Phishing Detection on Ethereum via Learning Representation of Transaction Subgraphs

Zihao Yuan[1,2], Qi Yuan[1,2], and Jiajing Wu[1,2(✉)]

[1] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China
wujiajing@mail.sysu.edu.cn
[2] National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510006, China

Abstract. With the widespread application of blockchain in the financial field, blockchain security also faces huge challenges brought by cybercrimes, especially phishing scams. It forces us to explore more efficient countermeasures and perspectives for better solution. Since graph modeling provides rich information for possible downstream tasks, we use a surrounding graph to model the transaction data of a target address, aiming to analyze the identity of an address by defining its transaction pattern on a high-level structure. In this paper, we propose a graph-based classification framework on Ethereum. Firstly we collect the transaction records of some verified phishing addresses and the same number of normal addresses. Secondly we form a set of subgraphs, each of which contains a target address and its surrounding transaction network in order to represent the original address on graph-level. Lastly, based on the analysis of the Ether flow of the phishing scam cycle, we propose an improved Graph2Vec, and make classification prediction on the subgraphs we built. The experimental results show that our framework has achieved a great competitiveness in the final classification task, which also indicate the potential value of phishing detection on Ethereum via learning the representation of transaction network.

GNNs used in context of providing information for downstream prediction (i.e. by ML models)

Keywords: Blockchain · Ethereum · Phishing detection · Graph classification

1 Introduction

Blockchain is a distributed ledger technology that implements trusted intermediary transactions in an environment of mutual distrust [19]. It could be described as a distributed and trusted database maintained by a peer-to-peer network based on a creative consensus mechanism. Blockchain technology has the characteristics of anti-counterfeiting, immutability, and the ability to expand application scenarios through smart contracts, with which it is regarded as the

Z. Zheng et al. (Eds.): BlockSys 2020, CCIS 1267, pp. 178–191, 2020.
https://doi.org/10.1007/978-981-15-9213-3_14

next generation of disruptive core technology. With the underlying support of blockchain technology, blockchain platforms such as Bitcoin and Ethereum have also taken this opportunity to flourish and be famous around the world as new digital currency trading platforms [17]. Among them, Ethereum is the most lag blockchain platform that provides a Turing-complete language to supports smart contracts, which will be executed in the Ethereum virtual machine (EVM). Due to the adaptability of the above characteristics of Ethereum, it has gained great development support in the field of economics and finance, and become a widely used platform for financial applications [9].

With the increasing prosperity of e-commerce, more and more people trade goods or services online, which also gives phishing scams more opportunities. Phishing scams refer to scammers illegally forge official websites or contact information to obtain users' private information, such as user names, passwords, and address numbers, for further gains [10]. The usual method is to send the victim a fake email that appears to come from the official, informing him to click on the link to modify relevant information, and the link actually points to a fake webpage for fraud, on which the information victim leaked will finally be obtained by scammers. In real cases, these fake information are generally spread through emails, Google ads, forums, and chat apps, and due to the low cost, they have great lethality in most of the time [1]. Nowadays, phishing scams have become one of the most widely used scams on Ethereum, which forces us to attach more attention to research in this field for adopting correct and efficient countermeasures [5,13,17].

In order to detect phishing addresses, we usually crawl their transaction records to extract distinguishing features [11]. Then we can form a directed graph according to the corresponding transaction records of those addresses. Note that in the Ethereum transaction network, each node represents an Ethereum address, while each edge indicates a certain Ether transfer between the addresses. Different types of addresses always exhibit different characteristics in transaction patterns, which could also be reflected in the network structure. And conversely, we can also evaluate the label of users through the relevant network information of corresponding addresses. In view of the development of research in the field of graph classification, we are increasingly able to mine valuable information via learning the embedding of network. Thus we use a surrounding transaction network to represent the target address, expecting to analyze identity of the address by defining its transaction pattern on a high-level structure. On the one hand, comparing with forming a large-scale network connected by all of those target addresses, using a second-order transaction subgraph to represent the original address costs less. We could easily extract the transaction records of a certain address on the Ethereum and build up a corresponding transaction network for it. On the other hand, through the graph embedding algorithm, the second-order transaction subgraph could also be used to extract meaningful information as representation features from the surrounding transaction network.

