electronics

MDPI

Article

Multiscale Feature Fusion and Graph Convolutional Network for Detecting Ethereum Phishing Scams

Zhen Chen, Jia Huang, Shengzheng Liu and Haixia Long *

College of Information Science Technology, Hainan Normal University, No. 99 Longkun South Road, Haikou 571158, China; cz221201@hainnu.edu.cn (Z.C.); hj221204@hainnu.edu.cn (J.H.); lsz231209@hainnu.edu.cn (S.L.)

* Correspondence: lhx@hainnu.edu.cn; Tel.: +86-13698981787

Abstract: With the emergence of blockchain technology, the cryptocurrency market has experienced significant growth in recent years, simultaneously fostering environments conducive to cybercrimes such as phishing scams. Phishing scams on blockchain platforms like Ethereum have become a grave economic threat. Consequently, there is a pressing demand for effective detection mechanisms for these phishing activities to establish a secure financial transaction environment. However, existing methods typically utilize only the most recent transaction record when constructing features, resulting in the loss of vast amounts of transaction data and failing to adequately reflect the characteristics of nodes. Addressing this need, this study introduces a multiscale feature fusion approach integrated with a graph convolutional network model to detect phishing scams on Ethereum. A node basic feature set comprising 12 features is initially designed based on the Ethereum transaction dataset in the basic feature module. Subsequently, in the edge embedding representation module, all transaction times and amounts between two nodes are sorted, and a gate recurrent unit (GRU) neural network is employed to capture the temporal features within this transaction sequence, generating a fixed-length edge embedding representation from variable-length input. In the time trading feature module, attention weights are allocated to all embedding representations surrounding a node, aggregating the edge embedding representations and structural relationships into the node. Finally, combining basic and time trading features of the node, graph convolutional networks (GCNs), SAGEConv, and graph attention networks (GATs) are utilized to classify phishing nodes. The performance of these three graph convolution-based deep learning models is validated on a real Ethereum phishing scam dataset, demonstrating commendable efficiency. Among these, SAGEConv achieves an F1-score of 0.958, an AUC-ROC value of 0.956, and an AUC-PR value of 0.949, outperforming existing methods and baseline models.

Keywords: blockchain; Ethereum; phishing scam detection; graph convolutional networks; feature fusion

Citation: Chen, Z.; Huang, J.; Liu, S.; Long, H. Multiscale Feature Fusion and Graph Convolutional Network for Detecting Ethereum Phishing Scams. *Electronics* **2024**, *13*, 1012. https://doi.org/10.3390/electronics13061012

Academic Editor: Krzysztof Szczypiorski

Received: 10 February 2024
Revised: 29 February 2024
Accepted: 4 March 2024
Published: 7 March 2024

1. Introduction

Blockchain technology, serving as the cornerstone for numerous cryptocurrencies such as Bitcoin and Ethereum [1], boasts a transaction processing capacity of 14.8 transactions per second [2] and ensures the permanent recording of transactions between parties. In addition, blockchain technology has also achieved initial success in fields such as manufacturing [3], agri-food [4], healthcare [5], and energy [6]. The ascent of blockchain has ushered in profound economic and technological shifts, particularly in digital assets and cryptocurrencies. However, this rapid evolution inevitably spawns a spectrum of security challenges, turning cryptocurrency and blockchain platforms into hotbeds for diverse criminal activities [7]. Since 2017, phishing scams on the Ethereum platform have constituted up to 50% of incidents [8], with victims incurring losses of up to USD 645,000 within a single week and attackers amassing illegal profits exceeding USD 3,000,000 within a month [9]. Such substantial economic damages undermine stakeholders' confidence in blockchain

development; hence, pinpointing phishing scams on Ethereum emerges as a paramount concern for researchers [10,11].

Traditional phishing often involves constructing counterfeit platforms like fake websites or software [12]. Attackers deceive users into believing they are interacting with legitimate entities, aiming to harvest sensitive information or prompt monetary transactions [13]. Such activities can manifest across various online platforms, including emails [14], social media, and sham websites. Nonetheless, within the Ethereum context, attackers eschew reliance on fabricated platforms, opting instead for tactics like high-reward promotions or directly disseminating false transaction addresses to victims via emails and chat groups, thus directly inducing cryptocurrency transfers [15]. These fraudulent activities predominantly occur within blockchain-related social media, chat groups, and Initial Coin Offering (ICO) platforms. Given the absence of a fixed pattern in Ethereum phishing scams, traditional detection methods prove ineffective [16].

Current research endeavors are focused on detecting phishing scam accounts on blockchain platforms like Ethereum. Researchers extract information from Ethereum user transaction data, categorize users into nodes based on unique addresses, and classify these nodes by capturing their features [17]. Presently, two predominant methodologies exist for feature extraction from nodes: one relies on manual feature engineering [18,19], employing traditional machine learning models for classification; the other involves applying various graph embedding techniques to the Ethereum transaction network to extract deep features [20]. Manual feature engineering-based methods heavily depend on the manual extraction of pertinent and productive features, such as out-degree, in-degree, total transaction amount, and time of the last transaction for a user. This process, demanding professional involvement, is both time-consuming and labor-intensive [9], and the extracted features merely represent user states, failing to encapsulate the relations between users, thereby marginally enhancing the accuracy of phishing detection. Adopting graph embedding techniques marks a novel breakthrough, yet it predominantly focuses on the topological structure within the transaction network, overlooking copious temporal transaction data [21]. Consequently, existing techniques inadequately integrate the topological and temporal transaction information between nodes, resulting in embeddings that insufficiently represent nodes and, in turn, impair phishing detection performance.

To address these challenges, this article proposes a multiscale feature fusion model leveraging graph convolutional networks for detecting Ethereum phishing scams. Our research contributes in the following ways:

(1) This study introduces a multiscale feature fusion model to detect phishing scam accounts on Ethereum. This model integrates manually extracted basic features with aggregated temporal transaction information and combines these with the topological structure of the transaction network, yielding comprehensive, practical, and in-depth features. Experimental outcomes demonstrate that this model achieves superior F1-scores, AUC-ROC values, and AUC-PR values in detecting Ethereum phishing scam accounts, surpassing existing methods and baseline models.
(2) The GRU mines all temporal transaction data between target nodes and their first-order neighbors, generating edge embedding representations. An attention mechanism assigns weights to these edge embeddings, aggregating them with structural relationships into the nodes to form time trading features, thereby further enriching the node embedding representations.
(3) Graph convolutional-based deep learning models detect Ethereum phishing scams, categorizing them based on nodes. The efficacy of this model is validated through comparisons with random walks, deep learning, and machine learning approaches.

The remainder of this article is organized as follows. Section 2 delineates the differences between traditional phishing networks and Ethereum phishing networks, and introduces the existing techniques for detecting Ethereum phishing. Section 3 details the technical specifics and the overarching detection framework of the proposed model, which employs multiscale feature fusion and graph convolutional networks for Ethereum phish-

ing scam detection. Subsequently, in Section 4, we present the evaluation metrics and loss functions for assessing the performance of phishing detection. Section 5 describes the dataset and experimental setup used, evaluates the performance of the proposed model in detecting phishing on Ethereum, and analyzes the sensitivity of the model parameters. Finally, conclusions and future work are discussed in Section 6.

2. Related Works

Traditional phishing often involves impersonating genuine company websites to harvest users' personal information, such as phone numbers, passwords, home addresses, etc. [22]. To tackle this issue, researchers have proposed multiple solutions [23]. For instance, Zouina and Outtaj [24] introduced a novel, lightweight phishing detection method entirely based on the Uniform Resource Locator (URL), employing the Support Vector Machine (SVM) as a classifier, and achieved a recognition rate of 95.80%. Compared to traditional phishing scenarios, phishing methods on Ethereum are more diverse. Phishers can send fake emails or messages to a vast pool of potential buyers before a cryptocurrency launch [25], luring them into transferring funds to a specific address. Consequently, conventional URL-based phishing detection methods do not apply to issues encountered on Ethereum.

