Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230812

A Fishing Detector with Attentional Mechanisms

Yanli WANG, Li WANG[1] and Shiwen SUN
Tianjing Key Laboratory of Intelligence Computing and Novel Software Techology, Tianjing University of Technology, Tianjin 300384, China

Abstract. The threat posed by phishing scams to Ethereum's security has grown significantly with the advancement of blockchain technology. As a result, the detection of phishing scams has emerged as one of the most prominent research areas in the field of blockchain. Most existing studies represent transaction information as a static subgraph and employ random walks to extract potential user features. However, real-world graphs often exhibit dynamic behavior and evolve over time. To address these challenges, we introduce a novel approach called Dynamic Weighted Node Classification (DWNC). In this approach, we partition transaction records into multiple temporal snapshots based on time. We then capture the structural and temporal characteristics of the nodes using the structural aggregation module and the time aggregation module, respectively. Finally, we leverage the learned features for classification purposes. The proposed DWNC method demonstrates superior performance in classification, as evidenced by its evaluation on nine benchmark and Ethereum datasets.

Keywords. Ethereum; Phishing detection; Node classification; Attention mechanism

1. Introduction

In recent years, blockchain [1] has gained widespread popularity in fields as diverse as finance, technology, and culture. Based on the current use of technology, blockchain has the potential to significantly impact the global economy. The most widely used is digital cryptocurrency [2][3]. However, as an indispensable part of the digital economy, digital cryptocurrency has become a hotbed for cybercrime due to its anonymity and decentralization. Ethereum has gained great popularity here and abroad as a cryptocurrency platform. This has made it a prime target for cybercrime [4][5][6][7], including phishing scams and Ponzi schemes. Phishing scams are one of the most widely used fraud methods, which pose a great threat to Ethereum's security. Therefore, detecting phishing scams on Ethereum has become a hot research topic.Our contribution can be summarized as follows:

1. When utilizing the Graph Attention Network (GAT) to capture node structural features, we incorporate the transaction frequency as the edge weight and integrate it into the structural aggregation module.

2. We analyze dynamic graphs by employing the GNN+RNN approach, which allows us to capture both the structural and temporal features of nodes.

[1] Corresponding Author: Li WANG, Tianjing Key Laboratory of Intelligence Computing and Novel Software Techology, Tianjing University of Technology, Tianjin 300384, China; Email: wltjut08@126.com

2. Related Works

[8] uses high-order transaction time to summarize the transaction history of relevant accounts to extract features. In [9], the features of illegal addresses are described in detail, [10] also considered the calling information of smart contracts. the method [11] uses the truncated random walk to obtain the local information of the node.[12] improved Deepwalk and proposed biased random walk. trans2vec is proposed in [13]. The T-edge algorithm proposed in [14] mostly improves the order of nodes in the random walk process. In [15], the target address and its surrounding transaction network are treated as a subgraph. The Graph2Vec[16] method embeds the transaction topology into the feature vector. Tsgn[17] maps the original transaction subgraph to a more complex edge subgraph.[18] splits the entire transaction graph into smaller graphs based on time stamps. [19] proposed an algorithm called TEGDetector. [20] is embedded in an unsupervised way by combining the graph convolution layer of the graph autoencoder to realize the classification of phishers by LightGBM[21].