Summarily, from the perspective of transaction network, our proposed detection strategy could have the following advantages:

1. Representing the addresses with the second-order transaction subgraphs costs little and could analyze the characteristics of transaction patterns for identifying in an effective and direct way.
2. Using graph classification methods for anomaly detection on Ethereum is a new perspective of related research, which could extract high-level information as features for classification from the transaction network. It makes up for the lack of information from a perspective of network in other methods.
3. Considering the analysis of the transaction network on Ethereum, the proposed framework is designed specifically for the phishing detection issues, in order to boost its competitiveness in final classification performance.

One of first to predict on subgraphs?

In this paper, we propose a novel framework to detect phishing scams on Ethereum from the perspective of the transaction network. First, we extract the labeled phishing addresses and corresponding transaction records from an authorized platform. According to the collected transaction records, we build several corresponding subgraphs. Second, we mainly focus on extracting features from the corresponding subgraphs. According to the analysis of the transaction pattern of phishing scammers, we adopt an improved graph embedding method to extract the latent features of the subgraphs as addresses' features for subsequent phishing classification. Finally, we adopt SVM(Support Vector Machine) to distinguish whether the address is a phishing scammer.

The following chapter of the paper is organized as follows. Section 2 provides related work about phishing scam detection on Ethereum and anomaly detection by graph classification methods. In Sect. 3, the proposed framework is discussed in detail. Sections 4 summarize experimental results and the analysis of them. Lastly, we conclude our work in Sect. 5.

2 Related Work

2.1 Phishing Detection

Since phishing scams have received much research attention, traditional phishing detection methods based on virtual similarity, association rule learning, and support vector machines have been proposed and used in various fields [1]. With continuous research, a variety of features related to phishing scammers have also been confirmed and summarized, such as the source code of phishing websites [20], the characteristics of link URLs [14], and the CSS characteristics of websites [12]. And as traditional phishing scams obtain the victims' private information mainly through phishing emails and websites, the above traditional detection methods are basically based on the detection of phishing content.

However, phishing frauds on Ethereum not only obtain key information via phishing websites, but also spread phishing addresses via emails, chatting apps and other ways [10]. Thus due to the diversity of phishing approaches on Ethereum, current traditional detection methods are unsuitable to be applied here directly.

Analysis of phishing scams on Ethereum can be conducted from its whole life cycle and the difference of behaviors between confirmed phishing addresses and normal addresses. Considering that a complete fraud attack is a dynamically changing process, we can build multi-angle features based on the different behavioral characteristics displayed in each stage [1]. After scientific and rigorous quantitative analysis of these features, the most effective and sufficient key indicators are extracted to help the subsequent models to classify the correct labels of users. In the previous work [16], we built the classification model by feature engineering from the perspectives of address features, network features, and Ether flow characteristics. Among them, through the selection of indicators that can fully show the specificity of phishing addresses, the model has better performance in the final downstream tasks. And our work is more focused on the transaction network, trying to explore the potential information available for classification tasks from a new perspective.

Previous work uses feature engineering rather than embedded representations?

2.2 Development and Applications of Graph Classification

Recently, we have witnessed an increasing number of applications involving objects with structural relationships, for which graph-structured data become more and more ubiquitous in many domains such as social networks [3], cybersecurity [6], bio- and chemo-informatics [7]. In such applications, graph is proved to be a natural and powerful tool for modeling and capturing dependency relationships between objects in the network structure. Different from conventional methods, we focus on using the entire-network structural relationships to explore information by graph classification algorithms, instead of representing data in a feature-value vector format directly. In other word, in the processing of this kind of graph-structured data, model based on a high-level structure may acquire additional valuable information missed by regular methods.

