Internet Financial Fraud Detection Based on Graph Learning

Ranran Li, *Graduate Student Member, IEEE*, Zhaowei Liu, Yuanqing Ma, Dong Yang, and Shuaijie Sun

***Abstract*— The rapid development of information technology such as the Internet of Things, Big Data, artificial intelligence, and blockchain has changed the transaction mode of the financial industry and greatly improved the convenience of financial transactions, but it has also brought about new hidden frauds, which have caused huge losses to the development of Internet and IoT finance. As the size of financial transaction data continues to grow, traditional machine-learning models are increasingly difficult to use for financial fraud detection. Some graph-learning methods have been widely used for Internet financial fraud detection, however, these methods ignore the stronger structural homogeneity and cannot aggregate features for two structurally similar but distant nodes. To address this problem, in this article, we propose a graph-learning algorithm TA-Struc2Vec for Internet financial fraud detection, which can learn topological features and transaction amount features in a financial transaction network graph and represent them as low-dimensional dense vectors, allowing intelligent and efficient classification and prediction by training classifier models. The proposed method can improve the efficiency of Internet financial fraud detection with better Precision, *F*1-score, and AUC.**

***Index Terms*— Fraud detection, graph learning, Internet finance.**

I. Introduction

PEOPLE'S consumption habits have been dramatically altered by the rapid growth of information technologies such as the Internet of Things, Big Data, Artificial Intelligence, Blockchain, and so on [1]. Mobile payment, IoT financial services, and Internet financial wealth management have all permeated many facets of economic and social activity. Consumer finance sector growth in China has been strong since 2014, with a number of mobile e-commerce businesses entering the market through installment payments and modest loans, which has boosted the growth of associated industries. Customers in China have been able to enjoy the convenience of online shopping that allows them to pay for their purchases over time, thanks to the introduction of consumer credit-based internet financial services like Ant Financial's Huabei and Alipay's Alipay in China, JD.com's Jingdong Baitiao, and Tencent's WeBank's WeiLiDai.

As mobile and IoT financial payment systems have grown in popularity, so has the number of hidden fraud threats that may be exploited by criminals. There may be a fertile environment for fraudulent actions because of the complicated network's secrecy. Fraud risks are becoming increasingly difficult to control, and fraud cases are frequent, causing great property losses and seriously jeopardizing social harmony. The detection of fraud on the Internet is crucial to protect the interests of ordinary users and maintain the harmony of society.

Traditional machine learning has been used for financial fraud detection [2]–[5]. However, using traditional machine learning methods requires manual data processing and consumes a lot of time on feature engineering. This problem is becoming more and more serious as the volume of financial transaction data grows. While graph learning can learn the deep representation of node features directly in the topology of the network graph, which not only avoids time-consuming feature engineering, but also enhances the feature representation of nodes, which is beneficial for fraud detection in financial transactions.

At present, graph learning has been developing rapidly, and a large number of innovative graph neural networks (GNNs) [6]–[10] have been proposed and applied in various fields. The most central idea of GNNs is the message passing mechanism, which updates the feature information of the central node by the feature information of the neighboring nodes. In the present GNN-based financial fraud detection methods [11]–[14], different neighbor node selection strategies are proposed to improve the fraud detection effect. Among them, FA-GNN [11] proposes three priority policy neighbor filtering strategies based on transaction networks to filter out the important neighbor nodes. GraphConsis [12] proposes a model that combines contextual embedding and nodes, designs a probabilistic sampling method for selective sampling of neighboring nodes, and uses an attention mechanism for aggregation of node features. CARE-CNN [13] obtains the similarity between neighboring nodes by designing a layer of label-aware similarity measures and uses reinforcement learning to find the optimal number of neighbors. RioGNN [14] proposes an enhanced relationship-aware neighbor selection mechanism to identify the most similar neighbors from multiple relationship views for aggregation and uses a label-aware neural similarity

Manuscript received 27 January 2022; revised 23 May 2022; accepted 1 July 2022. Date of publication 15 July 2022; date of current version 31 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62072391 and Grant 62066013, in part by the Key Research and Development Program of Yantai under Grant 2020XDRH092, in part by the Doctoral Startup Fund of Yantai University under Grant JS19B77, and in part by the Shandong Key Laboratory of Marine Ecological Restoration under Grant 201920. *(Corresponding author: Zhaowei Liu.)*

The authors are with the School of Computer and Control Engineering, Yantai University, Yantai 264005, China, and also with the Shandong Marine Resources and Environment Research Institute, Yantai 264005, China (e-mail: lrr.ytu@gmail.com; lzw@ytu.edu.cn; erma0402@163.com; yangdong@s.ytu.edu.cn; 1216848095@qq.com).

Digital Object Identifier 10.1109/TCSS.2022.3189368

measure to avoid over-generalization of embeddings between different types of nodes.

