Fu et al. Cybersecurity (2024) 7:3
https://doi.org/10.1186/s42400-023-00194-5

Cybersecurity

RESEARCH Open Access

CT-GCN+: a high-performance cryptocurrency transaction graph convolutional model for phishing node classification

Bingxue Fu[1], Yixuan Wang[1] and Tao Feng[1*]

Abstract

Due to the anonymous and contract transfer nature of blockchain cryptocurrencies, they are susceptible to fraudulent incidents such as phishing. This poses a threat to the property security of users and hinders the healthy development of the entire blockchain community. While numerous studies have been conducted on identifying cryptocurrency phishing users, there is a lack of research that integrates class imbalance and transaction time characteristics. This paper introduces a novel graph neural network-based account identification model called CT-GCN+, which utilizes blockchain cryptocurrency phishing data. It incorporates an imbalanced data processing module for graphs to consider cryptocurrency transaction time. The model initially extracts time characteristics from the transaction graph using LSTM and Attention mechanisms. These time characteristics are then fused with underlying features, which are subsequently inputted into a combined SMOTE and GCN model for phishing user classification. Experimental results demonstrate that the CT-GCN+ model achieves a phishing user identification accuracy of 97.22% and a phishing user identification area under the curve of 96.67%. This paper presents a valuable approach to phishing detection research within the blockchain and cryptocurrency ecosystems.

Keywords Blockchain, Information security, Phishing detection, Imbalance data, Transaction graph

Introduction

Blockchain, as a crucial emerging technology, is decentralized, tamper-evident and traceable (Yu et al. 2021). Due to these characteristics, the large number of cryptocurrencies (e.g., Bitcoin, Ethercoin) through innovative incentives and smart contracts have laid the groundwork for a more vigorous development of blockchain technology.

Unlike the Bitcoin blockchain, the Ether blockchain provides users with a decentralized computing environment that is not limited to just transaction users (Lee et al. 2020). Specifically, Ethereum can support multiple programming languages, allowing for the design of various decentralized applications on Ethereum, thereby expanding its scope of use (Wang et al. 2019). These characteristics have gradually made Ethereum popular among investors, becoming the second largest cryptocurrency after Bitcoin (Xie et al. 2021; Han et al. 2020). Even though Ethereum's market value is lower than Bitcoin, its crime probability is higher. According to the currency distribution of virtual currency crime cases in 2022, the proportion of virtual currency on the Ethereum blockchain in blockchain crime is 28.6%, while the proportion of virtual currency crime on the Bitcoin blockchain is only 13.7%. At the same time, the loss amount of blockchain security incidents in 2022 is mainly on BSC (Binance Smart Chain) and ETH (Ethereum) (CHAINDIGG 2022). Therefore the research for blockchain in this paper will be on the Ethernet blockchain.

*Correspondence:
Tao Feng
vonpower@ynufe.edu.cn
[1] School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming, China

Phishing is the illegal act of acquiring real account information through the use of seemingly legitimate entities. It mainly involves account names, passwords, and financial accounts (Ramzan 2010; Tan et al. 2020). In APWG's *Phishing Activity Trends Report* for the third quarter of 2022, 1,270,883 phishing attacks were recorded during the quarter. Attacks targeting the financial sector accounting for the highest percentage of all phishing attacks at 23.2% (Report 2022). Phishing in the traditional financial industry is the behavior of luring users to provide personal sensitive information or complete illegal transactions through false electronic means of communication, and its means mainly include fake identity, social engineering means and link luring, etc. (Alkhalil et al. 2021). In blockchain, the definition of phishing is not different, but it is still different from the traditional financial industry. It is mainly manifested in two aspects: firstly, the target is different, the phishing in blockchain is mainly for blockchain users and related services, which mainly includes fake wallets, fake ICOs and transaction fraud (Chen et al. 2022); secondly, blockchain has its own characteristics. Blockchain, for example, does not require third parties to perform transactions, and the information is completely open and transparent except for the alliance chain and private chain; the transaction medium is cryptocurrency rather than legal tender; cryptos cannot be used directly in the existing financial market and have to be exchanged for legal tender (Chen et al. 2020). It is because of the above differences that lead to the following challenges in blockchain phishing compared to phishing in the traditional financial industry: (1) The decentralization and anonymity of blockchain lead to the malicious nodes in phishing attacks being more hidden and difficult to track. One of the difficulties in categorizing blockchain phishing nodes compared to the traditional financial industry lies in the inability to accurately identify anonymous participants involved in an attack. (2) Blockchain technology ensures the trustworthiness and security of transactions through consensus algorithms and verification mechanisms. However, these mechanisms also provide new ways for phishing attacks. Attackers can deceive users and lure them to phishing websites through malicious behavior, such as forging transactions or exploiting smart contract vulnerabilities. (3) Transactions in blockchain networks are usually more complex and contain more data and interaction information. Moreover, users in blockchain networks are more inclined to participate and explore new technologies and projects, e.g., attackers can utilize social engineering methods to carry out phishing attacks by tricking users into providing private keys or visiting specific websites. Therefore, classifying blockchain network phishing nodes requires considering details and features in the transactions, which increases the complexity of classification. According to KNOWNSEC Blockchain Lab's (2022) hacked incident archive, the number of security incidents increased by about 37.3% in 2022 compared to 2021. Among them, attackers sent malicious tokens, resulting in losses of up to $8.1 million. As a result, it is urgent that researchers identify phishing attackers in blockchain cryptocurrency transactions.

We believe that transactions in blockchain have their own characteristics: firstly, the transactions between users have time sequence, so the entire transaction can be viewed as a time sequence with time information; secondly, blockchain is based on the assumption that "most of the people in a group are always honest", so the number of ordinary users is higher than the number of phishing scammers. Therefore, there is a serious data imbalance between phishing users and ordinary users. In order to better identify phishing users, in addition to utilizing the original features of the transaction (e.g., transaction amount, transaction cost, etc.) and the graph feature information of the transaction graph, we can also utilize the temporal features of the transaction. At the same time, in order to solve the data imbalance, we make the transaction graph from unbalanced to balanced by processing it.

Based on the existing research, we combine the above two ideas and propose our model: firstly, we take the information in the transaction records themselves as the basic features, such as the transaction amount, time interval, and the number of transactions, etc.; secondly, we extract the transaction records between the nodes that contain time information as edge features by LSTM method, and then we use the attention mechanism to convert the edge features into the graph node features, which completes the conversion of the transaction graph containing edge features as well as directions into an undirected graph containing only node features; finally, considering that the transaction graph contains certain nonlinear features and the transaction graph is an unbalanced graph, we combine the solution class unbalance method with graph neural network to extract as much information as possible so as to improve the accuracy of recognizing phishing users. Finally, the effectiveness of our model is proved by evaluating and comparing with different models.

The following are our main contributions:

(1) Pay attention to the timing of Ethereum cryptocurrency transactions. Time information has a certain role in Ether trading, through which we can not only determine the transaction time, block generation time and transaction speed, but also analyze the user's transaction behavior, so as to better

identify phishing users. Therefore, we extract time features from transactions using manual, moving average, and LSTM (Long Short Term Memory) methods, providing better feature support for accurately identifying phishing users.

(2) Apply the method of overcoming class imbalance in graph data to identify Ethereum phishing users. Since the transaction graph is unbalanced, meanwhile traditional methods for solving class imbalance are not applicable to graph data. In order to better extract the information from the transaction graph, we apply the method of solving class imbalance to the transaction graph, which not only overcomes the effect of ignoring the graph structure on node sampling to a certain extent, but also extends the method of phishing detection in the blockchain, and provides a reference significance for the application of graph convolutional technique combined with class imbalance technique in the field of network security.

(3) On the basis of the existing research on the time characteristics of Ethereum, we propose our cryptocurrency phishing detection model by combining different graph class imbalance processing methods. The model outperforms a variety of known same-task models, and provides a modellable idea for phishing detection in the blockchain ecosystem.

Our remaining parts are arranged as follows: "Related works" section presents the research work that is relevant to us; In the "Method" section, the problem definition is introduced in detail, and the cryptocurrency phishing detection model proposed by us is described in detail; In the "Experimental design" section, the experimental setup was introduced; The experiment is carried out and discussed in the "Experimental results and analysis" section to prove that the method has excellent performance in detecting phishing fraud of Ethereum cryptocurrency; Finally, a summary of the entire article is provided in "Conclusion" section.