With the rapid development of graph applications and more challenges from different fields, the research field of graph classification algorithms has gradually become more diversified. The current graph classification methods can be summarized into the following categories: graph kernel [15], deep learning [4], and factorization [2].

The cases of different application scenarios are also constantly proving that the embedded representation learned from network structure will obtain a better performance than handcrafted features. While the transaction network is also a typical graph analytics scenario, it is also suitable to explore the information on it from the perspective of high-level structure. According to the above, we decide to integrate ideas of graph classification into solving phishing detection problems on Ethereum.

3 Proposed Model

3.1 Problem Definition

Given a set of addresses on Ethereum $\mathbb{A} = \{A_1, A_2, ...\}$, we build a set of representation transaction graphs for each target address $\mathbb{G} = \{G_{A_1}, G_{A_2}, ...\}$.

$G_{A_x} = (V, E, Y)$ is a transaction network centered on the target address A_x, where V represents the set of addresses that are transaction neighbors of the center address, E represents the set of transactions between those addresses, with $Y \in \mathbb{R}^{|\mathbb{A}|\times|\gamma|}$ where γ is the set of labels of the target addresses and the corresponding graphs. For the scenario of phishing address identification, γ includes two kinds of labels that $+1$ for phishing address and -1 for normal address. Our goal is to predict the missing values in γ. We intend to learn the embeddings for all the representation transaction graphs $X \in \mathbb{R}^{|V|\times d}$, where d represents the number of embedding dimension. And these embeddings will be used as features of addresses for the downstream phishing detection task (Fig. 1).

?? Is embedding dimension a hyperparameter for size of latent space ??

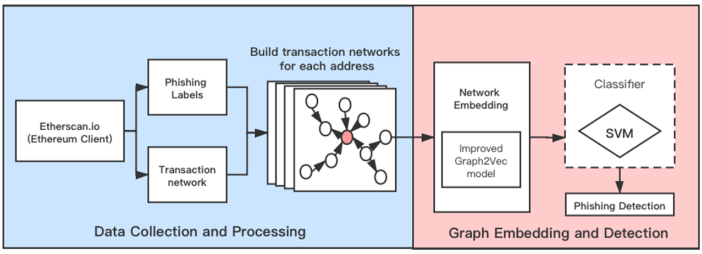

Fig. 1. Our proposed framework.

3.2 Data Collection and Processing

Because of the openness of public blockchain, each client on Ethereum is able to obtain all the transaction records, through which we could get the Ethereum dataset for model transaction network. In order to detect phishing scams, we need to get enough sufficient verified phishing addresses as our positive samples. As a block explorer and analytics platform for Ethereum, *Etherscan*[1] provide a growing list of phishing addresses labeled by the serious audit based on a majority of reports on the platform. We obtain 1660 verified phishing addresses which were reported in the list before October 17th, 2019 [18]. And then we randomly choose the same number of normal addresses to construct the sample list.

According to the list of addresses, we crawl the transaction network information for each target address through APIs offered by Etherscan. The transaction network information here consists of first-order transaction information between the target address and their first-order transaction neighbors, and the second-order transaction information that between first-order transaction neighbors and their next-order transaction neighbors. Note that the specific transaction records include: the address that initiated and accepted the transaction, the transaction timestamp, and the transaction amount. Based on the obtained information,

[1] https://etherscan.io/.

we can construct a second-order transaction subgraph centered on the target address. In this graph, each node represents a transaction neighbor of central address. A directed edge represents an Ether transfer transaction from the initial address to the recipient address. In the following work, we will use this kind of transaction subgraph to represent the original address.

Motivation for using second-order tx subgraphs

Addresses with various functions in a transaction network may exhibit different patterns with their neighbors, and thus behave differently in terms of network structure. So we assume that the embedding of second-order transaction subgraph of the target address retains the feature information of transaction behavior pattern. This is also the main motivation for us to use the graph classification algorithm for phishing detection in our proposed model.