Current Ethereum phishing scam detection methods predominantly fall into two categories. The former relies on professionals manually extracting pivotal features, followed by classification through machine learning algorithms. For instance, Chen et al. [19] constructed a transaction graph by collecting all transactions and labeling phishing addresses in Ethereum and proposed a graph-based cascade feature extraction method. They also introduced the dual-sampling ensemble algorithm, integrating multiple basic models trained through sampling samples and features, ultimately selecting LightGBM as the classifier. This approach successfully identified phishing accounts within the Ethereum blockchain system. Wen et al. [26] proposed two phishing detection frameworks: a feature learning-based framework and a phishing obfuscation framework based on inserted transaction records. Within the feature learning-based framework, they mined transaction records to extract multiple transaction features, including account and network features, combining these with machine learning models for phishing detection. Experiments demonstrated the effectiveness of this framework in phishing detection. Furthermore, a phishing obfuscation framework based on inserted transaction records was designed to achieve unidirectional phishing obfuscation by inserting malicious transaction records into accounts, thereby validating the robustness of the detection framework.

The latter category employs graph embedding techniques to reduce the dimensionality of high-dimensional graph data, constructing transaction graphs to obtain nodes' embedding representations. For instance, Grover et al. [27] introduced Node2vec. This method maximizes the preservation of neighborhood possibilities in a node network by learning the mapping of nodes to a low-dimensional feature space. This approach effectively explores different neighbors and retains the local structural features of the data. Building on this, Wu et al. [28] proposed a novel network embedding algorithm, trans2vec. Unlike Node2vec, trans2vec does not randomly select the next node as part of the node sequence. It obtains edge weights based on the total amount of transactions between two nodes and the time of the last transaction, proving more suitable for Ethereum phishing detection. Chen et al. [21] viewed accounts and transactions as nodes and edges, respectively, and proposed an Ethereum phishing scam detection method using GCN and an autoencoder. This method effectively aggregates the network's node features and spatial structure and can be extended to general feature engineering, emphasizing the importance of topological structure for node representation. Compared to some baseline methods, this approach demonstrated optimal performance.

Wen et al. [7] depart from graph embedding methods, introducing the hybrid deep learning model LBPS, which is LSTM (Long Short-Term Memory)—FCN (Fully Convolutional Network) and BP neural network-based phishing scam accounts detection model.

This method provides a novel approach for feature extraction from transaction records. It utilizes a BP neural network to capture implicit relationships among features extracted from transaction records. It employs an LSTM-FCN neural network to capture the temporal features in all transaction records of the target accounts. Experimental results indicate that LBPS outperforms baseline model methods and existing approaches, with an F1-score of 97.86%.

3. Methods

In this section, as illustrated in Figure 1, the proposed framework is elucidated in detail, comprising four main modules: basic feature module, edge embedding representation module, time trading feature module, and phishing node classification module based on GCN. The real Ethereum dataset, collected by Lin et al. [29], is utilized. It was sourced from the Ethereum block explorer and analytics platform Etherscan (etherscan.io, (accessed on 1 March 2024)) via its API, gathering historical transaction data of target accounts. Considering the vast scale of total transaction records, the K-order subgraph sampling method [30] is employed to capture the local structure of target accounts. After the data processing is completed, we select a set of target nodes and their neighboring nodes, extract all transaction data between them, and sort these data by transaction time. Subsequently, we input the time series data into a GRU to obtain adjustable dimension data, that is, edge embedding representations. This operation is shown in the red area of Figure 1. By repeating the above steps for the target nodes, we can obtain all edge embedding representations between the target node and its surrounding neighbor nodes. Next, we employ an attention mechanism to capture the information acquisition weights of the edge embedding representations. Multiplying the edge embeddings by the weights and then aggregating them yields the target node's time trading features, as shown in the time trading features module in the blue area of Figure 1. Finally, the obtained time trading features are concatenated with the node's basic features to achieve a complete feature representation of the node, and a graph convolutional neural network is used to detect the node.

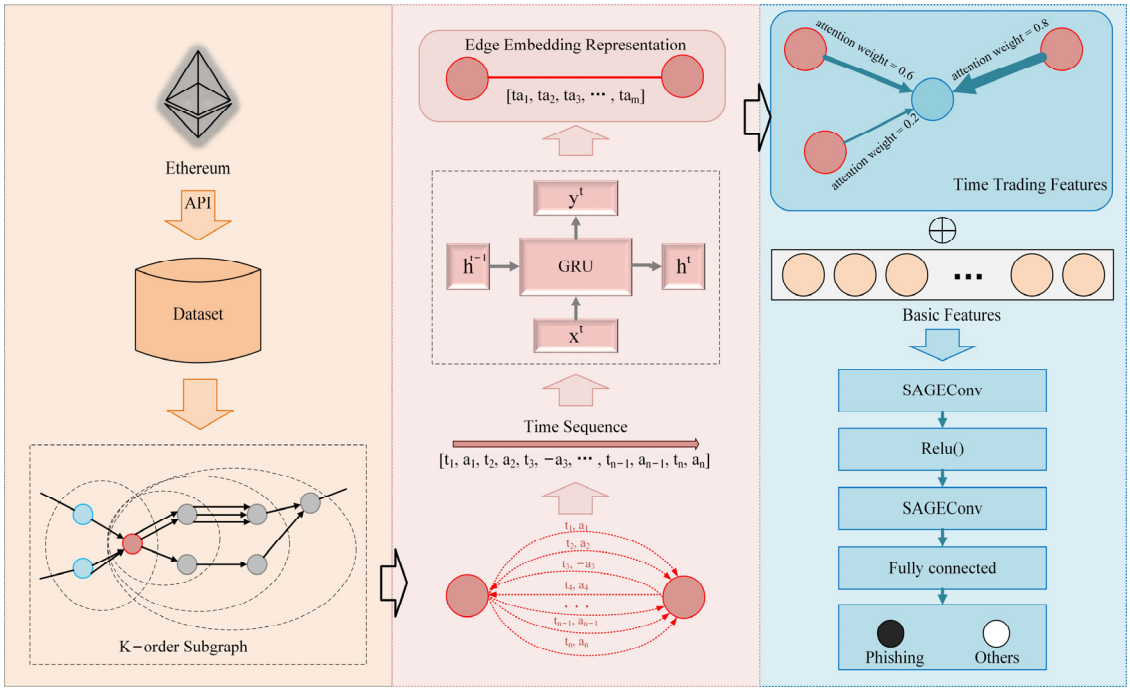


Figure 1. Overall frame diagram.

3.1. Basic Features

Blockchain is characterized by decentralization, immutability, openness, and anonymity. Hence, on the Ethereum platform, due to its openness, transaction records of all nodes (users) are accessible. However, the anonymity aspect precludes obtaining specific features of nodes. The initial dataset encompasses four features: from, to, transaction amount, and timestamp. Based on the acquired initial dataset, a 12-dimensional feature set was devised as the basic features for the target nodes. These basic features include total transaction count, outgoing transaction count, incoming transaction count, total neighbor count, average transaction amount, total transaction amount, average outgoing transaction amount, average incoming transaction amount, total outgoing transaction amount, total incoming transaction amount, the amount of the last outgoing transaction, and the amount of the previous incoming transaction.