3. Method

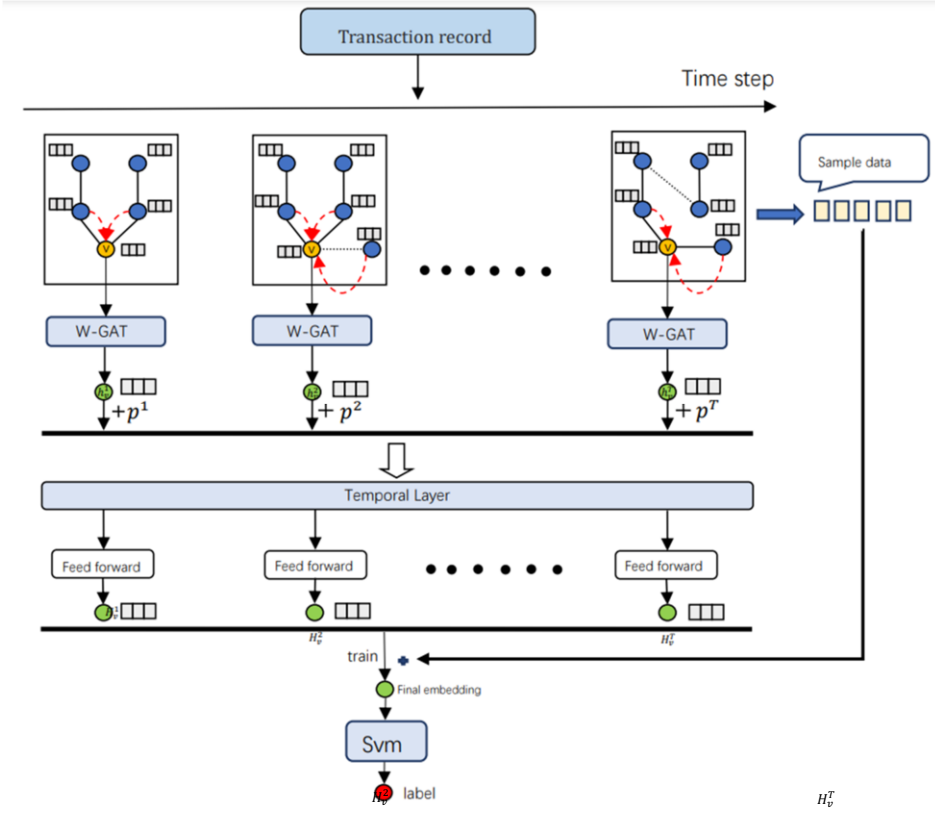

Figure 1. Architecture of DWNC model

As shown in Figure 1, the DWNC consists of two modules, which are structural aggregation module and temporal aggregation module. The input to the structural

aggregation module is a graph snapshot $\mathcal{G} \in \mathbb{G}$ and a series of node representations $\{x_\upsilon \in \mathbb{R}^D, \forall \upsilon \in \mathcal{V}\}$, where D is the dimension in which the node is embedded, and the initial input can be any dimension.The output after the structure aggregation module is $\{z_\upsilon \in \mathbb{R}^F, \forall \upsilon \in \mathcal{V}\}$, where F is the embedded dimension, our embedded dimension is 128 dimensions.and the new node attribute has a node-centered local neighborhood structure feature.The operation of the structural aggregation module is mainly divided into the following six steps:

a) One-hot encoding is used to represent the initial characteristics of the node.

b) Calculate the coefficient of attention.

$$e_{u\upsilon} = A_{u\upsilon} \cdot a^\top [W^s x_u || W^s x_\upsilon]) \tag{1}$$

Where $A_{u\upsilon}$ is the weight between node u and node v in the current snapshot.$W^s \in \mathbb{R}^{D\times F}$ is the weight transformation, which is shared by every node in the graph. $a \in \mathbb{R}^{2D}$ is the weight vector, and its job is to parameterize the attention function.

c) Use a LeakyRELU function to nonlinearize.

$$e_{u\upsilon} = \sigma(A_{u\upsilon} \cdot a^\top [W^s x_u || W^s x_\upsilon]) \tag{2}$$

d) Using a softmax operation on each neighboring node, the set of weight coefficients is obtained.

$$\alpha_{u\upsilon} = \frac{\exp(\sigma(A_{u\upsilon} \cdot a^\top [W^s x_u || W^s x_\upsilon]))}{\sum_{\omega \in \mathcal{N}_\upsilon} \exp(\sigma(A_{u\upsilon} \cdot a^\top [W^s x_u || W^s x_\upsilon]))} \tag{3}$$

Where $\mathcal{N}_\upsilon = \{u \in \mathcal{V}: (u, \upsilon) \in \xi\}$ is the local neighborhood set of node υ in snapshot $\mathcal{G}$; $\sigma(\cdot)$is the nonlinear activation function; $||$ is the concatenation operation.

e) The input features are weighted to get the features of the aggregated neighbor nodes.

$$z_\upsilon = \sigma(\sum_{u \in \mathcal{N}_\upsilon} \alpha_{u\upsilon} W^s x_u) \tag{4}$$

In order to prevent the model from focusing too much attention on its own location, we employ a multi-head attention mechanism.

$$h_\upsilon = \mathrm{Concat}(z_\upsilon^1, z_\upsilon^2, \ldots, z_\upsilon^H), \forall \upsilon \in V \tag{5}$$

H is the number of attention heads. h_υ is the output after multiple attention is applied to the current snapshot. After applying this layer to all snapshots, the final output is:

$$\{h_\upsilon^1, h_\upsilon^2, , , h_\upsilon^T\}, h_\upsilon^t \in \mathbb{R}^F \tag{6}$$

We need to add the absolute temporal position $\{p_\upsilon^1, p_\upsilon^2, , , p_\upsilon^T\}, p^t \in \mathbb{R}^F$ of all snapshots of node υ to the output of the structural aggregation module to make it have

a sense of order. Finally, a sequence of $\{h_\upsilon^1 + p^1, h_\upsilon^2 + p^2, , , h_\upsilon^T + p^T\}, h_\upsilon^t + p^t \in \mathbb{R}^F$ can be used as input for time aggregation and use a matrix $Y_\upsilon \in \mathbb{R}^{T\times F}$ to represent it. The output is a new representation sequence $Z_\upsilon \in \mathbb{R}^{T\times F'}$.