At the same time, we need to note that there is often a lot of redundant data in the Ethereum transaction records, i.e. a considerable number of addresses have little value for our research target because of their extreme situation. For example, addresses act as wallet interact with other addresses frequently, for which their numbers of neighbor addresses are too large for analysis of transaction network. This kind of data increases the burden of data processing, and even cause unexpected deviations to the results of model training. Thus it is necessary to set standards to clean the obtained dataset. According to previous work, we consider the addresses with the number of transactions less than 10 as inactive addresses and the addresses with the number more than 300 as overactive addresses, filtering them after transaction records collecting [16].

3.3 Improved Graph Classification Algorithm

Add Graph2Vec to reading list

Then we focus on how to analyze and process the extracted transaction network of the target addresses. Firstly, we introduce the key tool for processing from the field of graph classification, named Graph2Vec, which has shown great improvement in classification over substructure embedding methods. Graph2Vec is inspired by Doc2Vec, an extension to Word2Vec algorithm in NLP (Natural Language Processing) direction. Doc2Vec uses a derived model of skipgram named paragraph vector-distributed bag of words (PV-DBOW) to learn representations of word sequences [2]. Likewise, Graph2Vec learns the embeddings of a set of graphs based on rooted subgraphs extracted from them, actually expressing the potential connections between addresses in the transaction network. The major steps of Graph2Vec consist of the following two parts: 1) extract rooted subgraphs in order to represent all of the graphs in the form of a set of subgraphs. 2) learning embeddings of graphs by skip-gram model.

First, let H be the maximum height of rooted subgraphs, which means that for each node in the graph, Graph2Vec extract $(H+1)$ rooted subgraphs whose root is it. Then the t-th subgraph $(0 < t < H)$ of the certain node is considered as the surroundings around it within t hops. In a word, for a graph G with n nodes, Graph2Vec generates $n(H+1)$ subgraphs denoted by $c(G) = \{sg_1^{(0)}, sg_2^{(0)}, ..., sg_n^{(0)}, ..., sg_1^{(H)}, sg_2^{(H)}, ..., sg_n^{(H)}\}$. To extract these rooted subgraphs and quantize them into subgraph IDs, Graph2vec adopt the WL relabeling strategy that lays the basis for WL kernel [15].

After extracting rooted subgraphs from all the graphs, Graph2Vec learns the graph embeddings by skip-gram model. At the input layer of skip-gram model, the input graphs are encoded as one-hot vectors. And at the output layer, the predicted probability distribution is output over the substructure of inputted graphs. In the process of model training, the only hidden layer is able to gradually obtain the representation vector for the corresponding graph. Given a set of graphs $\{G_1, G_2, ..., G_n\}$ expected to be classified consist of subgraphs $c(G_1), c(G_2), ..., c(G_n)$, skip-gram model simultaneously learns embedding $f(G)$ for graph G and embedding for each rooted subgraph in $c(G)$. Then the model considers the probability that subgraph in $c(G)$ is contained in the graph G, maximizing the log-likelihood:

$$\sum_{j=1}^{n_i(H+1)} \log P_r(sg_j|G_i), \tag{1}$$

$$\frac{exp(f(G_i) \cdot f(sg_j))}{\sum_{sg \in V_{OC}} exp(f(G_i) \cdot f(sg))}, \tag{2}$$

n_i in Eq. 1 is the number of nodes in G. And V_{OC} in Eq. 2 means the vocabulary of subgraphs across all the graphs in the dataset. Given a pair of graphs G_1 and G_2, their final embeddings trained by the model get closer in the vector space if they are composed of many common rooted subgraphs, i.e graphs with analogous network structure will be embedded to similar representation vectors.