3.2. Edge Embedding Representation

In the basic feature module, temporal information is noted to be overlooked. Consequently, this module introduces temporal information to diversify the feature set further. It processes a large volume of temporal transaction information between two nodes into fixed-dimensional data to obtain node embedding representations better in subsequent steps.

Specifically, a target node is first selected, and all transaction information between it and one of its neighboring nodes is gathered: timestamps and transaction amounts. Notably, the transaction amount is positive for transactions initiated by the target node and negative for received transactions, enhancing phishing scam detection performance by incorporating transaction direction. Assuming target node o and neighbor node h, after processing and ascendingly sorting transactions between the node pair <o, h>, the time sequence can be obtained. The formula can be defined as follows:

$$Time\ Sequence = [t_1,\ a_1,\ t_2,\ a_2,\ t_3,\ -a_3,\ \cdots,\ t_n,\ a_n] \tag{1}$$

where t_n represents the timestamp of the nth transaction and a_n represents the amount of the nth transaction, with the sign indicating transaction direction.

The obtained time series data are inputted into the GRU, as it selectively updates and forgets information, demonstrating the capacity to model dependencies in time series data over extended periods. This allows the GRU to capture deep temporal features between nodes. Compared to LSTM, it has a more straightforward structure, making it easier to train and perform better with less data. Below are the mathematical expressions and parameter descriptions of the GRU [31], starting with the reset gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2}$$

where r_t is the reset gate vector at time step t, σ is the sigmoid function, W_r is the weight matrix of the reset gate, h_{t-1} is the hidden state of the previous time step, and x_t is the input of the current time step. Next is the update gate:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{3}$$

where z_t is the update gate vector at time step t and W_z is the weight matrix for the update gate. Next is the candidate hidden state:

$$\widetilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \tag{4}$$

where $\widetilde{h}_t$ is the candidate's hidden state at time step t and W is the weight matrix for the candidate's hidden state. Next is the final hidden state:

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \widetilde{h}_t \tag{5}$$

where h_t is the final hidden state at time step t, which is the weighted sum of the hidden state from the previous time step h_{t-1} and the current candidate's hidden state $\widetilde{h}_t$. After inputting the time series into the GRU, we can obtain the EER (edge embedding representation). The formula is defined as follows:

$$\begin{aligned} EER &= GRU([t_1,\ a_1,\ t_2,\ a_2,\ t_3,\ -a_3,\ \cdots,\ t_n,\ a_n]) \\ &= [ta_1,\ \ ta_2,\ ta_3, \cdots,\ ta_m] \end{aligned} \tag{6}$$

where ta_m represents the mth dimensional edge embedding representation between nodes o and h, with m being the set dimension of edge embedding representation. The GRU captures the deep temporal relationships of ordered transaction times and directed amounts between nodes in this module. The obtained fixed-dimensional m-edge embedding representation lays the foundation for subsequent analysis and processing.

3.3. Time Trading Features

In Ethereum, the transaction network differs from other modes, as nodes do not carry representative information. Research indicates that reliance solely on manually designed features does not comprehensively reflect the nodes' state, necessitating more node feature capture. In Ethereum, a node often engages in multiple transactions with other nodes. The aim is to integrate these transactions into the feature representation of the node. Hence, edge embedding representations for any two nodes have been obtained in previous modules. The subsequent challenge lies in aggregating all edge embedding representations for a target node. The key to solving the aggregation issue is obtaining the weight for each edge. The adopted approach utilizes an attention mechanism to capture the information of edge embedding representations and derive weights, detailed by the following formula:

$$weight = softmax(GRU([t_1, a_1, t_2, a_2, t_3, -a_3, \cdots, t_n, a_n])) \tag{7}$$

Upon obtaining all of the weights of the target node o, we can multiply the edge embedding representation and the weights and then aggregate them to obtain the TTF (time trading features) of the node o. The formula is as follows:

$$TTF_o = \sum_{h \in N_o} EER_h \cdot weight_h \tag{8}$$

where TTF_o represents the time trading features of node o, N_o denotes the neighboring edges of node o, edge $h \in N_o$, EER_h is the embedding representation of edge h, and $weight_h$ is the weight of edge h.

3.4. Phishing Scam Detection Based on GCN

Following these steps, the nodes' basic and time trading features have been acquired. Next, these two are concatenated to form a comprehensive feature representation of the node. To determine whether a node is a phishing node and to further integrate the topological information of the node into its features, three distinct types of graph convolutional neural networks will be utilized for experimentation: GCN, SAGEConv, and GAT.

GCN is a commonly used neural network for graph-structured data [32]. Its core concept involves updating the feature representation of a node by aggregating its features and those of neighboring nodes. The fundamental formula of GCN is as follows:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{9}$$

where $H^{(l)}$ represents the node feature matrix at layer l; $\widetilde{A} = A + I_N$, where A is the adjacency matrix of the graph and I_N is the identity matrix. This operation introduces

self-connections, allowing nodes to consider their features while aggregating the features of neighbors. $\widetilde{D}$ is the degree matrix of $\widetilde{A}$, $W^{(l)}$ is the weight matrix at layer l, and σ is the activation function.

SAGEConv iteratively aggregates and updates the feature representations of nodes to capture neighbor information within the graph structure [33]. Unlike traditional GCN, SAGEConv designs a learnable aggregation function to aggregate the features of neighboring nodes, enabling GraphSAGE to better scale to large-scale graph data. The computation in a SAGEConv layer can be divided into two steps: aggregating the features of neighbor nodes and updating the features of the target node. The fundamental formula is as follows:

$$h_{N(v)}^{l} = Aggregate^{l}\left(\left\{h_{u}^{l-1}, \forall u \in N(v)\right\}\right) \tag{10}$$

$$h_{v}^{l} = \sigma\left(W^{l} \cdot CONCAT\left(h_{v}^{l-1}, h_{N(v)}^{l}\right)\right) \tag{11}$$

where h_v^l represents the feature vector of node v at layer l; $N(v)$ denotes the set of neighbor nodes of node v; $Aggregate^l$ is the aggregation function at layer l, used for aggregating the features of neighbor nodes; W^l is the trainable weight matrix at layer l; σ is the activation function; $CONCAT$ represents the concatenation operation, used to concatenate the features of node v itself with the aggregated neighbor features.

GAT introduces an attention mechanism, dynamically determining the contribution of neighbor nodes on the feature update of the central node by calculating the attention coefficients between nodes [34]. This mechanism allows GAT to handle irregular graph data and adaptively determine the importance between nodes, enhancing the expressive capability of the model for the graph structure. The formula is as follows:

$$h_{i}' = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W h_{j}\right) \tag{12}$$

$$\alpha_{ij} = softmax\left(LeakyReLU\left(a^{T}\left[Wh_{i} \parallel Wh_{j}\right]\right)\right) \tag{13}$$

where h_i represents the feature vector of node i, h_i' represents the updated feature vector of node i, $N(i)$ denotes the set of neighbor nodes of node i, W is the trainable weight matrix, α_{ij} is the attention coefficient between nodes i and j, a is the trainable weight vector of the attention mechanism, and $\parallel$ denotes the concatenation operation.