However, all these methods aggregate central node features from neighboring nodes, so the final result learned is that nodes in close proximity to each other tend to have similar features. They both ignore the stronger structural homogeneity and cannot aggregate two nodes with similar structures but far away from each other, while in the actual Internet financial fraud scenario, two independent fraudulent users who are not associated in different regions are highly likely to have similar community structures. To address this problem, in this article, we propose an Internet financial fraud detection method, TA-Struc2Vec, which not only learns stronger structural homogeneity, but also further learns transaction amount homogeneity considering the important property of transaction amount in financial transaction networks. The financial transaction network is embedded in both structural homogeneity and transaction financial homogeneity, and the features of each user in the financial transaction network are extracted. Finally, machine-learning classification algorithms are used to classify the users in the financial transaction network to detect fraudulent users in the financial transaction network. Our contributions are as follows.

1) We propose a novel approach for Internet financial fraud detection, which overcomes the limitation that existing methods can only update the central node features from the adjacent nodes. And it can learn a stronger structure to aggregate the features of two extremely similar and distant nodes.
2) In our proposed method, we perform embedding learning for financial transaction networks from both structural homogeneity and transactional amount homogeneity to extract feature information of each user in the transaction network, which enhances the feature representation and facilitates fraudulent user detection in financial transactions.
3) By detecting phishing users in blockchain transactions, the experimental results prove that TA-Struc2Vec is indeed more competitive compared to other methods.

II. Background and Related Work

A. Graph Embedding

Graphics is an important form of data representation, appearing in a variety of real scenes. Effective graph analysis allows users to have a deeper understanding of the meaning behind the data, which is conducive to many useful applications such as node classification, node recommendation, and link prediction. However, most graphical analysis methods have the problems of a large amount of calculation and large space overhead. Graph embedding is an effective method to solve the problem of graph analysis. It converts the graphic data into a low-dimensional space, in which the graphic structure information and graphic attributes are preserved to the greatest extent. The methods of graph embedding are basically divided into three categories: methods based on factorization, methods based on random walks, and methods based on deep learning.

The graph embedding algorithm based on factor decomposition uses an adjacency matrix, Laplacian matrix, Katz similarity matrix, and so on to represent the connection of nodes and then decomposes the matrix to achieve the purpose of embedding. Models related to factorization include local linear embedding (LLE) [15], Laplacian feature mapping [16], graph representation for learning global structure information (GraRep) [17], and high-order proximity preserving embedding (HOPE) [18].

The graph embedding algorithm based on a random walk, combined with the idea of the Word2Vec algorithm, first collects the node sequence according to different strategies and treats the node sequence as a sentence. Then use the CBOW or Skip-gram model to learn to get the representation vector of the node. DeepWalk [19] uses a random walk strategy to obtain the node sequence. node2vec [20] is similar to DeepWalk, except that it uses a biased random walk, which is a tradeoff between breadth-first and depth-first graph search.

Based on the deep-learning method, the graph is combined with the neural network, and different GNN models are constructed to learn the characteristic information of the graph nodes. At present, GNN algorithms can be divided into four categories, namely cyclic GNN, convolution GNN, graph autoencoder, and spatio-temporal GNN. The cyclic GNN repeatedly applies the same set of parameters to the nodes in the graph to improve node representation. The work of literature [21], [22] is performed on directed acyclic graphs, the work of literature [23] is performed on undirected graphs, and SSE [24] improves the scalability and can be embedded in large graphs. The convolutional GNN is closely related to the cyclic GNN. The convolutional GNN does not use contraction constraints to iterate the node state but uses a fixed number of layers with different weights for each layer to solve the cyclic interdependence. Convolutional GNNs are divided into two categories: one is based on the frequency domain, and the other is based on the spatial domain. There are SDGNet [25], AGCN [8] in the frequency domain, and NN4G [9], CGMM [10] in the airspace. The graph autoencoder is a deep neural network structure that maps nodes to latent feature space and decodes graph information from the latent representation. Literature [26], [27] uses the structure of graphs, while literature [28], [29] considers both the structural information of the node and the characteristic information of the node. Spatio-temporal GNN captures the space and time dependence of graphs at the same time and has many applications in capturing the dynamics of graphs. Among them, the literature based on RNN [30], [31] and the literature based on CNN [32]–[34].

B. Graph-Based Fraud Detection

Graph learning is applied to different aspects of fraud detection, for example, disinformation detection in social media, fraudulent user detection in financial trading systems, Ponzi scheme detection, and so on.

For disinformation detection, FAHGT [35] uses a type-aware feature mapping mechanism to process heterogeneous graph data in order to identify false comments and solve the artifacts and inconsistencies in a uniform manner.

SAFER [36], in order to detect fake news on social platforms, models social networks in terms of several attributes such as the nature of the disseminated content and the content sharing behavior of users. DAGN-NN [37] can train the model with fewer samples and break through the limitations of traditional machine learning in the fake news detection task to achieve cross-domain fake news detection. FANG [38] is able to detect fake news on social platforms and can capture social context to high-fidelity representation with strong scalability.