Related works

Because phishing has existed for a long time (Rekouche 2011) and caused a large amount of losses, many researchers have conducted in-depth research on it. When studying traditional phishing detection problems, most researchers extract phishing fraud features from phishing websites and phishing pages. There are mainly three types of proven detection methods, such as black and white lists, classical machine learning, and deep learning (Feng et al. 2020). Among them, the black and white list method refers to the comparison of URLs with the normal URLs and phishing URLs known in advance, so as to distinguish the URLs (Bahnsen et al. 2017; Whittaker et al. 2010; Zhang et al. 2008). Although this method is stable and reliable, it usually does not cover all phishing websites. The study by Sheng et al. (2009) shows that approximately 50–80% of phishing domains are added to the blacklist after performing some financial losses. Traditional machine learning methods are used to identify phishing users by extracting features from training data and then feeding the extracted features into a machine learning model (Ma et al. 2009; Verma and Dyer 2015). Jain and Gupta (2018) proposed a machine learning anti phishing model based on URL features (PHISH-SAFE). On 33,000 phishing and legal URLs, SVM classifier was used to detect more than 90% of the ACC of phishing websites. Compared to traditional machine learning methods, deep learning methods have the advantage of avoiding tedious feature engineering. The main achievements of deep models in anti phishing research include: Bahnsen et al. (2017) treated each URL as an input sequence, and then classified and predicted it using the LSTM model. The final results showed that the ACC of this method was 98.7%. Yuan et al. (2020) proposed an improved bi-directional GRU model based on attention mechanism for phishing website URL detection (BiGRU Attention model), which not only recognizes phishing websites but also has interpretability, and the model's ACC is as high as 99.55%.

Due to the fact that blockchain phishing primarily targets transactions on the blockchain, the study of blockchain phishing will no longer begin with websites and URLs. In the research on blockchain phishing users, there are mainly two aspects: manual feature extraction and automatic feature extraction. Firstly, research on manually extracting features mainly includes Farrugia et al. (2020) obtaining 42 transaction user features through the Ethereum API, and then using the XGBoost model to classify and predict 4681 transaction users, resulting in a classification ACC of 96%. In order to detect potential Ponzi scheme in smart contracts, Chen et al. (2018) also estimated more than 400 Ponzi scheme in Ethereum by combining manually extracted features and XGBoost classifier. Although manually extracted features have a good classification effect, due to the intricate nature of blockchain cryptocurrency transaction networks, the original extracted features will inevitably miss important details. Moreover, due to uncontrollable factors such as time and malicious attacks, the original features are weak in representing and distinguishing legal and illegal transactions, so more efficient learning methods are needed (Zhu et al. 2021). Therefore, deep learning models are gradually favored by researchers because they require less human intervention. Yuan et al. (2020) used Deepwalk and Node2vec to automatically extract

features from Ethereum transaction graphs, and used one class support vector machine (one class SVM) for transaction user classification. The final experimental results showed that the Node2vec algorithm was superior to Deepwalk. Lin et al. (2019) proposed the T-EDGE model considering the time and quota factors of transactions, and experimental results showed that this method outperforms Deepwalk and Node2vec algorithms. In addition to using Graph Embedding to automatically extract features, many researchers also use graph convolutional neural networks to identify Ethereum phishing users. Wang et al. (2021) compared the three manual features extraction methods, Graph2Vec, Diffpool, for the transaction graph (TSGN) formed by Ethereum traders and their first-order neighbors. They found that the Diffpool method performed best. In the previous study of this paper (Bingxue et al. 2022), we proposed the CT-GCN model, which considered transaction direction and features at the edges of the transaction graph, and its classification accuracy was 88%. Although the CT-GCN model considers the impact of transaction direction on identifying phishing users of the network, it is a graph classification model that enables fraud detection in cryptocurrency transaction networks, i.e., determining whether or not a set of transactions contains phishing fraud.

Compared with the above studies on blockchain network phishing, our proposed CT-GCN+model has its own advantages and features. First, in terms of feature extraction, the CT-GCN+model not only adopts the manual feature extraction method to extract the basic features, but also adopts LSTM and graph neural network to automatically extract the features in the transaction graph; second, in terms of graph processing, the Ethereum transaction graph is a directed, multilateral, and imbalanced graph with time characteristics on its edges, while traditional graph neural networks and methods for solving class imbalances cannot be directly used. Therefore, we convert the polygonal directed transaction graph into a one-sided undirected graph by extracting temporal features, and then perform class imbalance processing on the processed graph. It is worth noting that there are no reports on class imbalance processing on Ethernet transaction graphs.

In summary, we summarize the results of the research on blockchain phishing compared with our proposed CT-GCN+model in Table 1.

Method

Construction of ethereum cryptocurrency transaction graph

To identify phishing users in Ethereum, we construct an Ethereum transaction graph G on the transaction records of Ethereum users. Since the transaction record contains many transaction information, such as user balance, transaction amount, transaction time, transaction cost and transaction direction, we treat the transaction record as an edge in the transaction graph, the transaction information as information on the edge, and the Ether users involved in the transaction record as nodes in the transaction graph (as shown in Fig. 1).

CT-GCN+model

Our model is mainly divided into a temporal information part for extracting temporal features and a graph balance processing part for solving class imbalance. Next, we will provide a detailed introduction to our model (Fig. 2).

Time information section

In our model, the temporal characteristics part is divided into two main parts, Timing Processing and Edge Processing.

Table 1 Blockchain phishing research comparison table

Categorization	Methodologies	Characteristics
Manual	After extracting features manually, classification is performed using a machine learning classification model	Advantages: 1. Can result in better categorization Disadvantages: 1. Can lead to loss of important information 2. Requires some experience
Automatic	After learning the features using the deep learning model, the features are fed into the classification model for classification	Advantages: 1. No need to extract features manually, reducing the interference of human factors 2. Doesn't require much experience Disadvantages: 1. Classification effect is lower than manual extraction of features
Manual + automatic (CT-GCN+)	After extracting the features manually, they are fed into the GCN to learn the features automatically, and the graph balancing process is considered in the GCN	Combining manual and automatic feature extraction methods, while considering graph balance processing, greatly improving classification efficiency