4. Metrics

Phishing scam detection is a binary classification problem. Accuracy, precision, F1-score, recall, AUC-ROC, and AUC-PR will be used to evaluate the performance of the proposed model. These metrics depend on four terms: true positive (TP), true negative (TN), false negative (FN), and false positive (FP) [35]. These correspond to the number of positive cases correctly predicted by the model, the number of negative cases correctly predicted by the model, the number of positive cases predicted as negative by the model, and the number of negative cases predicted as positive. The specific formulas are as follows:

$$Accuracy = \frac{TP + FN}{TN + FN + TP + FP} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{16}$$

$$Recall = \frac{TP}{FN + TP} \tag{17}$$

The closer the ROC curve is to the top left corner, the better the performance of the classifier. AUC-ROC is an important metric for measuring the quality of classifiers; the larger the AUC value, the better the model performance. The PR curve reflects the relationship between precision and recall of the model. AUC-PR measures how many true positives the model can capture while maintaining a high precision rate. The loss function used is Cross Entropy Loss, which measures the difference between the model's predicted probability distribution and the true label's probability distribution. The formula can be defined as follows:

$$Loss = -(y\log p + (1-y)log(1-p)) \tag{18}$$

where y is a binary variable indicating whether the category is correct or not; p represents the probability of the model correctly predicting the category.

5. Experiments and Results

5.1. Data Description

The dataset employed real Ethereum data acquired by Lin et al. [29], comprising 445 nodes identified as phishing nodes by Etherscan and an equal number of unlabeled random nodes. Additionally, K-order sampling was utilized to capture the local information of accounts. This sampling method generates a directed K-order subgraph centered around each target account. Considering the potential for illicit transactions in the preceding node and the subsequent three nodes of the target node, the dataset collated qualifying subgraphs for 890 target nodes, eventually amalgamating into a large-scale network with 86,623 nodes. The dataset was partitioned into 70% for training and the remainder for testing.

5.2. Experimental Environment

The experimental section was implemented using the Python programming language. The runtime and testing environment included a 13th Gen Intel(R) Core (TM) i9-13900HX 2.20 GHz CPU, sourced from Intel Corporation, Santa Clara, CA, USA, NVIDIA GeForce RTX 4060 Laptop GPU and 16.0 GB RAM, sourced from NVIDIA Corporation, Santa Clara, CA, USA. Windows 11 operating system, version 22H2, is developed and produced by Microsoft Corporation, headquartered in Redmond, WA, USA, and PyTorch version 1.13.1.

To comprehensively evaluate the performance of the proposed phishing scam detection model, it was compared with traditional deep learning models, deep learning models integrating attention mechanisms, and classic machine learning models, as well as methods like LightGBM [36], Deep Walk [37], and Node2Vec [27]. During the experimental process, particular attention was paid to two key factors: timestamp mapping [38] and GRU aggregation dimensions. The impact of these factors on model performance is significant. Through in-depth analysis of timestamp mapping at different time steps and diverse settings for GRU aggregation dimensions, this study revealed these factors' influence on the model's overall performance. This process optimized the precision of phishing scam detection and enhanced its efficiency, providing more effective technical support for combating phishing scams. Moreover, to understand the significance of each module, ablation experiments were designed using only basic features and only time trading features [39]. The hyperparameters of the graph convolution-based deep learning model used are shown in Table 1.

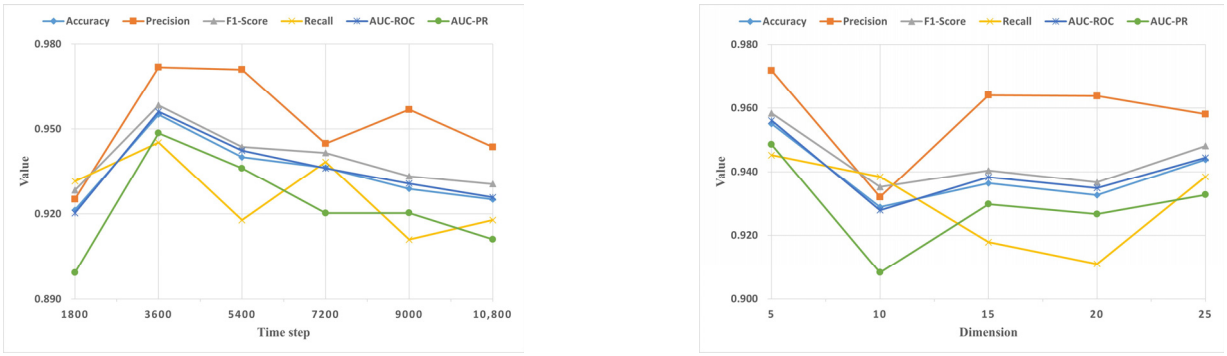
Table 1. Graph convolution model parameters.

Settings	Hyperparameters
Graph convolution layers	Layer 1: GraphConv (17, 128) Layer 2: ReLU () Layer 3: GraphConv (128, 17)
FC layer	Layer 1: Linear (17, 2)
Configuration	Epoch = 500; learning rate = 0.001; batch size = 256; optimizer = 'Adam'; loss = 'Cross Entropy Loss'

5.3. Results

5.3.1. Performance of Different Parameters

To explore the impact of time step length and GRU aggregation dimension on model performance, experiments were initially conducted with the SAGEConv model, focusing on different time step lengths and aggregation dimensions. Figure 2a displays the trend of model metrics on the test set with varying time step lengths. The X-axis represents time in seconds, incrementing by 1800 s per time step. It is evident from the graph that when the time step length reaches 3600 s, all metrics attain peak performance. However, as the time step length increases, a general downward trend in metrics is observed, reaching the lowest performance at 10,800 s. This phenomenon indicates that excessively low or high time step lengths may adversely affect model performance. Therefore, when selecting the time step length, careful consideration of model performance is imperative to ensure an appropriate time interval, allowing the model to capture and utilize temporal transaction information more accurately for optimal detection effectiveness.

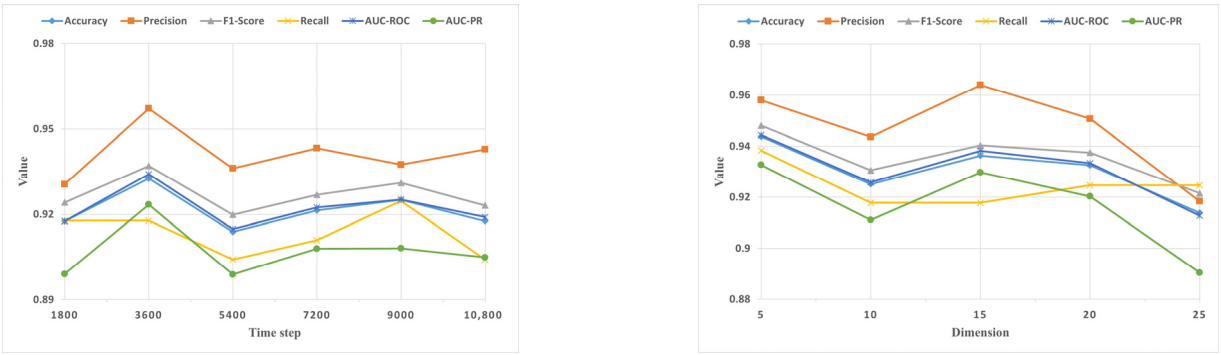


(**a**) Performance of SAGEConv model under different time steps

(**b**) Performance of SAGEConv model under different aggregation dimensions

Figure 2. Performance of SAGEConv model under different time steps and aggregation dimensions.

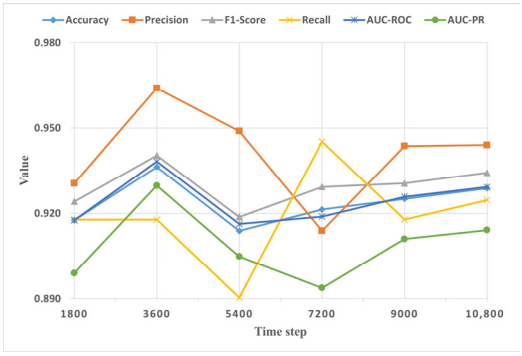
Figures 3a and 4a demonstrate the metric trends for the GCNConv and GATConv models under different time step lengths. Observations indicate that most evaluation metrics for these two models peak when the time step length is set to 3600 s. Although the recall value reaches its optimum at 9000 s (for GCNConv) and 7200 s (for GATConv), considering the overall performance of the models, 3600 s is determined as the optimal time step length for all three graph convolutional models. This decision is based on a comprehensive model performance evaluation, ensuring balanced and superior detection results without compromising other critical metrics.