For fraudulent user detection, FdGars [39] uses content features and behavioral features to analyze normal and malicious users in order to detect fraudulent comments in Internet systems and uses a graph convolutional neural network approach to classify users based on the analyzed information. SemiGNN [40], a semisupervised attention GNN, is proposed to address the lack of labeled data in the network and uses a hierarchical attention mechanism to perform fraudulent user detection by embedding learning from multiple views. GraphConsis [12] proposes a model that combines contextual embedding and nodes by designing a GNN framework and introducing the problem of contextual inconsistency to detect fraudulent users, filtering neighboring nodes, and using an attention mechanism for aggregation of node features. CARE-CNN [13] analyzes both feature artifacts and relational artifacts for fraudsters' artifacts and proposes a new model that enhances the GNN aggregation process and uses three unique modules to combat artifacts. RioGNN [14] proposes an enhanced relationship-aware neighbor selection mechanism that identifies the most similar neighbors from multiple relationship views for aggregation and uses a label-aware neural similarity metric to avoid over-generalization of embeddings between different types of nodes.

For Ponzi scheme detection, unlike the machine learning approaches [41], [42] used to detect vulnerabilities in Ethernet smart contracts, the graph-based approaches focus more on the construction of the Ethernet transaction network. Among them, [43] constructs a transaction network using transactions of target contracts and models the detection of Ponzi schemes as a node classification task, based on a graph convolutional network (GCN) for identification and detection. Jin *et al.* [44] propose a heterogeneous feature enhancement approach to capture heterogeneous information related to the use of line patterns.

III. Framework

In this section, we will first explain in detail how to use graph-learning methods to detect fraud. It mainly includes three parts: graph construction, graph embedding, and node classification. The overall framework is shown in Fig. 1. First, we will explain in detail how to construct a transaction graph based on user transaction information in an online financial platform. And then build an improved method based on Struc2Vec [45] for embedding online financial transaction graphs. This method not only considers the structure of the node, but also considers the transaction amount. Finally, the logistic algorithm is used to classify the transaction nodes in an online financial platform and identify malicious nodes.

A. Graph Construction

To use our algorithm to extract features of transaction nodes, we treat accounts and transactions in online financial transaction data as nodes and edges. More importantly, we use a time sliding window to obtain information about online financial transactions over a certain period to construct a weighted directed graph. In this graph, the edges between nodes represent the transactions, and the transaction amounts are used as the weights of the edges.

B. Graph Embedding

Generally, Struc2Vec can effectively learn potential representations of structural identity. In an online financial transaction graph, each transaction has its inherent attribute, that is, the amount. In order to be able to express the characteristics of online financial transaction nodes more accurately, we consider the transaction amount and the structure of the node at the same time. And designed an improved algorithm TA-Struc2Vec based on Struc2Vec. The core idea is that if two nodes have the same degree and the same transaction amount, then the two nodes are similar. If the two nodes have the same neighbor degree and the same transaction amount, then the similarity of these two nodes is higher than the former.

Let $G = (V, E)$ denote a directed weighted graph, V denote a set of nodes, and E denote a set of edges. We use $n = |V|$ to represent the number of nodes and $d*$ to represent the diameter of the graph, that is, the maximum distance between any two points in the graph. We use $R_k(u)$ to represent the set of nodes whose distance to node u is $k(k \geq 0)$. $D(S)$ represents the sequence formed after the nodes in a certain node set S are sorted from small to large according to the degree, where $S \subseteq V$.

$M(S)$ represents the sequence of node transaction amount in a certain node set S and the order of the nodes is consistent with $D(S)$. $s(S)$ represents a sequence formed by adding the degree and the transaction amount of the node according to a certain weight to the nodes in a certain node set S and sorting them from small to large according to the size of the obtained value

$$s(S) = \alpha(D(S)) + \beta(M(S)). \tag{1}$$

Among them, we set $\alpha/\beta = \lambda(\lambda \geq 1)$ for more consideration of the structure of nodes. Let $f_k(u, v)$ denote the similarity of nodes u, v, taking into account their K-Hop neighborhood, the initial value $f_{-1} = 0$. The $f_k(u, v)$ function is defined as follows:

$$f_k(u, v) = f_{k-1}(u, v) + g(s(R_k(u), \quad s(R_k(v))) \\ k \geq 0 \text{ and } |R_k(u)|, \quad |R_k(v)| > 0. \tag{2}$$

Among them, $R_k(u)$ represents the set of vertices with a distance of k from the vertex u and $g(D_1, D_2)$ represents the distance between two ordered queues. The calculation method uses the DTW algorithm. The DTW algorithm is usually used to calculate the similarity of two time series. It has a wide range of applications in the field of speech recognition and is