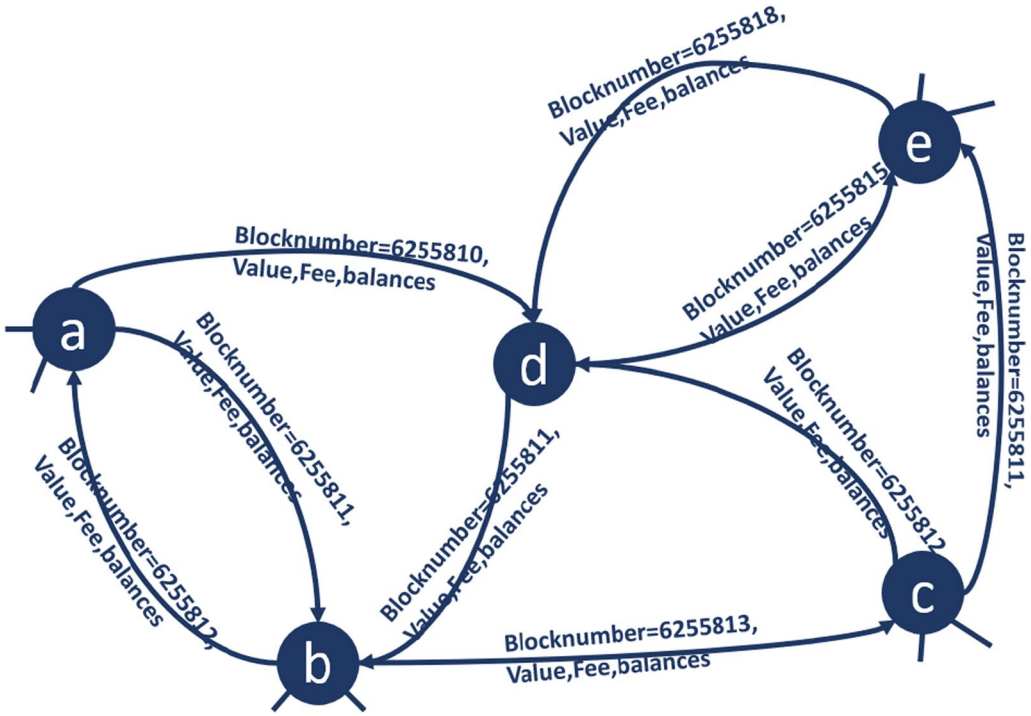

Fig. 1 Ethernet transaction graph

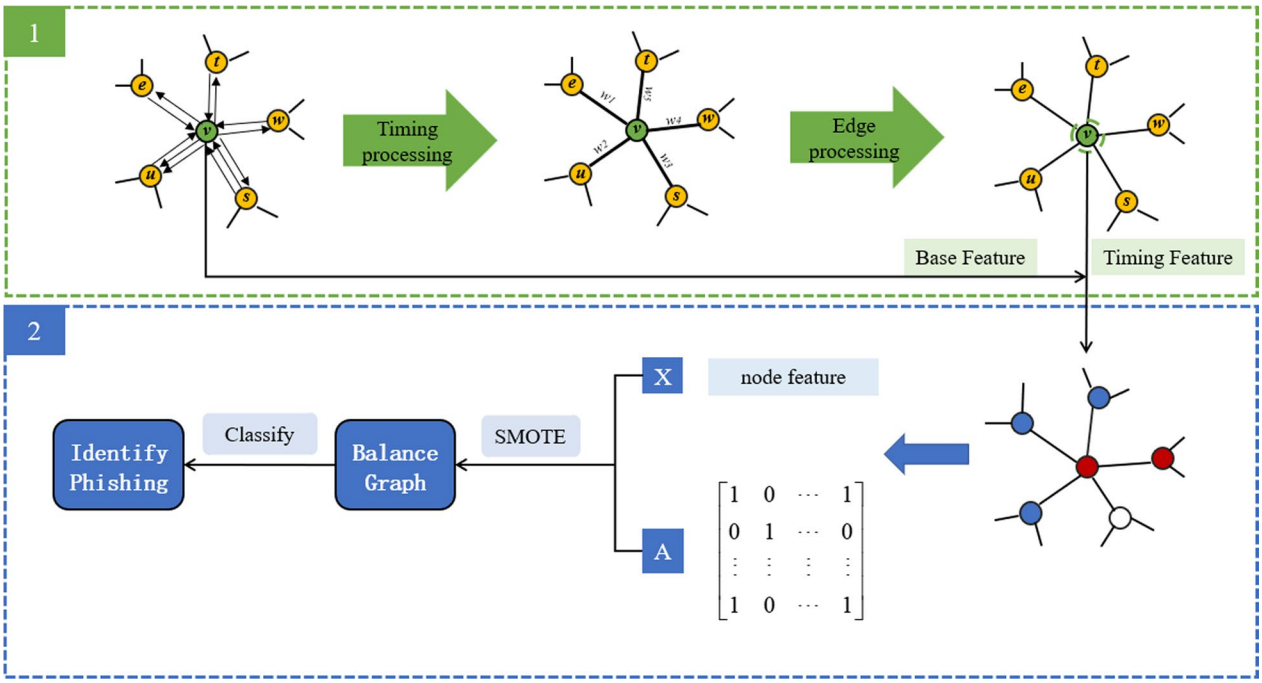

Fig. 2 CT-GCN+ model architecture diagram

(1) Basic Features (BaF)

In the Ethernet transaction graph G, each node (transaction account) itself does not have any characteristics, because all characteristic information exists on the transaction record. Therefore, in order to better express node information and conduct comparative experiments, we chose 9 features as the basic features of the node, including the degree of the node (egress, ingress, and total degree), the number of neighbors (egress, ingress, and total number of neighbors), the ratio of total transaction volume to the number of neighbors, the proportion of neighbors with all transaction amounts of 0, and the maximum value in total transaction volume (see Table 2 for details).

(2) Timing Processing

In this section, we mainly refer to Li et al.'s (2022) treatment of time when targeting the temporal transaction aggregation graph network for Ether phishing scam detection. It is mainly divided into three parts: (1) Since transactions between accounts in Ethereum are carried out in chronological order, the transaction records between each pair of nodes (transaction accounts) are treated as a time series based on the transaction time; (2) Considering that the direction of the transaction and the amount of the transaction are crucial in the whole transaction, the amount of the transaction is assigned according to the direction of the transaction, i.e. positive for sending and negative for receiving; (3) Using the LSTM model to extract features from the transaction time series to obtain the edge features between each pair of nodes. (The schematic diagram is as follows).

In Fig. 3, v_1, v_2, L, v_n and t_1, t_2, L, t_n denote the transaction amount and transaction time between node v and node u. h_1, h_2, L, h_n denote the results of (v_1, t_1), $(-v_2, t_2)$, L, $(-v_n, t_n)$ after LSTM. We finally choose hn as the edge feature between node v and node u, which is $\tilde{e}_{vu}$.

(3) Edge Processing

Through Timing Processing, the polygons in transaction graph G are transformed into single edges, where each pair of nodes is connected by only one undirected edge with information. In order to integrate the information on the undirected edges around the nodes into the central node (i.e., processing the edge information), an Attention model was introduced to ultimately obtain the temporal characteristics (TF) of the nodes (Fig. 4).

Table 2 Base features of nodes

Feature representation	Feature meaning
degree_in/to/total_sum	Incoming/outgoing/total transaction volume
neighbor_in/to/total	Incoming/outgoing/total neighbors
Inverse_frequency	Ratio of total number of transactions to number of neighbors
Amount_0_percentage	Percentage of neighbors with all 0 transactions
degree_total_max	The maximum value of the total transaction volume

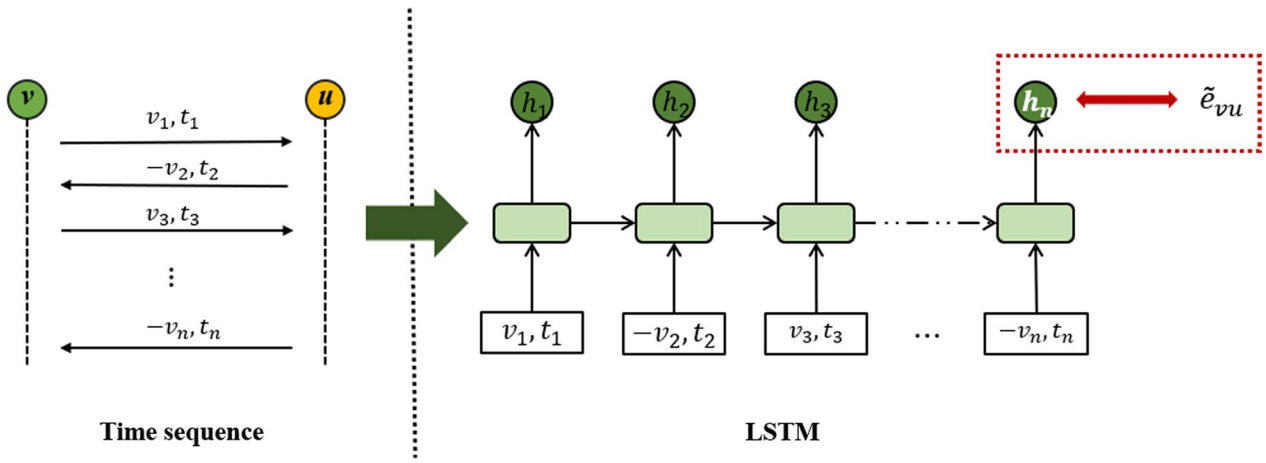

Fig. 3 Timing processing (this image is from reference Li et al. 2022)

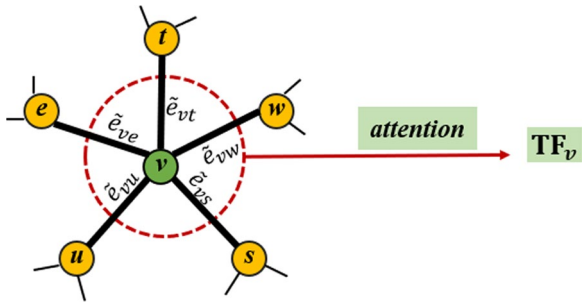

Fig. 4 Edge processing

$$\begin{aligned} TF_v &= concat\ (head_1, head_2, \cdots, head_h)W \\ head_i &= Attention\ (Q, K, V) \\ &= softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \end{aligned} \tag{1}$$

Among them, Q, K, and V respectively represent the matrix of transaction numbers between each pair of nodes, the feature matrix formed by the combination of node basic features and edge features (output of LSTM), and the edge only feature matrix; W represents the weight matrix of the multi head attention mechanism.