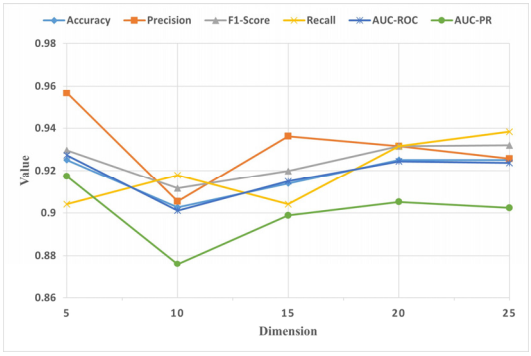
(**a**) Performance of GCNConv model under different time steps

(**b**) Performance of GCNConv model under different aggregation dimensions

Figure 3. Performance of GCNConv model under different time steps and aggregation dimensions.



(**a**) Performance of GATConv model under different time steps



(**b**) Performance of GATConv model under different aggregation dimensions

Figure 4. Performance of GATConv model under different time steps and aggregation dimensions.

Figure 2b details the trend of various metrics for the SAGEConv model on the test set as the aggregation dimension changes. The results show that the model exhibits high performance at smaller aggregation dimensions, reflecting the efficiency of the GRU in aggregating temporal transaction information. When the aggregation dimension increases from 15 to 25, the fluctuation in model performance remains minimal, within a range of 0.02. This suggests that the model is robust against fluctuations in higher aggregation dimensions.

In contrast, Figure 4b shows similar trends of metrics for the GATConv model, while the GATConv model in Figure 3b exhibits a different pattern, particularly when the aggregation dimension exceeds 15, where the overall performance of the model tends to decline, indicating that an increase in aggregation dimension adversely affects model performance. Notably, at an aggregation dimension of 5, most metrics for all three models reach their performance peak. Therefore, after considering various metrics and model performances, 5 is ultimately selected as the optimal aggregation dimension for all three graph convolutional models.

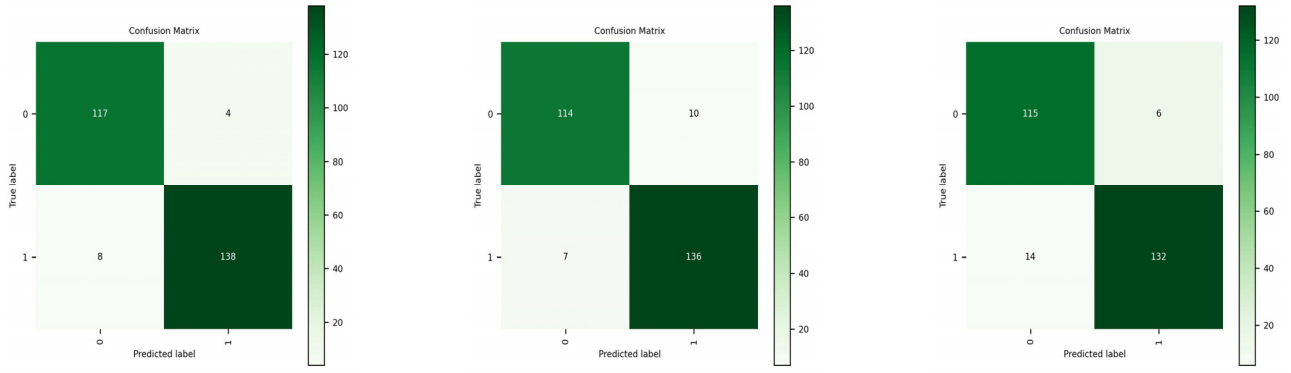
5.3.2. Performance of Different Models

For a comprehensive evaluation of the performance of the proposed model, it was compared with traditional deep learning models, classic machine learning models, and methods such as random walk. Specific comparison results are presented in Table 2. Three graph convolution-based deep learning methods were employed, including GCN, SAGEConv, and GAT. These methods achieved satisfactory results, with SAGEConv performing optimally, attaining an AUC-PR value of 0.949, AUC-ROC value of 0.956, F1-score of 0.958, precision value of 0.972, and accuracy value of 0.955. These metrics demonstrate superior performance compared to other models, except for the recall value, which is 0.945, lower than the value of 0.986 of GNB (Gaussian Naive Bayes). Deep learning includes three models: CNN (convolutional neural network), LSTM, and Attention-CNN. Table 2 shows that the overall performance of deep learning models surpasses traditional machine learning models and random walk methods but falls short of graph convolution-based deep learning models. This may be due to the inability of traditional deep learning models to learn the topological relationships between nodes, resulting in inferior performance. Among these, the Attention-CNN model performs relatively better than the GAT model. Random walk methods, Node2Vec, and Deep Walk show average performance, with Node2Vec performing relatively better as its AUC value exceeds traditional machine learning methods. This might be attributed to the biased selection of the next node by Node2Vec, allowing for better perception of the information preceding and succeeding a node. Among machine learning methods, the widely used LightGBM achieved an F1-score of 0.884, performing well compared to traditional machine learning and random walk methods.

Table 2. Performance of different models in detecting phishing scams.

Method	Models	Accuracy	Precision	F1-Score	Recall	AUC-ROC	AUC-PR
Graph convolution	SAGEConv	0.955	0.972	0.958	0.945	0.956	0.949
	GCNConv	0.936	0.951	0.941	0.932	0.937	0.923
	GATConv	0.925	0.957	0.930	0.904	0.927	0.917
Deep learning	CNN	0.914	0.956	0.918	0.884	0.917	0.908
	LSTM	0.906	0.942	0.912	0.884	0.909	0.896
	Attention-CNN	0.925	0.938	0.931	0.925	0.925	0.908
Random walk	Node2Vec	0.764	0.784	0.757	0.731	0.878	0.861
	Deep Walk	0.730	0.750	0.721	0.694	0.798	0.790
Machine learning	LightGBM	0.873	0.878	0.884	0.890	0.871	0.842
	RF	0.854	0.869	0.866	0.863	0.853	0.825
	SVM	0.622	0.593	0.720	0.959	0.554	0.575
	GNB	0.588	0.567	0.740	0.986	0.584	0.592

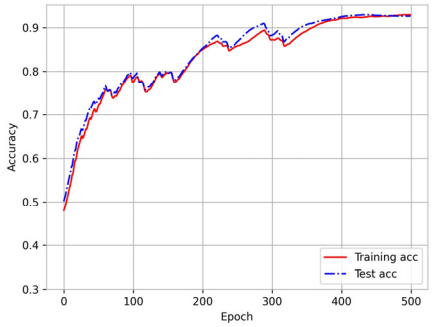
Furthermore, confusion matrices were utilized to analyze the SAGEConv, GCNConv, and GATConv models, with the corresponding results depicted in Figure 5, respectively. The numbers in the confusion matrices include the true positive rate and false negative rate for phishing scam classification. The numbers on the diagonal represent the quantity of correctly classified samples. In contrast, the numbers off the diagonal indicate the quantity of samples where predicted classifications differ from actual classifications, i.e., the quantity of misclassified samples. Analysis of the confusion matrices reveals that SAGEConv exhibits commendable performance on the Ethereum dataset, while the performances of GCNConv and GATConv are comparatively less satisfactory.