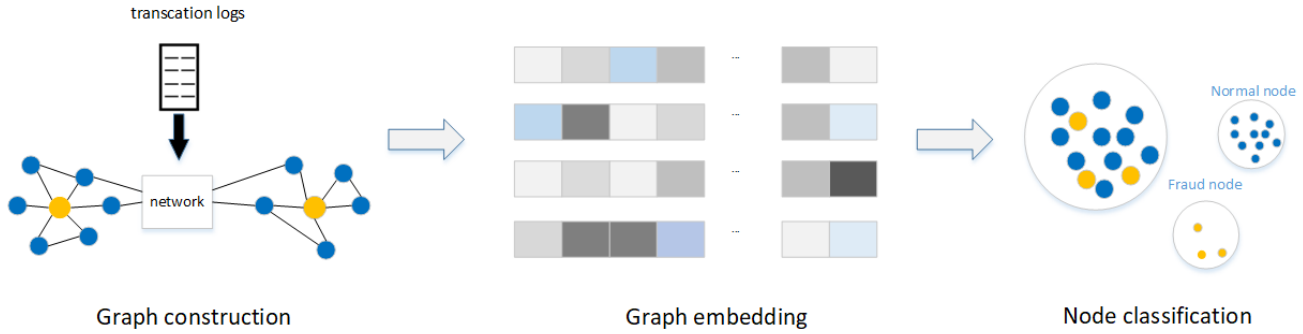

Fig. 1. Architecture of the proposed framework for Internet financial fraud detection.

defined as follows:

$$d(a,b) = \frac{\max(a,b)}{\min(a,b)} - 1. \tag{3}$$

Then construct a multilayer weighted graph to encode the structural similarity between nodes. Each layer is a complete graph with weights, and the weight of each edge is defined as follows:

$$w_k(u,v) = e^{-f_k(u,v)}, \quad k = 0, \ldots, k^*. \tag{4}$$

The layers are connected by directed edges. Specifically, for any node u_k of the kth layer, there are directed edges (u_k, u_{k-1}) and (u_k, u_{k+1}), The weights are

$$\begin{aligned} w(u_k, u_{k+1}) &= \log(L_k(u) + e), \quad k = 0, \ldots, k-1 \\ w(u_k, u_{k-1}) &= 1, \quad k = 1, \ldots, k. \end{aligned} \tag{5}$$

Among them, $L_k(u)$ represents the number of edges that point to the u node in the k layer whose weight is greater than the average weight of the layer. Defined as follows:

$$L_k(u) = \sum_{v \in V} 1(w_k(u,v) > \overline{w_k}) \tag{6}$$

where $\overline{w_k}$ represents the average value of all edge weights of the kth layer. $L_k(u)$ actually indicates how many nodes in the kth layer are similar to node u. If node u is similar to many nodes, it must be at a low level and there is too little information to consider. At this time, the value of $L_k(u)$ will be very high, and there will be a situation of $w(u_k, u_{k+1}) > w(u_k, u_{k-1})$. In this case, the node of this layer is not suitable for the context of the node, and you should consider jumping to a higher layer to find a suitable context. After constructing the multilayer weighted graph, use the random walk method in DeepWalk to obtain a similar node sequence of a node. In a certain layer, the probability of walking to other nodes is related to the weight of the edge, that is, the probability is obtained by normalizing the weight

$$P_k(u,v) = \frac{e^{-f_k(u,v)}}{Q_k(u)}. \tag{7}$$

The denominator is the normalization factor

$$Q_k(u) = \sum_{\substack{v \in V \\ v \neq u}} e^{-f_k(u,v)}. \tag{8}$$

Algorithm 1 $TA - Struc2Vec$ Algorithm

Input: The transaction graph G, embedding dimension d, walk length l, window size o, bias parameters α and β, the order of the neighbor node K.
Output: Embedding features Z.

```
for i =0 to K do
  for each node v ∈ V do
    Get the neighbor node set S of node v;
    Get s(S) with Equation 1;
  end for
  for each node u ∈ V do
    for each node v ∈ V do
      Calculate the similarity of nodes u, v with Equation 2;
      Calculate the weight of nodes u, v with Equation 4;
    end for
  end for
  Construct a weighted complete graph of the current layer;
end for
for i =0 to K do
  Calculate the weights between layers with Equation 5 and
  Equation 6;
end for
Calculate the probability of walking to other nodes with
Equation 7 and Equation 9;
Generate a node sequence of length l by random walk with
bias;
Z = Word2Vec(node sequence, d, o);
return Z.
```

If the layer is adjusted, it is divided into two directions, the probability is also related to the weight of the edge

$$\begin{aligned} P_k(u_k, u_{k+1}) &= \frac{w(u_k, u_{k+1})}{(w(u_k, u_{k+1}) + w(u_k, u_{k-1})} \\ P_k(u_k, u_{k-1}) &= 1 - P_k(u_k, u_{k+1}). \end{aligned} \tag{9}$$

Finally, according to the node sequence obtained by sampling, the Skip-gram model in Word2Vec is used for feature extraction. The pseudocode of the proposed TA-Struc2Vec is listed in Algorithm 1.