Figure balance processing section

The temporal features of the transaction graph nodes are obtained through the temporal information section; then the temporal features are combined with the node base features as the input to the graph balance processing section. To solve the problem of graph imbalance, we mainly adopted a combination of GCN and SMOTE methods. The detailed process of the balance processing section (Ando and Huang 2017) is as follows:

The Embed_SMOTE model first learns two graph convolution layers (GCL) for the imbalanced graph to obtain the graph embedding features (embedding) of each node:

$$\begin{aligned} H_1 &= \mathrm{ReLU}(\hat{A}XW_1) \\ H_2 &= \mathrm{ReLU}(\hat{A}H_1W_2) \end{aligned} \tag{2}$$

where The X matrix denotes the identity matrix of the graph nodes, A denotes the adjacency matrix of the graph, and $\hat{A} = \hat{D}^{-\frac{1}{2}}(A+I)\hat{D}^{-\frac{1}{2}}$, $\hat{D}$ is diagonal matrix and $\hat{D} = I + \sum_j A_{ij}$, I represents identity matrix; W_1, W_2 is the weight parameter in GCL (Fig. 5).

Then perform SMOTE sampling balance on node features, but without any changes to the edges, and finally the balanced node features are fed into the linear layer (FC) for classification.

$$\begin{aligned} X' &= \mathrm{SMOTE}(H_2) \\ Z &= \mathrm{ReLU}(\hat{A}X'W_3) \\ \hat{y} &= \mathrm{FC}(Z) \end{aligned} \tag{3}$$

Although our model was inspired by Li et al. (2020), it still differs from it. The main differences are as follows: (1) The statistical characteristics of transaction graph nodes are different, and we use basic features (BaF) instead; (2) The output of Graph Convolutional Layer (GCL) is different, that is, GCL is no longer considered as a structural feature extractor to output features, but instead adopts an end-to-end approach, treating it as a combination of feature extractors and classifiers to directly complete the classification output and classification results; (3) Cancel the introduction of LightGBM classifier because the classification has been completed by introducing a linear layer; (4) Due to the serious class imbalance in the transaction graph we constructed, in order to solve this problem, we improved the graph neural network GCN by introducing the SMOTE method (Zhao et al. 2021; Ando and Huang 2017).

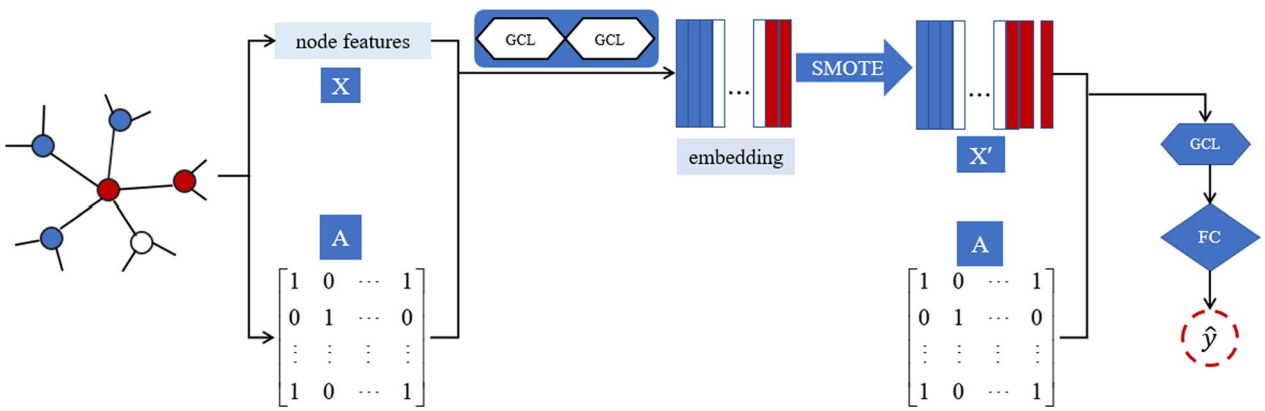

Fig. 5 Embedded_SMOTE flowchart

Experimental design

Experiment on the relevant features of the graph

Time characteristics

The transaction records of Ethereum mainly include five pieces of information: transaction time, sender's balance, receiver's balance, transaction amount, and transaction cost. Except for the transaction time series, the remaining four series are time series. In order to better mine the information of time series, we extracted the features of time series using three methods: manual extraction, moving average method, and LSTM.

(1) Manual extraction method

We count the range and standard deviation of the transaction time in three ways: out, in and total; and the sum, maximum, minimum, mean and standard deviation of the four information other than the transaction time in three ways: out, in and total, and the features consisting of the four information such as transaction amount are collectively called manual time series features, abbreviated as TmF (Specific feature information is shown in Table 3).

(2) Moving average method

Due to the consistent use of the moving average method for all four time series, we chose to use the transaction amount as an example to explain the moving average method. By referring to Li et al.'s (2020) treatment of the time series of bitcoin address balances, we do the following for the time series V of Ethereum user transaction amounts: First, the first-order difference of the time series V is done to obtain the differenced series V1, and calculating the mean and standard deviation of the V1 series; then, a 2/4/6/8-period moving average of the V1 series is calculated; finally, the mean and standard deviation of the V1 series and all moving averages are used as the eigenvalues of the V series, and the TsF features are obtained by this method.

(3) LSTM method

The LSTM network is specifically designed to solve long-standing problems, with the main idea being to address each point in time T_n will have a corresponding state C_t, and C_t it is not only related to the state of the current time point, but also to the state of the past time point. Therefore, for each time point T_n it can be corrected by adjusting the input of weights, forgetting, and other methods to revise C_t.

Therefore, we use the LSTM method to automatically extract time series features for each of the four time series, and the features obtained by this method are named TaF. It is worth noting that the time series using the LSTM method is consistent with the time series using the moving average method in (2).

Graph embedding

Graph Embedding technology represents nodes in a graph in the form of low dimensional dense vectors. It requires similar nodes in the original graph to be close to each other in the low dimensional representation space, and the resulting expression vector can be used for downstream tasks such as node classification, link prediction, visualization, or reconstruction of the original

Table 3 Manual time series features (TmF)

Feature representation	Feature meaning
Time_in_range/std	Time span/standard deviation of entry
Time_to_range/std	Time span/standard deviation of out
Time_range/std	Total time span/standard deviation
value_in_sum/max/min/mean/std	Sum/maximum/minimum/mean/standard deviation of entered amount
value_to_sum/max/min/mean/std	Sum/maximum/minimum/mean/standard deviation of outgoing amount
value_sum/max/min/mean/std	Total amount/maximum/minimum/mean/standard deviation
fee_in_sum/max/min/mean/std	Sum/maximum/minimum/mean/standard deviation of entered expenses
fee_to_sum/max/min/mean/std	Sum/maximum/minimum/mean/standard deviation of outgoing expenses
fee_sum/max/min/mean/std	Total expenses sum/maximum/minimum/mean/standard deviation
sender/receiver_balances_in _sum/max/min/mean/std	Incoming sender/receiver balance Sum/Max/Min/Mean/Standard Deviation
sender/receiver_balance_to _sum/max/min/mean/std	Outgoing sender/receiver balance Sum/Max/Min/Mean/Standard Deviation
sender/receiver_balances _sum/max/min/mean/std	Total sender/receiver balance Sum/Max/Min/Mean/Standard Deviation

graph. Since the Graph Embedding method extracts features from the Ethereum transaction graph G from the perspective of graph structure, we will use these features as structural features (StF). In this section, the main methods involved are T-EDGE and Node2vec.