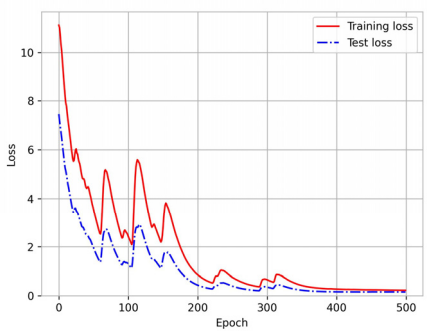
(**a**) (**b**) (**c**)

Figure 5. Confusion matrix of (**a**) SAGEConv model; (**b**) GCNConv model; (**c**) GATConv model.

For a more intuitive demonstration of model performance in phishing detection, accuracy and loss curves on the training and test sets of the Ethereum dataset are provided. Given the optimal performance of SAGEConv, only its results are displayed, specifically in Figure 6a,b. Due to excessive fluctuations in the original accuracy and loss curve data for SAGEConv, which did not adequately reflect the convergence trend, an exponentially weighted moving average [40] was employed to reduce data fluctuations, resulting in graphics that more accurately reflect the change in trends. Exponentially weighted moving average denotes the exponentially decreasing weighting coefficients over time, with coefficients for values closer to the current moment being larger. The smoothed curves indicate that loss and accuracy have stabilized, suggesting effective model training.



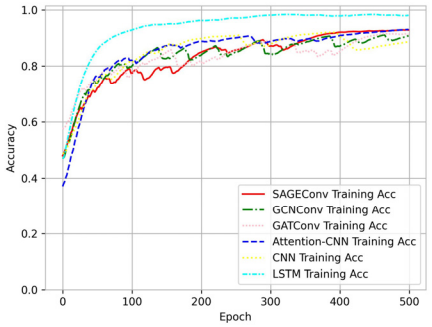
(**a**) Accuracy curves of SAGEConv in training and test sets



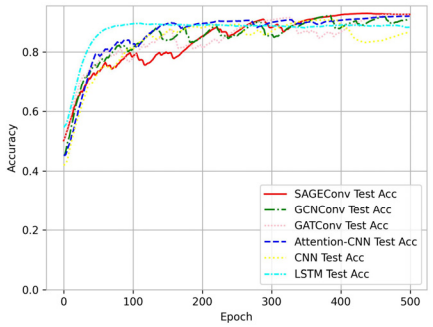
(**b**) Loss curves of SAGEConv in training and test sets

Figure 6. Accuracy and loss curves of SAGEConv in training and test sets.

In addition, comparative curves for accuracy and loss for the training and test sets are also provided between graph convolution-based deep learning methods and traditional deep learning methods. Figure 7a,b show the accuracy curves of different models for the training and test sets. It can be observed from the figure that the accuracy curve of LSTM for the training set is higher than other models, including SAGEConv, after convergence. However, in the test set, the curve of LSTM is only higher than that of CNN, while the accuracy of SAGEConv is higher. This indicates that LSTM may be overfitting on the training set, resulting in insufficient generalization performance. In contrast, although SAGEConv performed averagely on the training set, it performed well on the test set, indicating that the model has good generalization ability and robustness.



(**a**) Accuracy curves of different models for training set

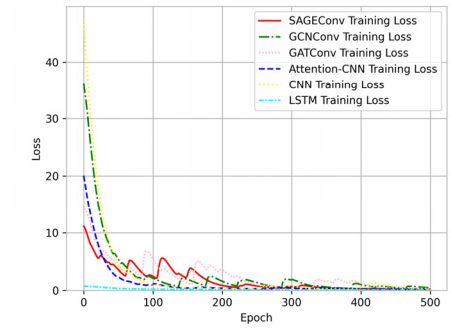


(**b**) Accuracy curves of different models for test set

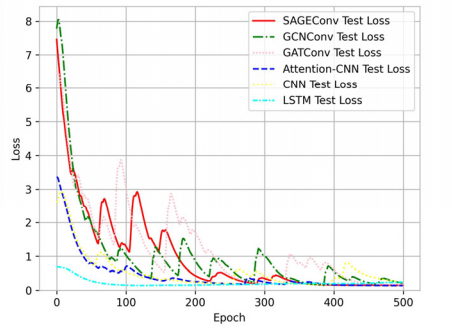
Figure 7. Accuracy curves of different models for training and test sets.

Figure 8a,b showcase the loss curves of different models for the training and test sets. It can be observed that LSTM achieves less loss from the start, with no significant change in loss as training epochs increase. This further suggests a potential overfitting issue with LSTM, resulting in mediocre classification effects. Conversely, the loss with SAGEConv gradually decreases with the increase in training epochs, ultimately converging and finally achieving better performance.

Figure 9a,b present the ROC and PR curves of different models for the test set, demonstrating the SAGEConv model's performance superiority. Figure 9a,b show that the top three highest AUC-ROC and AUC-PR values belong to graph convolution-based deep learning models, indicating that network models integrating node topological structures can detect phishing scams more effectively. Specifically, the top-performing model is SAGEConv, with an AUC-ROC value of 0.9561 and an AUC-PR value of 0.9485. This surpasses the other two graph convolution-based models and exceeds traditional deep learning models, machine learning models, and random walk methods.

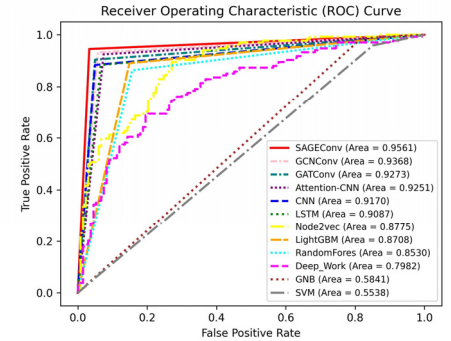


(**a**) Loss curves of different models for training set

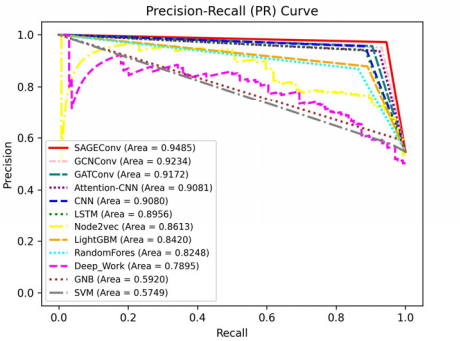


(**b**) Loss curves of different models for test sets

Figure 8. Loss curves of different models for training and test sets.



(**a**) ROC curves of different models for the test set



(**b**) PR curves of different models for the test set

Figure 9. ROC and PR curves of different models for the test set.

Overall, deep learning models perform better at handling this binary classification problem compared to random walk methods and machine learning models. Among random walk algorithms, Node2Vec is considered to have commendable performance among non-deep learning methods, revealing that in dealing with graph-structured data, random walk-based methods might hold more advantages over traditional machine learning methods. This suggests that random walk algorithms like Node2Vec can more effectively capture interactions and hidden patterns between nodes, providing a beneficial analytical tool for graph-structured data analysis of complex relationships and structural features of graph data.

5.3.3. Performance of Different Feature Fusion Methods

To validate the effectiveness of each module, the basic feature module and time trading feature module were removed separately, and the corresponding experimental results under the SAGEConv model are presented in Figure 10. The results indicate a decline in model detection performance following the removal of either module. Specifically, after the removal of the time trading feature module, the AUC-ROC of the model decreased by 3.5%, the AUC-PR by 4.7%, and the F1-score by 3%. This phenomenon is mainly attributed to the ability of the GRU to effectively learn and extract a significant amount of temporal transaction information between two nodes, thereby forming representative edge embeddings. This result highlights the importance of temporal transaction information in phishing scam detection and points to the critical role of time trading features within the model.

