, due to the high complexity of its second-order network, GNN and GNNWUF cannot properly extract the structure features, which causes its deviation from normal nodes' cluster in the 2-D projection view. For derived variables, its $aveInValue$,$outValue$, $aveOutValue$ and $frequency$ significantly deviate from other normal nodes' distribution. Therefore, all three feature extraction methods cause prediction results.

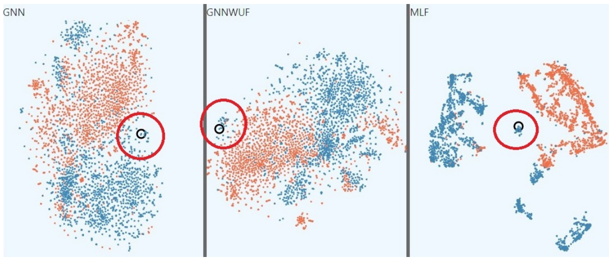


Figure 6: The 2-D projection view, consisting all nodes from March to April in 2018. The circled node is wrongly classified with all three feature extraction methods.

7 DISCUSSION

Traditional GNN method uses topological features of the transaction network to generate vector embeddings. However, due to the complexity of graph neural networks, these embeddings tend to be indecipherable. So it is difficult to understand what topological features they reflect and how they are generated. Our system incorporates visualization methods and provides the possibility to explore their topological properties, such as belonging to a specific transaction group or being the central nodes of a community. Besides, our work has the potential to be generalized for broader applications and fields. Although our method are based on financial transaction data in this paper, it can be adapted to other areas where similar dynamic networks exist.

In the process of exploration, however, we also discover some limitations of our system.

(1) Incapability for dynamic anomaly detection. So far, our system has provided various approaches to interpreting characteristics of phishing nodes in a downloaded dataset. However, it is unable to dynamically identify phishing nodes from a large scale of online transaction data. Dynamic and adaptive methods need to be integrated for more effective phishing detection.

(2) Time consumption on large-scale networks. On system has shown support for the exploration of a real-world blockchain dataset. While the huge volume of data has consumed lots of time in data retrieving and plot rendering. Future work may include an effective processing method of large-scale graph data.

8 CONCLUSION

In this paper, we propose a system for exploring transaction networks by combining GNN, variable derivation, classification method and visualization. Nodes in the networks are represented as vectors by GNN, GNNWUF and derived variables. Then we use two machine learning approaches to classify the nodes. The ROC results show all three feature extraction methods have a good performance. For visualization, the vectors are mapped onto two-dimensional space using t-SNE. In our visualization system, this will form the 2-D projection view. We use a timeline to select a certain time span for rendering the transaction networks in the topological view and classification results in the classification view. The interaction between the classification view and the node transaction network view allows us to further explore the locally topological features of the nodes. The feature distribution view is created to explore the transaction feature of the nodes, which enables us to understand the distribution of node features and interpret classification results. We also present the node and transaction information of interest in the detailed information view, providing us the possibility to discover nodes with unique behavior patterns and perform fund flow traces. In the future, we will integrate other dynamic methods into our system for phishing detection exploration.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (NSFC No.62202105) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0103), General Program (No. 21ZR1403300), Sailing Program (No. 21YF1402900) and ZJLab.

REFERENCES

[1] [n. d.]. The Ethereum Blockchain Explorer. *etherscan* ([n. d.]). https://etherscan.io/

[2] Stefano Bistarelli and Francesco Santini. 2017. Go with the -Bitcoin- Flow, with Visual Analytics. *Proceedings of the 12th International Conference on Availability, Reliability and Security* (2017). https://doi.org/10.1145/3098954.3098972

[3] Katy Börner, Chaomei Chen, and Kevin W. Boyack. 2005. Visualizing knowledge domains. *Annual Review of Information Science and Technology* 37, 1 (2005), 179–255. https://doi.org/10.1002/aris.1440370106

[4] Liang Chen, Jiaying Peng, Yang Liu, Jintang Li, Fenfang Xie, and Zibin Zheng. 2020. Phishing Scams Detection in Ethereum Transaction Network. *ACM Trans. Internet Technol.* 21, 1, Article 10 (Dec. 2020), 16 pages. https://doi.org/10.1145/3398071