(1) T-EDGE

Due to the fact that the Ethereum transaction graph is a multi-sided directed sequence graph, that is, it has multiple edges between each pair of nodes and each edge has a direction and time, Lin et al. (2019) proposed a graph embedding method based on the three factors of transaction amount, time, and direction: T-EDGE method. The main content of this method means: first the edges of each node (i.e. transaction records) are arranged in chronological order, and each edge itself carries a weight (i.e. transaction amount); Then calculate the node sampling probability based on time deviation sampling/amount deviation sampling/time amount average deviation sampling; Finally, the sampling sequence is obtained based on the node sampling probability and edge direction, and the sampling sequence is calculated to achieve the node structure characteristics. The sampling probability calculation methods for time deviation sampling, amount deviation sampling, and time amount average deviation sampling are as follows:

$$\begin{aligned} P_{TBS}(e) &= \frac{\eta(T(e))}{\sum_{e\prime \in N_t(v_i)} \eta(T(e\prime))} \\ P_{WBS}(e) &= \frac{\eta(W(e))}{\sum_{e\prime \in N_t(v_i)} \eta(W(e\prime))} \\ P_{TBS+WBS}(e) &= \frac{P_{TBS}(e)^{\alpha} P_{WBS}(e)^{1-\alpha}}{\sum_{e\prime \in N_t(v_i)} [P_{TBS}(e)^{\alpha} P_{WBS}(e)^{1-\alpha}]}, (0 \le \alpha \le 1) \end{aligned} \tag{4}$$

where $T(e)$ denotes the transaction time on edge e, $W(e)$ denotes the amount of the transaction on edge e, α denotes the degree of sampling according to the time amount deviation, as a hyperparameter; In addition, in order to follow the rule that the closer the transaction time (large), the greater the degree of association between nodes and the greater the transaction amount the greater the degree of association between nodes, we set:

$$\eta = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{5}$$

(2) Node2vec

Because the deviation sampling of Node2vec is mainly divided into depth first walk and breadth first walk based on edge weights, and this method is based on the fact that there is one and only one edge in each direction between each pair of nodes in the graph, so the multilaterally directed Ethernet transaction graph needs to be pre-processed before using the method, i.e., the multilateral is changed into a single edge.

Class imbalance experiments on the graph

In addition to the SMOTE method of the features after the graph convolution that we have used, there are also methods to change the graph into a balanced graph before performing the graph convolution and to perform SMOTE on features and edges simultaneously. The detailed process is as follows:

(1) SMOTE

Before learning the graph neural network (GCN) for imbalanced graphs, The graph nodes are sampled in accordance with SMOTE to generate the corresponding nodes n', at this point the number of fishing nodes in the graph nodes is approximately the same as that of normal fishing nodes; For edges, they are directly copied according to the original imbalanced graph, i.e. the size of A_balance is $(n' + n) \times (n + n')$; Finally, a relatively balanced graph is obtained, and then double-layer GCL training is performed on the balanced graph to achieve the goal of identifying phishing users (Fig. 6).

(2) Graph_ SMOTE

The Graph_SMOTE model is based on Embed_ SMOTE, a new adjacency matrix is obtained by decoding the generated high-dimensional embedding. Then the nodes and edges are trained simultaneously to make both nodes and edges balanced and more reasonable (Fig. 7).

Experimental data

We crawled the transaction information of phishing users and normal users from the Ethernet (China) website by means of web crawlers. The starting block for the phishing account is 1,997,275 (August 2, 2016), while the block for which we crawled the data on July 25, 2021 is 12,892,110, and there are more than a billion transaction records between these two blocks. Due to our lack of equipment and the limitations of the website for crawlers, as well as the representativeness of the data crawled, we chose to use a transaction network consisting of the crawling hub node and its first-order neighbors to represent the transaction records between block 1,997,275 and block 12,892,110. In order to make our transaction network cover all phishing users as much as possible, we

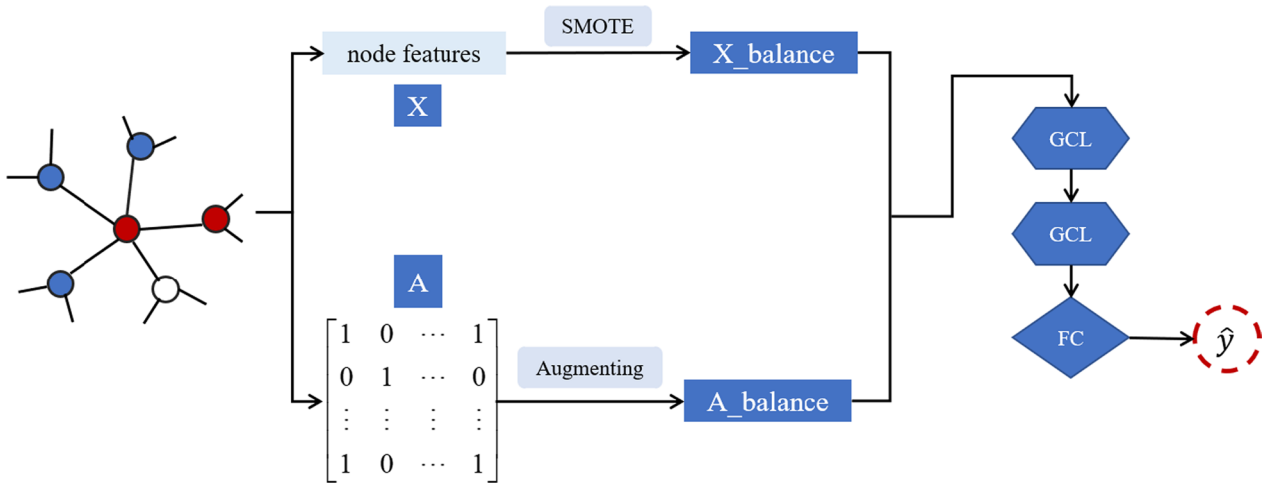

Fig. 6 SMOTE flowchart

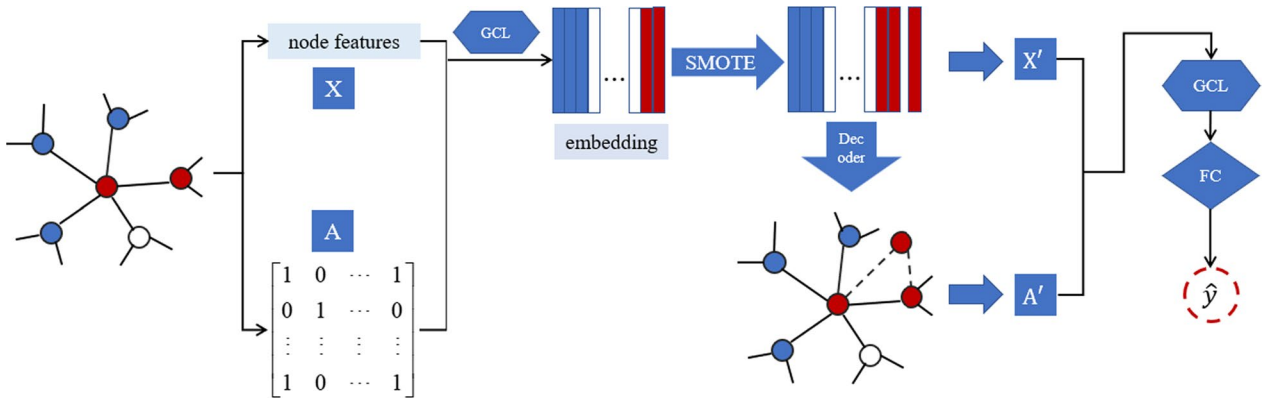

Fig. 7 Graph_SMOTE flowchart

choose all phishing users and their corresponding number of normal users as the central node, and then crawl the transaction information of their first-order neighbors and their first-order neighbors, which ultimately constitutes our experimental data.

We crawled a total of 220,000 transaction records consisting of 3879 phishing users as the central node, as well as 230,000 transaction records consisting of 3874 normal users as the central node. After cleaning the above crawler data (Jiajing et al. 2022; Zheng et al. 2022), we ultimately obtained 336,500 transaction records, and our transaction graph G is built on it. In transaction graph G, there are a total of 93,007 Ethereum users, while the number of phishing users is only 1928.