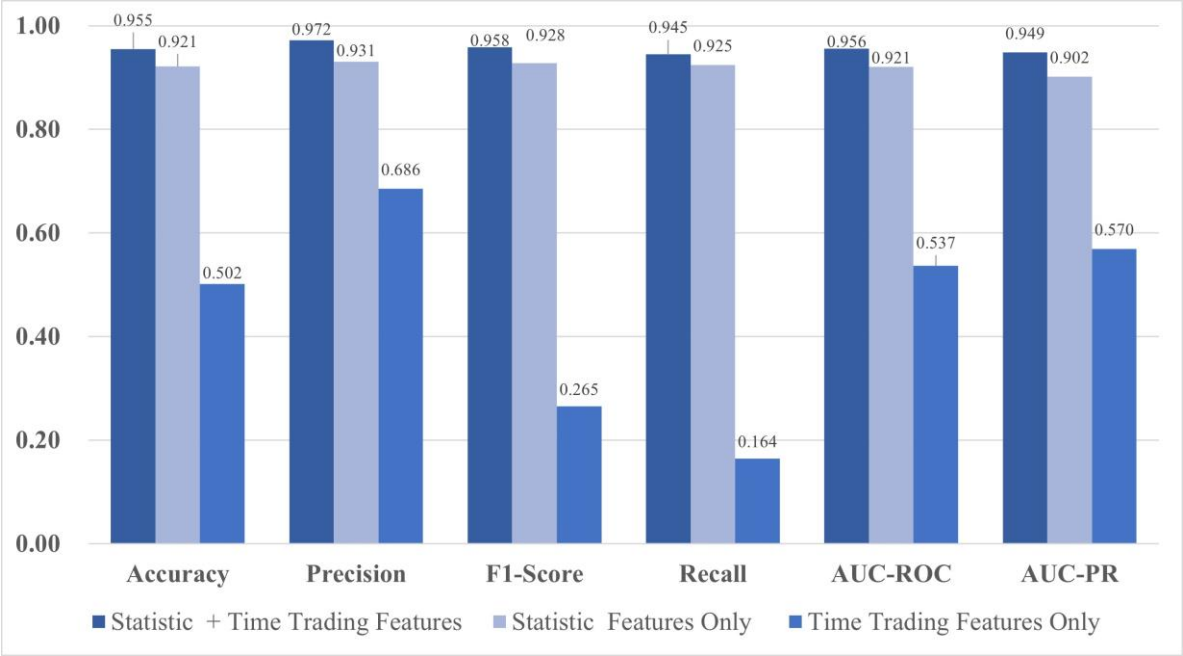


Figure 10. Effectiveness performance of each module.

A noticeable decline in performance metrics was observed after removing the basic feature module, underscoring the importance of manually designed features for the detection task. Especially in environments like Ethereum, where nodes inherently lack features, manually designed features become crucial in imparting explicit meanings to nodes. However, based on previous research, reliance solely on manually designed features is inadequate for further enhancing detection performance. This is primarily because integrating structural relationships and temporal information between nodes into the feature design process is challenging. This calls for a more considerate utilization and incorporation of interaction patterns and temporal dynamics between nodes in feature design to capture and utilize information within the Ethereum network more comprehensively. Therefore, the basic feature module and time trading feature module complement each other, both being indispensable. Together, they form a complete feature framework that fully leverages the basic characteristics of nodes and delves into the temporal transaction dynamics between nodes, offering a new solution for effective phishing scam detection.

6. Conclusions

To reduce the frequency of phishing scams in blockchain and attract more investors, this paper proposes a multiscale feature fusion method combined with a graph convolutional network to enhance the performance of phishing scam detection on Ethereum. This model integrates manually extracted basic features and aggregated temporal transaction information features and fuses them with the topological structure of the transaction network, yielding a more comprehensive, effective, and in-depth set of features, further enhancing the performance of Ethereum phishing detection.

Specifically, the basic features of nodes were first designed based on the Ethereum transaction dataset. Next, by employing the GRU, temporal transaction information between nodes was mined to obtain embedding representations of all edges between a node and its first-order neighbors. In the time trading feature module, weights were assigned to the embedding representations of each edge using an attention mechanism, aggregating structural and temporal transaction information into the nodes. Finally, combining the nodes' basic and time trading features, a graph convolution-based deep learning model was used to classify nodes. The model's effectiveness was evaluated using a real-world Ethereum dataset, demonstrating superior performance compared to existing methods.

While this study has achieved promising results on the dataset in question, the generalization and scalability of its capabilities remain to be assessed. Therefore, future research plans involve deploying the model in blockchain phishing scam detection scenarios. This will not only validate the robustness of the model but also lay a practical foundation for further optimization and enhancement of the model's performance. Currently, phishing scams remain a significant challenge in the blockchain domain, and it is hoped that more researchers and blockchain platform practitioners will join this line of research, working collectively to contribute to the construction of a safer and more reliable blockchain ecosystem.

Author Contributions: Conceptualization, Z.C.; methodology, Z.C. and J.H.; validation, Z.C., J.H. and S.L.; formal analysis, Z.C. and J.H.; investigation, Z.C.; resources, H.L.; data curation, Z.C.; writing—original draft preparation, Z.C.; writing—review and editing, H.L.; visualization, Z.C. and S.L.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62262019), the Hainan Provincial Natural Science Foundation of China (No. 823RC488, No. 620RC603, and No. 721QN0890), the Haikou Science and Technology Plan Project of China (No. 2022-016), and the Hainan Province Graduate Innovation Research Project (No. Qhys2023-408, Qhys2023-407).