[5] M. Ghoniem, J.-D. Fekete, and P. Castagliola. [n. d.]. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. *IEEE Symposium on Information Visualization* ([n. d.]). https://doi.org/10.1109/infvis.2004.1

[6] Stefan Hachul and Michael Jünger. 2005. Drawing Large Graphs with a Potential-Field-Based Multilevel Algorithm. *Graph Drawing Lecture Notes in Computer Science* (2005), 285–295. https://doi.org/10.1007/978-3-540-31843-9_29

[7] J. Heer and D. Boyd. [n. d.]. Vizster: visualizing online social networks. *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* ([n. d.]). https://doi.org/10.1109/infvis.2005.1532126

[8] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. 2014. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLoS ONE* 9, 6 (2014). https://doi.org/10.1371/journal.pone.0098679

[9] Christoph Kinkeldey, Jean-Daniel Fekete, Tanja Blascheck, and Petra Isenberg. 2020. Visualizing and Analyzing Entity Activity on the Bitcoin Network. (March 2020). https://hal.inria.fr/hal-02499003 working paper or preprint.

[10] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 2020. A survey on the security of blockchain systems. *Future Generation Computer Systems* 107 (2020), 841–853. https://doi.org/10.1016/j.future.2017.08.020

[11] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. 2016. An analysis of the Bitcoin users graph: inferring unusual behaviours. *Studies in Computational Intelligence Complex Networks and Their Applications V* (2016), 749–760. https://doi.org/10.1007/978-3-319-50901-3_59

[12] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. 2017. Data-driven analysis of Bitcoin properties: exploiting the users graph. *International Journal of Data Science and Analytics* 6, 1 (2017), 63–80. https://doi.org/10.1007/s41060-017-0074-x

[13] Tomer Moscovich, Fanny Chevalier, Nathalie Henry, Emmanuel Pietriga, and Jean-Daniel Fekete. 2009. Topology-aware navigation in large networks. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09* (2009). https://doi.org/10.1145/1518701.1519056

[14] Michal Ostapowicz and Kamil Żbikowski. 2019. Detecting Fraudulent Accounts on Blockchain: A Supervised Approach. *arXiv e-prints*, Article arXiv:1908.07886 (Aug. 2019), arXiv:1908.07886 pages. arXiv:1908.07886 [cs.CR]

[15] Robert Pienta, Minsuk Kahng, Zhiyuan Lin, Jilles Vreeken, Partha Talukdar, James Abello, Ganesh Parameswaran, and Duen Horng Chau. 2017. FACETS: Adaptive Local Exploration of Large Graphs. *Proceedings of the 2017 SIAM International Conference on Data Mining* (2017), 597–605. https://doi.org/10.1137/1.9781611974973.67

[16] Blaž Podgorelec, Muhamed Turkanović, and Sašo Karakatič. 2019. A Machine Learning-Based Method for Automated Blockchain Transaction Signing Including Personalized Anomaly Detection. *Sensors* 20, 1 (2019), 147. https://doi.org/10.3390/s20010147

[17] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural*

networks 20, 1 (2008), 61–80.

[18] Wei Shao, Hang Li, Mengqi Chen, Chunfu Jia, Chunbo Liu, and Zhi Wang. 2018. Identifying bitcoin users using deep neural network. In *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 178–192.

[19] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. 2014. BitIodine: Extracting Intelligence from the Bitcoin Network. *Financial Cryptography and Data Security Lecture Notes in Computer Science* (2014), 457–468. https://doi.org/10.1007/978-3-662-45472-5_29

[20] Jiajing Wu, Qi Yuan, Dan Lin, Wei You, Weili Chen, Chuan Chen, and Zibin Zheng. 2020. Who Are the Phishers? Phishing Scam Detection on Ethereum via Network Embedding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2020), 1–11. https://doi.org/10.1109/tsmc.2020.3016821

[21] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park, and Kari Smolander. 2016. Where is current research on blockchain technology?—a systematic review. *PloS one* 11, 10 (2016), e0163477.

[22] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and H. Wang. 2018. Blockchain challenges and opportunities: a survey. *Int. J. Web Grid Serv.* 14 (2018), 352–375.