Experimental evaluation indicators

ACC (Accuracy) represents the percentage of correctly predicted sample sizes in the total, and is often used as one of the indicators to evaluate the quality of classification. However, considering that our dataset is a class imbalanced dataset, relying solely on ACC will result in certain errors. Therefore, we introduce both F1 Score and AUC as evaluation indicators. The essence of F1 scoring is not to miss any chance of making mistakes; And AUC, as the area of the ROC curve, essentially reduces the chance of model errors and is more moderate compared to F1 scores. Therefore, we ultimately used ACC, F1 score, and AUC as our experimental evaluation indicators.

$$\begin{aligned} \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\ \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \\ F_1\text{ - score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{6}$$

where TP is the number of positive samples correctly predicted; FP represents the number of positive samples that were incorrectly predicted; FN represents the number of negative samples that were incorrectly predicted; TN is the number of correctly predicted negative samples.

Experimental setup

Time series processing in time features

When extracting time features, we consider the transaction records of nodes and all their first-order neighbors as a time series, so each node has a corresponding time series composed of transaction records. However, in actual Ethereum trading networks, the transaction volume of different nodes is inconsistent, which can be confirmed from the data we have collected. At the same time, due to the fact that Ethereum users have transactions in and out, we mark the sent transactions as positive (+) and the received transactions as negative (−) when constructing the time series, in order to distinguish the direction of transactions.

According to Table 4, it can be seen that 95.45% of the 93,007 nodes in the transaction graph G have a transaction volume less than or equal to 20. Therefore, in order to make the time series relatively consistent and ensure more accurate experimental results, we have unified the length of the time series to 20. For time series with a length less than 20, the insufficient part is supplemented with − 1; For time series with a length longer than 20, select the 20 transaction data that later in time (i.e. the highest BlockNumber value).

Table 4 Node time series length

Time series length	Percentage (%)	Cumulative percentage (%)
1	71.88	–
2	12.97	84.85
3	3.28	88.13
4	2.04	90.17
5	0.96	91.13
6	0.73	91.86
7	0.47	92.32
8	0.41	92.73
9	0.27	93.00
10	0.26	93.26
11	0.32	93.59
12	0.30	93.89
13	0.25	94.14
14	0.26	94.40
15	0.22	94.61
16	0.19	94.80
17	0.17	94.97
18	0.18	95.15
19	0.16	95.30
20	0.14	95.45

Graph embedding

The experimental parameters for the graph embedding model are set as follows: the embedding feature dimension is 128; The width of the Skip_gram is 10; The walk length is 10, and each node can be traversed up to 10 times.

CT-GCN+

The transaction graph G we constructed contains 93,007 nodes, which are limited by computer hardware facilities. We chose to randomly select 10,000 and 20,000 nodes from G to form a transaction subgraph and train them on the transaction subgraph (Table 5).

Table 5 Transaction subgraph

Total number of G1 nodes	Number of G1 fishing nodes	Total number of G2 nodes	Number of G2 phishing nodes
10,000	217	20,000	413
10,000	196	20,000	401
10,000	221	20,000	432
10,000	197	20,000	409
10,000	235	20,000	411

(1) LSTM

In this experiment, it is no longer the transaction records of nodes and all their first-order neighbors that form a time series, but rather the transaction records between each pair of nodes that form a time series. Therefore, the transaction amount is first assigned according to the direction of the transaction, i.e. send as positive and receive as negative; Then, based on the statistics in the transaction graph G, it can be seen that there are a total of 246,970 node pairs in G, namely 246,970 time series, and there are 243,916 time series with a length less than or equal to 25. Therefore, we unify the time series length to 25; Finally, the uniformly long time series is input into the LSTM model to obtain the edge features between each pair of nodes.

(2) Attention

By using the LSTM model, the polygons in transaction graph G are transformed into single edges, i.e., each pair of nodes is connected by only one undirected edge with information. In order to integrate the information

on the undirected edges around the nodes into the central node, the TTAGN model introduces the Attention model. It should be noted that at this stage, the input sequence is composed differently and the length of the input sequence is determined by the number of neighbors of the central node. According to statistics, 85% of nodes have neighbors less than or equal to 50, so we set the input sequence length of the Attention layer to 50. And as the three most important input matrices of the Attention model, Q, K, and V are composed of the number of transactions between each pair of nodes, the features formed by the combination of node basic features and edge features (output of LSTM), and edge features, respectively.

(3) Figure balance processing section

Obtain the time characteristics of nodes through LSTM and Attention; Then, the combination of time features and node basic features is used as input for the graph balance processing section. For some parameter settings, see Table 6.

Experimental results and analysis

Although the experiments we are involved in differ in data processing, their essence is to conduct multiple sampling and repeated experiments on the transaction graph G composed of the data we collect. Therefore, the experimental results are to some extent comparable. Except for the graph balance processing section that does not involve a specific classifier, all other experiments were conducted on XGBoost classifiers.

Experimental results

According to the experimental results in Table 7, after adding temporal features to the basic features, the classification performance has been improved to a certain extent, indicating that considering temporal factors can effectively improve the recognition ability of Ethereum phishing users. The bold numbers in the table represent the best results. Among these three different temporal feature processing methods, manual feature extraction is the most effective, with F1 scores, ACC, and AUC all increased by about 8%; Next is the moving average method, with an increase of around 5%; The last one is the LSTM method, with an improvement of only about 3%. However, overall, any method that combines manual extraction of temporal features has an F1 score, ACC, and AUC of around 89%.

To confirm the impact of the number of transactions between nodes on the identification of phishing users, we consider two methods in the Node2vec model, namely using the number of transactions in the same direction between each pair of nodes as the weight (Node2vec1) and having the same weight between each pair of nodes (Node2vec2). From the experimental results, it can be seen that the classification performance of Node2vec2 is not as good as Node2vec1, with a difference of about 3% between the two. Therefore, this problem has been confirmed. Compared to the Node2vec model, the T-EDGE model is slightly better than the Node2vec model, with F1 scores, ACC, and AUC exceeding 90%.

Since in the above experiments requiring XGBoost classification, it is by randomly selecting the nodes so that the experimental data are already class balanced at the time of input to the XGBoost model, i.e., at the time of constructing the data. But in the class imbalance experiment on the graph, the data is balanced by considering the SMOTE method at different stages. Based on all experimental results, the effect of class imbalance

Table 6 Parameter settings for the balance processing part of the figure

Parameters	Value
epoch	200
up_scale	45
im_ratio	0.02
batch_size	32
embedding_size	64
lr	0.001
dropout	0.01

Table 7 Experimental results

Experimental model	F1	ACC	AUC
BaF	0.8077	0.8172	0.8178
BaF + TmF	0.8963	0.8982	0.8985
BaF + TsF	0.8575	0.8603	0.8606
BaF + TaF	0.8288	0.8355	0.8360
BaF + TmF + TsF	0.8919	0.8943	0.8946
BaF + TmF + TaF	0.8901	0.8916	0.8919
BaF + TsF + TaF	0.8487	0.8525	0.8528
BaF + TmF + TsF + TaF	0.8883	0.8903	0.8906
BaF + TmF + Node2vec1	0.9497	0.9316	0.9113
BaF + TmF + Node2vec2	0.9295	0.9055	0.8908
BaF + TmF + T-EDGE	0.9498	0.9317	0.9154
Time information section + SMOTE	0.9503	0.9686	0.9485
Time information section + Graph_SMOTE	0.8422	0.9547	0.9445
CT-GCN+ (our model)	**0.9507**	**0.9722**	**0.9667**

treatment on the graph is better, with ACC and AUC scores exceeding 95%. In the class imbalance experiment on the graph, our model performs the best. The reason for this may be that in this method, the structure of the original transaction graph was not changed, thereby reducing the negative impact of graph structure information on classification results.

Ablation experiment

Since our model is mainly divided into two parts: time information and graph balance processing, in order to understand the magnitude of the role played by each part in the whole experiment, we did a model ablation experiment. The specific results are as follows: (where w/o_Embed represents the absence of graph balance processing part, and w/o_Time represents the absence of time information part).

According to the results of Fig. 8, In the entire Ethereum, both graph balance processing and time information have played a positive role in identifying phishing users, and the graph balance processing part has been more effective than the time information part.