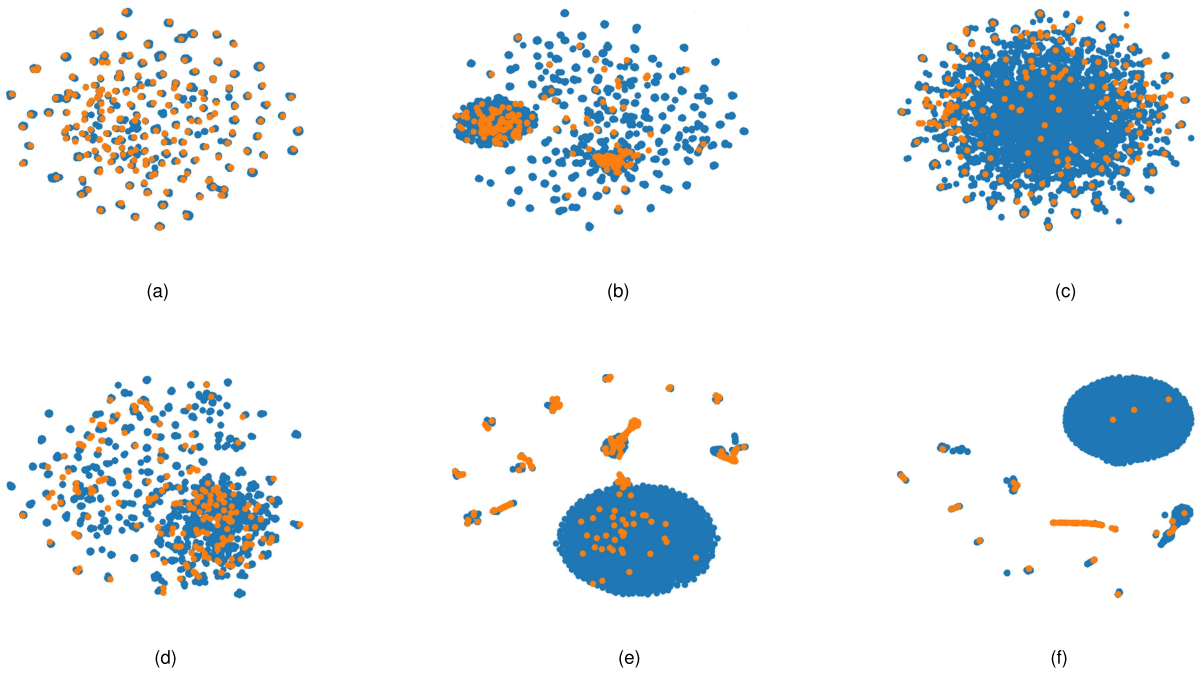

Fig. 2. Visualization of nodes in 2-D; the yellow dots represent phishing nodes and the blue dots represent normal nodes. (a) DeepWalk. (b) GraphConsis. (c) CARE-GNN. (d) RioGNN. (e) Struc2Vce. (f) TA-Struc2Vec.

C. Node Classification

After the feature extraction is completed by the TA-Struc2Vec graph embedding method, we use machine-learning methods to classify the nodes. In this article, we mainly classify phishing nodes and ordinary nodes, which is a two-class classification problem. Therefore, we can use the logistic regression algorithm or the SVM algorithm to analyze and identify the nodes of the online financial transaction network and detect malicious nodes among them. When multiple categories are involved, we can use multiple classification algorithms [46] for classification and recognition.

IV. Experiment

In this section, we first describe the dataset, then compare TA-Struc2Vec with other algorithms, and perform visualization and parametric analysis.

A. Dataset Description

Thanks to the work of the paper [47], a large amount of Ethereum transaction data has been collected through the interface provided by Etherscan. They started from the phishing nodes that have been published in Ethersacn and crawled a large-scale Ethereum transaction network from Ethereum transaction records through a second-order breadth search. The Ethereum transaction network has 2 973 382 nodes and 13 551 214 edges, of which 1157 are marked as phishing nodes. First, we randomly obtain 200 marked phishing nodes and 200 normal nodes and use these nodes as central nodes. Then use the K-level sampling method to obtain 400 subgraphs and set k-in $= 2$ and k-out $= 2$. Finally, connect these subgraphs to obtain an Ethereum transaction network with 6500 nodes.

B. Experimental Settings

1) Baselines: We comapare TA-Struc2Vec with seven baselines.