The temporal aggregation module is intended to obtain information about graph structure changes over time. The input representation of node υ at time-step t is expressed as y_υ^t.

the operation of the temporal aggregation module is defined as follows:

a) First multiply the input by the matrix $W_q \in \mathbb{R}^{F\times F'}$,$W_k \in \mathbb{R}^{F\times F'}$ and $W_\upsilon \in \mathbb{R}^{F\times F'}$ to get $Q = Y_\upsilon W_q$,$K = Y_\upsilon W_k$ and $V = Y_\upsilon W_\upsilon$. Where Q, K and V are query vector sequence, key vector sequence and value vector sequence respectively.

b) Compute the dot product of vectors.

$$e_\upsilon^{ij} = \left(\frac{(QK^T)_{ij}}{\sqrt{F'}} + M_{ij}\right) \tag{7}$$

Where $M \in R^{T\times T}$ is a mask matrix with each entry $M_{ij} \in \{-\infty, 0\}$. If $M_{ij} = -\infty$, the attention weight value is 0 after passing through the softmax layer and $\beta_\upsilon^{ij} = 0$ indicates that attention moves from time step i to j. To encode the temporal order, we define M as:

$$M_{ij} = \begin{cases} 0, & i \le j \\ -\infty, & \text{otherwise} \end{cases} \tag{8}$$

c) After normalizing the weight matrix, weighted average the value and output it.

$$\beta_\upsilon^{ij} = \frac{\exp(e_\upsilon^{ij})}{\sum_{k=1}^{T}\exp(e_\upsilon^{ik})}, Z_\upsilon = \beta_\upsilon(V) \tag{9}$$

Where β_υ^{ij} is the attention weight obtained after passing through the softmax layer, and $\beta_\upsilon \in R^{T\times T}$ is the attention weight matrix.

d) Finally, multiple attention heads are used to splice the final output:

$$H_\upsilon = \text{Concat}(Z_\upsilon^1, Z_\upsilon^2, \ldots, Z_\upsilon^H), \forall \upsilon \in V \tag{10}$$

Where H is the number of attention heads, $H_\upsilon \in \mathbb{R}^{T\times F'}$ is the output of temporal multi-head attention, and embedded dimension is also 128 dimensions.

We need to define an objective function.

$$L_\upsilon = \sum_{t=1}^{T}\sum_{u\in N_{walk}^t(\upsilon)} -\log(\sigma(< H_u^t, H_\upsilon^t >)) - w_n \cdot \sum_{u'\in P_n^t(\upsilon)} \log(1 - \sigma(< H_{u'}^t, H_\upsilon^t >)) \tag{11}$$

Where $P_n^t(\upsilon)$ is a negative sampling distribution for nove υ in at time t.

Finally we dichotomized the learned features.

4. Experiment

4.1. Datasets

We conducted performance evaluation of the algorithm using a real Ethereum dataset released on the Xblock platform. The dataset comprises 874 labeled phishing addresses and 886 labeled non-phishing addresses. We selected the tagged phishing and non-phishing nodes as the central nodes and extracted their neighboring nodes and the neighbors of their neighbors. This allowed us to construct a large-scale network encompassing 924,662 nodes and 3,887,136 edges, representing the transaction entities and their corresponding transactions. Specific information about the data is provided in Table 1.

Table 1. Datasets

Nodes	Edges	Time steps	Classes	Average degree	Fishing node	Non-fishing node
926442	1430221	16	2	3.09	874	886

4.2. Experimental Results

Table 2. Ethereum data experiment results

	Precision	Recall	F1-score	Accuracy
Deepwalk	**0.41**	**0.78**	**0.54**	**0.44**
Graphsage	**0.25**	**0.50**	**0.33**	**0.49**
GAT	**0.25**	**0.50**	**0.33**	**0.50**
GCN	**0.52**	**0.52**	**0.52**	**0.53**
Line	**0.60**	**0.60**	**0.60**	**0.60**
Struc2vec	**0.73**	**0.73**	**0.73**	**0.73**
Node2vec	**0.79**	**0.79**	**0.79**	**0.79**
Trans2vec	**0.79**	**0.79**	**0.79**	**0.79**
DWNC	**0.85**	**0.89**	**0.86**	**0.86**

To thoroughly assess the detection performance of our algorithm, we selected several network embedding methods as benchmarks for comparison. All comparison methods utilized the same address data as described in this paper. We conducted the experiments using the source code and parameters provided by the original author of each method.