In terms of the ACC results, the w/o_Embed model is better than the other two models. It is possible that this is because it does not perform class imbalance processing. This leads to a prediction result more likely to be based on more samples, which increases the ACC. Based on the F1 score, the results of the w/o_Embed model are very unstable, which highlights the importance of the balance processing part of the graph.

Parameter sensitivity experiment

The impact of transaction graph size

From the experimental results, it can be seen that in the transaction subgraph G1 composed of 10,000 nodes, different class imbalance processing methods have their own advantages. Among them, the ACC of the time information part + SMOTE model can reach 98%, while the AUC of the time information part + Graph_SMOTE model is 92.75%, which is the highest among the three methods. But in the transaction subgraph G2 composed of 20,000 nodes, our model method is the best, with F1 scores, ACC, and AUC all above 95%. In summary, our proposed model performs better and is more stable in node classification scenarios with 20,000 nodes than in node classification scenarios with 10,000 nodes (Table 8).

Parameter impact of time information section

In order to explore the degree of influence of various parameters in the time information section, we selected different LSTM dimensions and Attention sizes for experiments (Figs. 9 and 10).

The final results showed that although the results of different LSTM output dimensions showed greater differences compared to the results of different Attention sizes, the degree of influence of LSTM dimensions and Attention sizes decreased with the increase of the constructed transaction graph size, and the difference was not significant, This indicates that the temporal information part has good robustness in larger transaction graphs, and the classification results are more susceptible to different LSTM dimensions. However, overall, the LSTM

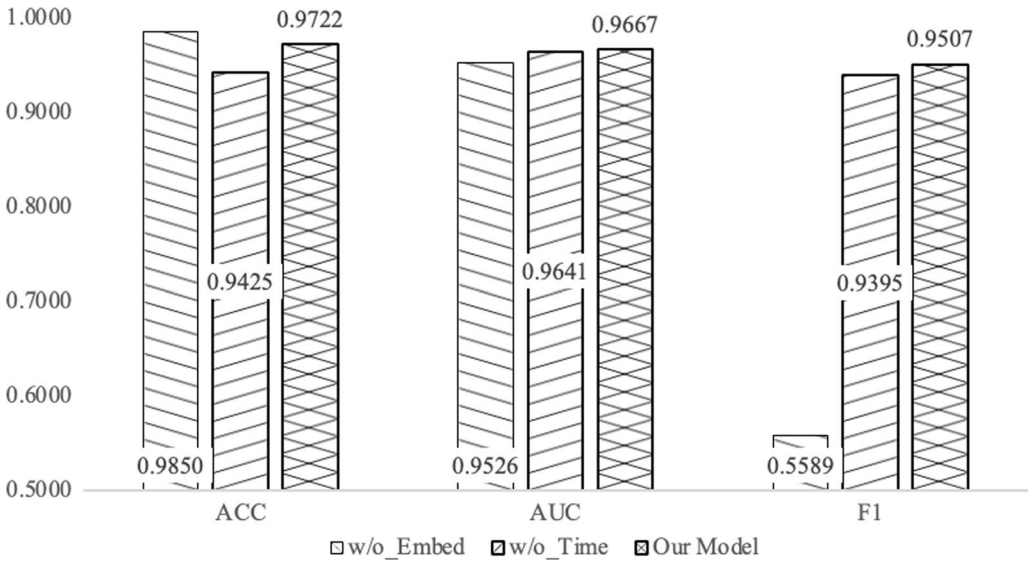

Fig. 8 Experimental results for each part of the model

Table 8 Experimental results of different transaction graph sizes

	G1			G2		
	F1	**ACC**	**AUC**	**F1**	**ACC**	**AUC**
Time information section + SMOTE	0.8871	**0.9814**	0.8889	0.9503	0.9686	0.9485
CT-GCN+ (our model)	**0.8932**	0.9584	0.9185	**0.9507**	**0.9722**	**0.9667**
Time information section + Graph_SMOTE	0.8907	0.9409	**0.9275**	0.8422	0.9547	0.9445

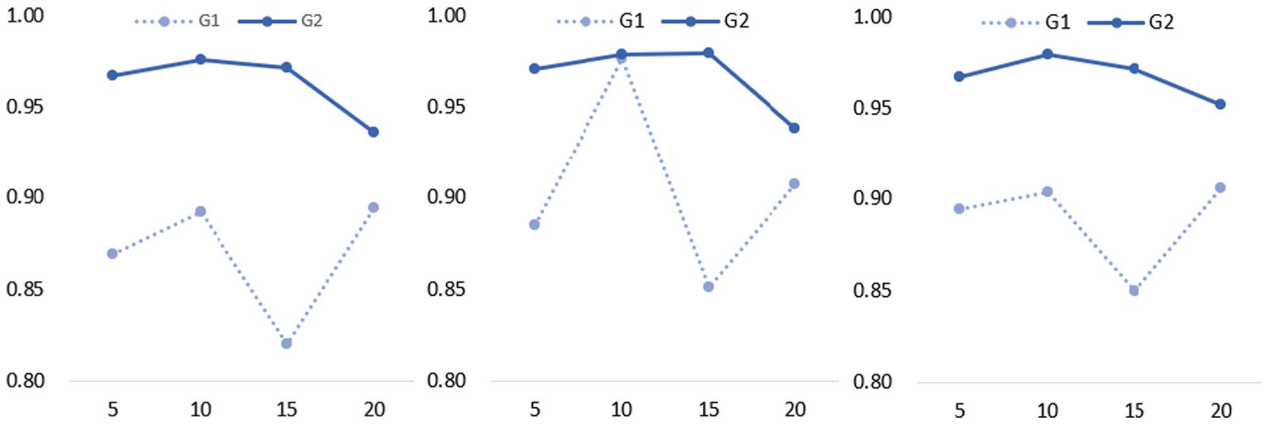

Fig. 9 The impact of different LSTM output dimensions (From left to right: F1, ACC, AUC)

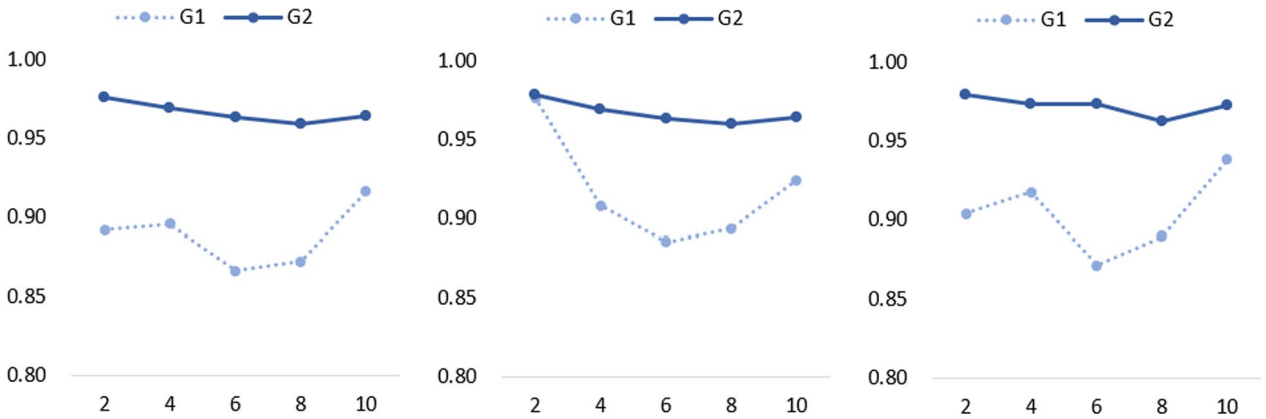

Fig. 10 The impact of different Attention sizes (From left to right: F1, ACC, AUC)

dimension of 10 and the Attention size of 10 have the best effect, with F1 scores, ACC, and AUC reaching 90%.

Conclusion

Due to the lack of third-party supervision of cryptocurrency transactions in blockchain, there is a series of online fraud and fraud behaviors in this emerging financial ecosystem, seriously threatening the security of the entire system. Therefore, in order to achieve more efficient identification of phishing users in Ethereum, we also consider the time and class imbalance problems in Ethereum phishing user identification, and propose the CT-GCN+ model. The final experimental results prove the importance of time and graph structure in the identification process of Ethereum phishing users. Meanwhile, compared to the model without considering graph balancing, our model has a 5% improvement in ACC and AUC. Moreover, compared to models that consider time and graph balance, our model remains the optimal model.