Data Availability Statement: The dataset used in this study can be accessed via the following link: https://github.com/lindan113/xblock-network_analysis/tree/master/Phishing%20node%20classification (accessed on 1 March 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Roșca, I.; Butnaru, A.-I.; Simion, E. Security of Ethereum Layer 2s. *Cryptol. Eprint Arch.* **2023**, *2023*, 124.
2. Chen, H.; Pendleton, M.; Njilla, L.; Xu, S. A Survey on Ethereum Systems Security: Vulnerabilities, Attacks, and Defenses. *ACM Comput. Surv.* **2021**, *53*, 1–43. [CrossRef]
3. Leng, J.; Ruan, G.; Jiang, P.; Xu, K.; Liu, Q.; Zhou, X.; Liu, C. Blockchain-Empowered Sustainable Manufacturing and Product Lifecycle Management in Industry 4.0: A Survey. *Renew. Sustain. Energy Rev.* **2020**, *132*, 110112. [CrossRef]
4. Mirabelli, G.; Solina, V. Blockchain-Based Solutions for Agri-Food Supply Chains: A Survey. *IJSPM* **2021**, *17*, 1. [CrossRef]
5. Hasselgren, A.; Kralevska, K.; Gligoroski, D.; Pedersen, S.A.; Faxvaag, A. Blockchain in Healthcare and Health Sciences-A Scoping Review. *Int. J. Med. Inform.* **2019**, *134*, 104040. [CrossRef]
6. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain Technology in the Energy Sector: A Systematic Review of Challenges and Opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
7. Wen, T.; Xiao, Y.; Wang, A.; Wang, H. A Novel Hybrid Feature Fusion Model for Detecting Phishing Scam on Ethereum Using Deep Neural Network. *Expert Syst. Appl.* **2023**, *211*, 118463. [CrossRef]
8. Conti, M.; Kumar, E.S.; Lal, C.; Ruj, S. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452. [CrossRef]
9. Li, S.; Gou, G.; Liu, C.; Hou, C.; Li, Z.; Xiong, G. TTAGN: Temporal Transaction Aggregation Graph Network for Ethereum Phishing Scams Detection. In Proceedings of the ACM Web Conference 2022; ACM: Virtual Event, Lyon, France, 25 April 2022; pp. 661–669.
10. Wang, Z.; Jin, H.; Dai, W.; Choo, K.-K.R.; Zou, D. Ethereum Smart Contract Security Research: Survey and Future Research Opportunities. *Front. Comput. Sci.* **2021**, *15*, 152802. [CrossRef]
11. Wenhua, Z.; Qamar, F.; Abdali, T.-A.N.; Hassan, R.; Jafri, S.T.A.; Nguyen, Q.N. Blockchain Technology: Security Issues, Healthcare Applications, Challenges and Future Trends. *Electronics* **2023**, *12*, 546. [CrossRef]
12. Naqvi, B.; Perova, K.; Farooq, A.; Makhdoom, I.; Oyedeji, S.; Porras, J. Mitigation Strategies against the Phishing Attacks: A Systematic Literature Review. *Comput. Secur.* **2023**, *132*, 103387. [CrossRef]
13. Goenka, R.; Chawla, M.; Tiwari, N. A Comprehensive Survey of Phishing: Mediums, Intended Targets, Attack and Defence Techniques and a Novel Taxonomy. *Int. J. Inf. Secur.* **2023**, 1–30. [CrossRef]
14. Febriyani, W.; Fathia, D.; Widjajarto, A.; Lubis, M. Security Awareness Strategy for Phishing Email Scams: A Case Study One of a Company in Singapore. *JOIV Int. J. Inform. Vis.* **2023**, *7*, 808–814. [CrossRef]
15. Liu, S.; Cui, B.; Hou, W. A Survey on Blockchain Abnormal Transaction Detection. In *Blockchain and Trustworthy Systems*; Communications in Computer and Information Science; Chen, J., Wen, B., Chen, T., Eds.; Springer Nature: Singapore, 2024; Volume 1896, pp. 211–225. ISBN 978-981-9981-00-7.
16. Wan, Y.; Xiao, F.; Zhang, D. Early-Stage Phishing Detection on the Ethereum Transaction Network. *Soft Comput.* **2023**, *27*, 3707–3719. [CrossRef]

17. Wang, J.; Chen, P.; Yu, S.; Xuan, Q. TSGN: Transaction Subgraph Networks for Identifying Ethereum Phishing Accounts. In *Blockchain and Trustworthy Systems*; Communications in Computer and Information Science; Dai, H.-N., Liu, X., Luo, D.X., Xiao, J., Chen, X., Eds.; Springer: Singapore, 2021; Volume 1490, pp. 187–200. ISBN 9789811679926.
18. Farrugia, S.; Ellul, J.; Azzopardi, G. Detection of Illicit Accounts over the Ethereum Blockchain. *Expert Syst. Appl.* **2020**, *150*, 113318. [CrossRef]
19. Chen, W.; Guo, X.; Chen, Z.; Zheng, Z.; Lu, Y. Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem. *IJCAI* **2020**, *7*, 4456–4462.
20. Yuan, Q.; Huang, B.; Zhang, J.; Wu, J.; Zhang, H.; Zhang, X. Detecting Phishing Scams on Ethereum Based on Transaction Records. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
21. Chen, L.; Peng, J.; Liu, Y.; Li, J.; Xie, F.; Zheng, Z. Phishing Scams Detection in Ethereum Transaction Network. *ACM Trans. Internet Technol.* **2021**, *21*, 1–16. [CrossRef]
22. Abdelhamid, N.; Ayesh, A.; Thabtah, F. Phishing Detection Based Associative Classification Data Mining. *Expert Syst. Appl.* **2014**, *41*, 5948–5959. [CrossRef]
23. Karim, A.; Shahroz, M.; Mustofa, K.; Belhaouari, S.B.; Joga, S.R.K. Phishing Detection System Through Hybrid Machine Learning Based on URL. *IEEE Access* **2023**, *11*, 36805–36822. [CrossRef]
24. Zouina, M.; Outtaj, B. A Novel Lightweight URL Phishing Detection System Using SVM and Similarity Index. *Hum. Cent. Comput. Inf. Sci.* **2017**, *7*, 17. [CrossRef]
25. Moghimi, M.; Varjani, A.Y. New Rule-Based Phishing Detection Method. *Expert Syst. Appl.* **2016**, *53*, 231–242. [CrossRef]
26. Wen, H.; Fang, J.; Wu, J.; Zheng, Z. Transaction-Based Hidden Strategies against General Phishing Detection Framework on Ethereum. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May 2021; pp. 1–5.
27. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13 August 2016; pp. 855–864.
28. Wu, J.; Yuan, Q.; Lin, D.; You, W.; Chen, W.; Chen, C.; Zheng, Z. Who Are the Phishers? Phishing Scam Detection on Ethereum via Network Embedding. *IEEE Trans. Syst. Man Cybern Syst.* **2022**, *52*, 1156–1166. [CrossRef]
29. Lin, D.; Wu, J.; Yuan, Q.; Zheng, Z. T-Edge: Temporal Weighted Multidigraph Embedding for Ethereum Transaction Network Analysis. *Front. Phys.* **2020**, *8*, 204. [CrossRef]
30. Lin, D.; Wu, J.; Yuan, Q.; Zheng, Z. Modeling and Understanding Ethereum Transaction Records via a Complex Network Approach. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2737–2741. [CrossRef]
31. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
32. Abu-El-Haija, S.; Kapoor, A.; Perozzi, B.; Lee, J. N-Gcn: Multi-Scale Graph Convolution for Semi-Supervised Node Classification. In Proceedings of the Uncertainty in Artificial Intelligence, Virtual, 3–6 August 2020; pp. 841–851.
33. Duan, X.; Yan, B.; Dong, A.; Zhang, L.; Yu, J. Phishing Frauds Detection Based on Graph Neural Network on Ethereum. In *Wireless Algorithms, Systems, and Applications*; Lecture Notes in Computer Science; Wang, L., Segal, M., Chen, J., Qiu, T., Eds.; Springer Nature: Cham, Switzerland, 2022; Volume 13471, pp. 351–363. ISBN 978-3-031-19207-4.
34. Zhou, X.; Yang, W.; Tian, X. Detecting Phishing Accounts on Ethereum Based on Transaction Records and EGAT. *Electronics* **2023**, *12*, 993. [CrossRef]
35. Kumar, P.; Gupta, G.P.; Tripathi, R. TP2SF: A Trustworthy Privacy-Preserving Secured Framework for Sustainable Smart Cities by Leveraging Blockchain and Machine Learning. *J. Syst. Archit.* **2021**, *115*, 101954. [CrossRef]
36. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A Highly Efficient Gradient Boosting Decision Tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3149–3157.
37. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online Learning of Social Representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24 August 2014; pp. 701–710.
38. He, J.; Qi, J.; Ramamohanarao, K. Timesan: A Time-Modulated Self-Attentive Network for next Point-of-Interest Recommendation. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
39. Li, S.; Wang, R.; Wu, H.; Zhong, S.; Xu, F. SIEGE: Self-Supervised Incremental Deep Graph Learning for Ethereum Phishing Scam Detection. In Proceedings of the 31st ACM International Conference on Multimedia, Ottawa, ON, Canada, 26 October 2023; pp. 8881–8890.
40. Sukparungsee, S.; Areepong, Y.; Taboran, R. Exponentially Weighted Moving Average—Moving Average Charts for Monitoring the Process Mean. *PLoS ONE* **2020**, *15*, e0228208. [CrossRef] [PubMed]