In Table 2, our method outperforms other methods across various evaluation metrics. Furthermore, it can be observed that different random walk strategies have a certain influence on the results. Taking into account both time and transaction information during feature extraction leads to improved results. In this paper, we propose that the frequency of transactions reflects the closeness between the parties involved. Hence, our method incorporates transaction frequency as the weight and simultaneously learns the structural and temporal characteristics of the nodes. The aforementioned results demonstrate that our proposed approach exhibits superior performance compared to traditional random walk-based and graph-based neural network methods.

5. Conclusions

In this paper, we address the challenge of detecting phishing nodes within the Ethereum trading network. For the first time, we propose the utilization of transaction frequency as a weight to assess the proximity between the two parties involved in a transaction. To overcome the limitation of traditional approaches that cannot capture the temporal characteristics of nodes, we introduce a novel solution called DWNC, which combines the GAT and RNN methods. In DWNC, we treat the two sides of a transaction as nodes, and represent the transaction process as edges. By using transaction frequency as the weight, we construct a large weighted undirected graph. Subsequently, we leverage the structural aggregation module to extract the structural characteristics of each node, and employ the Temporal aggregation module to capture the time characteristics of the nodes. Finally, we classify the learned features to identify phishing nodes.

References

[1] M. Iansiti et al, The truth about blockchain, Harvard business review, 2017.
[2] H. Sira-Ramirez, Ethereum: A secure decentralised generalised transaction ledge, Ethereum project yellow paper, pp. 1–32, 2014.
[3] Y. Yuan et al, Blockchain and cryptocurrencies: Model,techniques,and applications, IEEE Transactions on Systems, Man, andCybernetics: Systems, pp. 1421–1428, 2018.
[4] M. Vasek et al, There's no free lunch, even using bitcoin:Tracking the popularity and profits of virtual currency scams, San Juan, Puerto Rico, USA, pp. 44–61, 2015.
[5] W. Chen et al, Detecting ponzi schemes on ethereum: Towards healthier blockchain technology, Lyon, France, pp. 1409–1418, 2018.
[6] M. Vasek et al, Analyzing the bitcoin ponzi scheme ecosystem, Nieuwpoort, West Flanders, Belgique, pp. 101–112, 2018.
[7] Y. Huang et al, Understanding (mis) behavior on the eosio blockchain, Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 4, pp. 1–28, 2020.
[8] Z. Yuan et al, Phishing detection on ethereum via learning represen-tation of transaction subgraphs, Dali, Yunnan, China, pp. 178–191, 2020.
[9] Q. Yuan, Detecting phishing scams on ethereum based on transaction records, Seville, Spain, pp. 1–5, 2020.
[10] Y. Lin et al, An evaluation of bitcoin address classification based on transaction history summa-rization, Seoul, Korea, pp. 302–310, 2019.
[11] B. Perozzi et al, Deepwalk: Online learning of social representations, New York, New York, USA, pp. 701–710, 2014.
[12] A. Grover et al, Node2vec: Scalable feature learning for networks, San Francisco, California, USA, pp. 855–864, 2016.
[13] J. Wu et al, Who are the phishers? phishing scam detection on ethereum via network embedding, IEEE Transactions on Systems, Man, andCybernetics: Systems, 2020.
[14] D. Lin et al, T-edge: Temporal weighted multidigraph embedding for ethereum transaction network analysis, Frontiers in Physics, vol. 8, no. 2, pp. 204, 2020.
[15] Z. Yuan et al, Phishing detection on Ethereum via learning representation of transaction subgraphs, Dali, Yunnan, China, pp. 178–191, 2020.
[16] A. Narayanan, graph2vec: Learning distributed representations of graphs , Halifax, Nova Scotia, Canada, pp. 21–23, 2017.
[17] J. Wang et al, Tsgn: Transaction subgraph networks for identifying ethereum phishing accounts, Shenzhen, China, pp. 187–200, 2021.
[18] M. Weber, Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics, arXiv preprint arXiv:1908.02591, 2019.
[19] D. Tam, Identifying illicit accounts in large scale e-payment networks–a graph representation learning approach, Macao, China, 2019.
[20] L. Chen, Phishing scams detection in ethereum transaction network, ACM Transactions on Internet Technology, vol. 21, no. 10, pp. 1–16, 2020.
[21] G. Ke et al, Lightgbm: A highly efficient gradient boosting decision tree, Los Angeles, California, USA, pp. 3146–3154, 2017