In the research process of this article, there are still some issues worth further analysis and discussion: (1) Since our data collection is to identify users first and then collect transaction information of users and their first-order neighbors, it leads to the defect that the transaction graph constructed at the time of classification has incomplete transaction collection between nodes. Therefore, in future work, if graph node classification is considered, the dataset should be constructed as much as possible by determining a certain time period and extracting all transaction data within that time period during data collection. (2) This article discusses the experimental results of XGBoost as a classifier, which can be used for comparative analysis of more classifiers (such as SVM, deep neural networks, etc.) in future work. (3) CT-GCN+ can be applied to other cryptocurrency related studies to a certain extent. There are two main reasons for this: (a) Our model is based on the analysis of nodes' behaviors and characteristics in the network, which are usually common in different cryptocurrency networks. And whether it is Ether or other cryptocurrencies, nodes in the network face similar phishing risks and threats. (b) Our model is based on collecting and analyzing transaction data and applying corresponding algorithms and models to identify and classify phishing nodes. However, these techniques are migratable in their fundamentals and can be applied to different cryptocurrency networks. Therefore, in future research, the application of this method to other cryptocurrencies can be considered.

Acknowledgements
Not applicable.

Author contributions
TF designed and supervised the study; BF collected the datasets and developed the methods; YW did the manuscript translation and typesetting. All authors contributed to the writing and the interpretation of the results, and read and approved the final version of the manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets used and analyzed during the current study available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
This article does not contain any studies with human participants or animals performed by any of the authors.

Competing interests
The authors declare no competing interests.

Received: 12 July 2023 Accepted: 13 October 2023
Published online: 01 February 2024

References

Alkhalil Z, Hewage C, Nawaf L, Khan I (2021) Phishing attacks: a recent comprehensive study and a new anatomy. Front Comput Sci 3:563060
Ando S, Huang CY (2017) Deep over-sampling framework for classifying imbalanced data. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 770–785
Anti-Phishing Working Group (2022) Phishing activity trends report, 3th Quarter 2022, https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf
Bahnsen AC, Bohorquez E, Villegas S, Vargas J, Gonzalez FA (2017) Classifying phishing URLS using recurrent neural networks. In: 2017 APWG symposium on electronic crime research (eCrime), pp 1–8
Bingxue Fu, Xing Yu, Feng T (2022) CT-GCN: a phishing identification model for blockchain cryptocurrency transactions. Int J Inf Secur 21:1223–1232
CHAINDIGG (2022) Blockchain and virtual currency crime trends research report. http://cdnf.chaindigg.com/baogao/2022_blockchain_crime_report.pdf
Chen W, Guo X, Chen Z, Zheng Z, Lu Y (2020) Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem. InIJCAI 7:4456–4462
Chen Y, Chen H, Zhang Y, Han M, Siddula M, Cai Z (2022) A survey on blockchain systems: attacks, defenses, and privacy preservation. High-Confid Comput 2:100048
Chen W, Zheng Z, Cui J, Ngai E, Zheng P, Zhou Y (2018) Detecting ponzi schemes on ethereum: Towards healthier blockchain technology, pp 1409–1418
Farrugia S, Ellul J, Azzopardi G (2020) Detection of illicit accounts over the Ethereum blockchain. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113318
Feng T, Yue C (2020) Visualizing and interpreting RNN models in Url-based phishing detection. In: Proceedings of the 25th ACM symposium on access control models and technologies, SACMAT '20. New York, USA, pp 13–24. Association for Computing Machinery
Han Q, Jiajing W, Zheng Z (2020) Long-range dependence, multi-fractality and volume-return causality of ether market. Chaos Interdiscipl J Nonlinear Sci 30:011101
Jain AK, Gupta BB (2018) PHISH-SAFE: URL features-based phishing detection system using machine learning. Cyber Secur. https://doi.org/10.1007/978-981-10-8536-9_44
Jiajing Wu, Yuan Qi, Lin D, You W, Chen W, Chen C, Zheng Z (2022) Who are the phishers? phishing scam detection on ethereum via network embedding. IEEE Trans Syst Man Cybern Syst 52(2):1156–1166
KNOWNSEC Blockchain Lab (2023) 2022 blockchain typical security incident research summary. https://zhuanlan.zhihu.com/p/597098080
Lee XT, Khan A, Sen Gupta S, Ong YH, Liu X (2020) Measurements, analyses, and insights on the entire ethereum blockchain network. In: Proceedings of the web conference 2020 (WWW '20). Association for Computing Machinery, New York, NY, USA, pp 155–166.
Li Y, Cai Y, Tian H, Xue G, Zheng Z (2020) Identifying illicit addresses in bitcoin network. Blockchain Trust Syst 1267:99–1118
Li S, Gou G, Liu C, Hou C, Li Z, Xiong G (2022) TTAGN: temporal transaction aggregation graph network for ethereum phishing scams detection. In: Proceedings of the ACM web conference 2022 (WWW '22), pp 661–669
Lin D, Jiajing Wu, Yuan Qi, Zheng Z (2019) T-EDGE: temporal weighted multidigraph embedding for ethereum transaction network analysis. Front Phys 8:204
Ma J, Saul L, Savage S, Voelker G (2009) Beyond blacklists: learning to detect malicious web sites from suspicious URLS, pp 1245–1254
Ramzan Z (2010) Phishing attacks and countermeasures. Springer, Berlin, pp 433–448
Rekouche K (2011) Early phishing. Computer Science
Sheng S, Wardman B, Warner G, Cranor LF, Hong J, Zhang C (2009) An empirical analysis of phishing blacklists. In: CEAS 2009
Tan CL, Chiew KL, Yong KSC, Sze SN, Abdullah J, Sebastian Y (2020) A graph-theoretic approach for the detection of phishing webpages. Comput Secur 95:101793
Verma R, Dyer K (2015) On the character of phishing URLS: accurate and robust statistical learning classifiers. pp 111–121
Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang F-Y (2019) Blockchain-enabledsmart contracts: architecture, applications, and future trends. IEEE Trans Syst Man Cybern Syst 49(11):2266–2277

Wang J, Chen P, Yu S, Xuan Q (2021) TSGN: transaction subgraph networks for identifying ethereum phishing ACCounts. In: Dai HN, Liu X, Luo DX, Xiao J, Chen X (eds) Blockchain and trustworthy systems. BlockSys 2021. Communications in Computer and Information Science, vol 1490. Springer, Singapore.

Whittaker C, Ryner B, Nazif M (2010) Largescale automatic classification of phishing pages. In: Network and distributed system security symposium

Xie Y, Jin J, Zhang J, Yu S, Xuan Q. Temporal-amount snapshot multigraph for ethereum transaction tracking. CoRR, abs/2102.08013,2021

Yu S, Jin J, Xie Y, Shen J, Xuan Q (2021) Ponzi scheme detection in ethereum transaction network. In: Dai HN, Liu X, Luo DX, Xiao J, Chen X (eds) Blockchain and trustworthy systems. BlockSys 2021. Communications in Computer and Information Science, vol 1490. Springer, Singapore

Yuan L, Zeng Z, Lu Y, Ou X, Tao F (2020) A character-level BiGRU-attention for phishing classification, pp 746–762

Yuan Q, Huang B, Zhang J, Wu J, Zhang H, Zhang X (2020) Detecting phishing scams on ethereum based on transaction records. In: 2020 IEEE international symposium on circuits and systems (ISCAS), pp 1–5

Zhang J, Porras P, Ullrich J (2008) Highly predictive blacklisting. In: Proceedings of the 17th conference on security symposium, SS'08, USENIX Association, USA, pp 107–122

Zhao T, Zhang X, Wang S (2021) GraphSMOTE: imbalanced node classification on graphs with graph neural networks. In: Proceedings of the 14th ACM international conference on web search and data mining (WSDM '21), 833–841

Zheng J, Zeng Z, Feng T (2022) Gcn-eta: High-efficiency encrypted malicious traffic detection. Secur Communi Netw. https://doi.org/10.1155/2022/4274139

Zhu H, Chen J, Li Z, Yin S (2021) Blockchain anomaly transaction detection method based on multi feature adaptive fusion. J Commun 42(05):41–50

Publisher's Note