DeepWalk [19] uses a random walk method, starting from a node, obtaining a certain long sequence of nodes, and then stopping. Then combine the ideas in Word2Vec, use the Skip-Gram model or CBOW model training, and extract the embedding vector of each node.

node2vec [20] uses a breadth-first algorithm and depth-first algorithm to replace the random walk in DeepWalk, which enriches the strategy of sample collection.

Line [48] defines the first-order similarity and the second-order similarity, where the first-order similarity refers to the connection strength of the point pairs in the original space. The first-order similarity is used to describe the global characteristics of the graph. Second-order similarity refers to the strength of the connection between a single point and its neighbors. This feature is local and limited to describing the relationship between the center point and its neighbors.

Struc2Vec [45] proposes a method to measure the similarity of nodes. Two nodes are similar if they have the same degree, and they are more similar if the neighbors of these two nodes also have the same degree.

GraphConsis [12] proposes a model that combines contextual embedding and nodes for selective sampling by generating

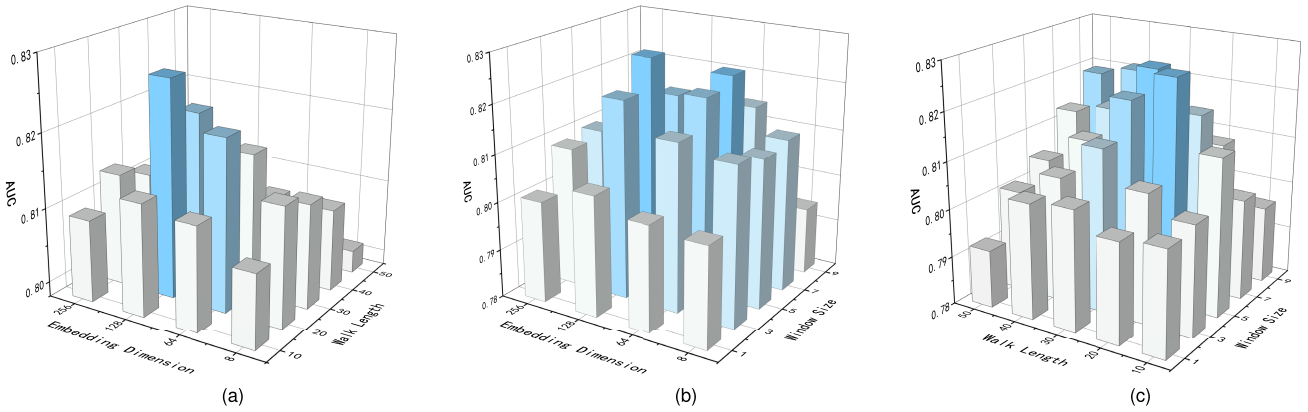

Fig. 3. Sensitivity analysis among embedding dimension d, walk length l, and window size o. (a) d with l. (b) d with o. (c) l with o.

corresponding sampling probabilities and utilizes an attention mechanism for aggregation of node features.

CARE-CNN [13] obtains the similarity between neighboring nodes by designing a layer of label-aware similarity measures and uses reinforcement learning to find the optimal number of neighbors.

RioGNN [14] proposes an enhanced relationship-aware neighbor selection mechanism to identify the most similar neighbors from multiple relationship views for aggregation and uses a label-aware neural similarity measure to avoid overgeneralization of embeddings between different types of nodes.

2) Experimental Design: We use graph learning algorithms to embed the Ethernet transaction network, extract the features of each user, and use logistic regression algorithms to classify phishing users and normal users to detect phishing users. Considering the imbalance of data, we use Precision, Recall, $F1$-score, and AUC as the evaluation metrics for the effectiveness of our experiments. And we compare the experimental effects under different machine learning to verify the effectiveness of the logistic regression algorithm.

C. Experimental Results

In the experiment, we set the training ratio to 80%, the dimension of the embedding vector of each node to 128, the window size to 5, and the sequence length of each node to 10. Among them, α is set to 0.2, and β is set to 0.02, and we use the logistic regression algorithm to classify nodes.

The experimental results are shown in Table I. TA-Struc2Vec outperforms RioGNN in Precision, F1-score, and AUC metrics and outperforms other algorithms in all metrics, which proves the effectiveness of TA-Struc2Vec in fraudulent user detection. Furthermore, we demonstrate the effectiveness of the logistic regression algorithm by comparing the classification effect of the logistic regression algorithm with SVM, decision tree, and random forest algorithms using the features extracted by TA-Struc2Vec. The comparison results are shown in Table II, and it can be seen that the logistic regression algorithm is slightly inferior to SVM except for Precision, which is due to other algorithms.

TABLE I
OVERALL PERFORMANCE COMPARISON

Algorithm	Precision	Recall	F1-score	AUC
DeepWalk	0.382	0.143	0.206	0.496
node2vec	0.634	0.314	0.420	0.734
Struc2Vec	0.811	0.627	0.715	0.801
Line	0.557	0.183	0.240	0.573
GraphCosis	0.816	0.569	0.671	0.780
CARE-GNN	0.753	0.604	0.670	0.795
RioGNN	0.801	**0.651**	0.717	0.819
$TA-Struc2Vec$	**0.852**	0.641	**0.741**	**0.827**

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT CLASSIFIERS

Algorithm	Precision	Recall	F1-score	AUC
Decision Tree	0.779	0.616	0.688	0.801
Random Forest	0.811	0.627	0.715	0.802
SVM	**0.859**	0.639	0.733	0.815
Logistic Regression	0.852	**0.641**	**0.741**	**0.827**

D. Visualization

Our downstream task uses machine-learning algorithms for the classification of phishing nodes and normal nodes using machine-learning algorithms, which are very dependent on the features of the nodes, and the more different the two features are, the easier it is to classify them. We used the t-SNE [49] nonlinear dimensionality reduction algorithm for visualization, which reduces the high-dimensional features of nodes to a 2-D space, where nodes with more similar features tend to cluster together, so that we can directly observe the effect of feature extraction by graph learning methods. When the nodes with the same color are more obviously aggregated and the nodes with different colors are more obviously separated, the extracted features are better and facilitate the classification by machine learning algorithms.

To show more clearly the effectiveness of TA-Struc2Vec, we map the 128-dimensional node features extracted by DeepWalk, GraphCosis, CARE-GNN, RioGNN, and Struc2Vec to 2-D space, respectively. The visualized results are shown in Fig. 2. We can see that phishing nodes are mixed with normal nodes in the visualization of GraphCosis, CARE-GNN, and

RioGNN algorithms. In contrast, the TA-Struc2Vec algorithm has better results and clearly separates the phishing nodes from the normal nodes.

E. Parameter Analysis

Now, we try to find the effect of hyperparameters on TA-Struc2Vec, including embedding dimension d, walk length l, and window size o. First, we evaluate the effect of d and l by setting o to 5, the range of d to $\{8, 64, 128, 256\}$, and the range of l to $\{10, 20, 30, 40, 50\}$. Then the effect of d and o is evaluated by setting l to 20, the range of d is $\{8, 64, 128, 256\}$, and the range of o is $\{1, 3, 5, 7, 9\}$. Finally, the effect of l and o is evaluated by setting d to 128, the range of l to $\{10, 20, 30, 40, 50\}$, and the range of o to $\{1, 3, 5, 7, 9\}$. The other parameters were set to optimal. To show the effect of different parameter values, we take AUC in this study. The experimental effect is shown in Fig. 3, where we can see that AUC is stronger when d is between 64 and 128, l is between 20 and 40, and o is between 5 and 7.

V. Conclusion and Future Work

In this article, we propose a graph-learning method TA-Struc2Vec for fraudulent user detection in Internet finance, which can learn the embedding of financial transaction networks in terms of both structural homogeneity and transaction amount homogeneity and extract the feature information of each user in the transaction network. It overcomes the limitation that existing methods can only update the central node features from neighboring nodes, enhances the feature representation, and facilitates the detection of fraudulent users in financial transactions. Experiments on the real-world dataset show that TA-Struc2Vec outperforms other existing methods for fraudulent user detection in Internet finance. In future work, the algorithm will be improved and implemented in conjunction with spatio-temporal properties to effectively learn the features of newly generated vertices in a dynamic network graph to achieve better financial fraud detection.

References

[1] U. Paschen, C. Pitt, and J. Kietzmann, "Artificial intelligence: Building blocks and an innovation typology," *Bus. Horizons*, vol. 63, no. 2, pp. 147–155, Mar. 2020.

[2] Y. Han, S. Yao, T. Wen, Z. Tian, C. Wang, and Z. Gu, "Detection and analysis of credit card application fraud using machine learning algorithms," *J. Phys., Conf.*, vol. 1693, no. 1, Dec. 2020, Art. no. 012064.

[3] B. Stojanović *et al.*, "Follow the trail: Machine learning for fraud detection in fintech applications," *Sensors*, vol. 21, no. 5, p. 1594, Feb. 2021.

[4] I. Sadgali, N. Sael, and F. Benabbou, "Performance of machine learning techniques in the detection of financial frauds," *Proc. Comput. Sci.*, vol. 148, pp. 45–54, Jan. 2019.

[5] D. P. Foster and R. A. Stine, "Variable selection in data mining: Building a predictive model for bankruptcy," *J. Amer. Stat. Assoc.*, vol. 99, no. 466, pp. 303–313, Jun. 2004.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[7] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 499–508.

[8] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 1–8.

[9] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 498–511, Mar. 2009.

[10] D. Bacciu, F. Errica, and A. Micheli, "Contextual graph Markov model: A deep and generative approach to graph processing," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 294–303.

[11] J. Liu, J. Zheng, J. Wu, and Z. Zheng, "FA-GNN: Filter and augment graph neural networks for account classification in ethereum," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2579–2588, Jul. 2022.

[12] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1569–1572.

[13] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 315–324.

[14] H. Peng, R. Zhang, Y. Dou, R. Yang, J. Zhang, and P. S. Yu, "Reinforced neighborhood selection guided multi-relational graph neural networks," *ACM Trans. Inf. Syst.*, vol. 40, no. 4, pp. 1–46, Oct. 2022.

[15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[16] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001, pp. 1–7.

[17] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 891–900.

[18] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1105–1114.

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

[20] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[21] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 714–735, May 1997.

[22] A. Micheli, D. Sona, and A. Sperduti, "Contextual processing of structured data by recursive cascade correlation," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1396–1410, Nov. 2004.

[23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2008.

[24] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steady-states of iterative algorithms over graphs," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1106–1114.

[25] Z. Zhang, Y. Li, H. Dong, H. Gao, Y. Jin, and W. Wang, "Spectral-based directed graph network for malware detection," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 957–970, Apr. 2021.

[26] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1225–1234.

[27] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 1–8.

[28] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.

[29] W. Wang *et al.*, "HGATE: Heterogeneous graph attention auto-encoders," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 28, 2021, doi: 10.1109/TKDE.2021.3138788.

[30] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," 2018, *arXiv:1803.07294*.

[31] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.*, 2018, pp. 362–373.

[32] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.

[33] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[34] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–10.
[35] S. Tang, L. Jin, and F. Cheng, "Fraud detection in online product review systems via heterogeneous graph transformer," *IEEE Access*, vol. 9, pp. 167364–167373, 2021.
[36] S. Chandra, P. Mishra, H. Yannakoudakis, M. Nimishakavi, M. Saeidi, and E. Shutova, "Graph-based modeling of online communities for fake news detection," 2020, *arXiv:2008.06274*.
[37] H. Yuan, J. Zheng, Q. Ye, Y. Qian, and Y. Zhang, "Improving fake news detection with domain-adversarial and graph-attention neural network," *Decis. Support Syst.*, vol. 151, Dec. 2021, Art. no. 113633.
[38] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan, "FANG: Leveraging social context for fake news detection using graph representation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1165–1174.
[39] J. Wang, R. Wen, C. Wu, Y. Huang, and J. Xion, "FdGars: Fraudster detection via graph convolutional networks in online app review system," in *Proc. World Wide Web Conf.*, 2019, pp. 310–316.
[40] D. Wang *et al.*, "A semi-supervised graph attentive network for financial fraud detection," in *Proc. 2019 IEEE Int. Conf. Data Mining*, Nov. 2019, pp. 598–607.
[41] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "ContractWard: Automated vulnerability detection models for ethereum smart contracts," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1133–1144, Apr. 2021.
[42] J. Song, H. He, Z. Lv, C. Su, G. Xu, and W. Wang, "An efficient vulnerability detection model for ethereum smart contracts," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2019, pp. 433–442.
[43] S. Yu, J. Jin, Y. Xie, J. Shen, and Q. Xuan, "Ponzi scheme detection in ethereum transaction network," in *Proc. Int. Conf. Blockchain Trustworthy Syst.*, 2021, pp. 175–186.
[44] C. Jin, J. Jin, J. Zhou, J. Wu, and Q. Xuan, "Heterogeneous feature augmentation for Ponzi detection in ethereum," 2022, *arXiv:2204.08916*.
[45] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "Struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 385–394.
[46] J. Li, Y. Liu, R. Yin, H. Zhang, L. Ding, and W. Wang, "Multi-class learning: From theory to algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1593–1602.
[47] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, and Z. Zheng, "Phishing scams detection in ethereum transaction network," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–16, Feb. 2020.
[48] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

Ranran Li (Graduate Student Member, IEEE) is currently pursuing the M.Sc. degree with the School of Computer and Control Engineering, Yantai University, Yantai, China.

His current research interests include graph representation learning and anomaly detection in internet platforms.

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

He is currently an Associate Professor with Yantai University, Yantai, China. His main research interests are distributed artificial intelligence, machine learning, virtual reality, and collaboration with proposed software companies in directions, including digital twin and industrial meta-universe related research.

Dr. Liu is a member of the China Computer Federation (CCF).

Yuanqing Ma was born in 1979.

He is currently a Senior Engineer with the Shandong Institute of Marine Resources and Environment, Yantai, China. He has published more than 30 articles and six monographs edited or co-edited. His main research interest is the intelligent monitoring of the marine ecological environment.

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

His current research interests include machine learning with graphs and graph topology learning.

Shuaijie Sun received the B.Eng. degree in computer control and engineering from Yantai University, Yantai, Shandong, China, in 2022. He will pursue the master's degree in international material flow management with the Trier College of Sustainable Technology, Yantai University.