Then we put our attention to the analysis of the transaction network on Ethereum. As mentioned above, various of addresses exhibit different behavior patterns in the transaction networks formed with their neighbors. For example, financial applications, such as exchange markets, usually interact with a large amount of addresses frequently, for which the extracted network of this kind of address may contain many neighbor addresses with bilateral transactions. Intuitively, we could also summarize the typical behavior patterns of such abnormal addresses through the analysis of phishing life cycle. According to the previous work [10], The ultimate goal of scammers is to defraud Ether from other normal addresses, so their corresponding nodes will be connected to more in-degree neighbor nodes in the transaction network. Moreover, the phishing address usually completely cuts off the contact with the victim after it succeeds, for which there is often only exists a single transaction record between the phishing address and the victim address.

After visualization of the network as Fig. 2, we find that most of the transaction networks around the phishing addresses present like a converging star chart, i.e the Ether flow for most transactions is gathered from the outside to the central phishing address. We attach great importance to the value of this feature in our idea of detecting phishing addresses through the representation transaction subgraphs. The current graph classification algorithms, including Graph2Vec, which could only identify the label of address, are rarely able to handle the label of edge, so the improvement method we propose tries to integrate direction information of edge into Graph2Vec model. To achieve this goal, we try to use

Proposed improvment upon Graph2Vec algorithm: Edge direction

the edge label as the carrier of the transaction direction information, for which we label each edge of the network according to the Ether flow's direction of the corresponding transaction record.

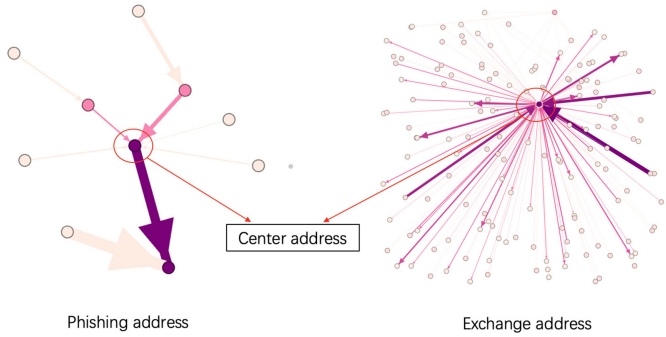

Fig. 2. Visualization of the transaction network of phishing address and exchange address.

In order to make full use of the feature information brought by the edge direction, we convert the original network based on address nodes and transaction edges into line graphs with the original transaction edge as the core analysis object of the model. In detail, the specific process is described as follows:

Direction stored in label of edge

- In the process of constructing the transaction graph of each target address through the collected information, the direction information of each transaction is recorded at the same time, and stored in the network through the label of the edge. There are two edge labels respectively correspond to deposit transactions and transfer transactions.
- In order to take into address both the structural information of the original graph and the importance of the edge information, we intend to transform the original graph to a new form named line graph. To build up the line graph, we covert the original edges to the nodes of the new graph, and connect the nodes that share the common endpoint node while they act as edges in the original graph. Then we keep the label of the original edge as the label of its corresponding node in the new graph, so we could make use of the information carried by the original edge in Graph2Vec model by the edge-to-node conversion.
- At last, we put the converted corresponding line graph into Graph2Vec model to acquire the representation vectors for all the graphs. It not only retains the original network structure information, but also incorporate edge direction information as one of the important elements into the process of model training. After acquiring the embeddings of graphs, we utilize general classifiers to perform downstream classification tasks (Fig. 3).

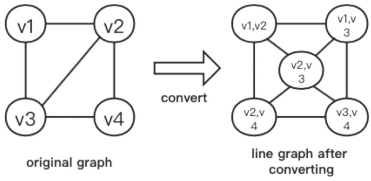

Fig. 3. Original graph and corresponding line graph after converting.

4 Experimental Evaluations

4.1 Dataset

We used the 801 phishing addresses after filtering and the same number of unlabeled addresses obtained above after filtering as the original data set of the experiment. In order to build a classification model and test its performance, we divided the original data set into two parts. The first part occupies 80% of the original dataset and is used as the training dataset of the classifier. The other part is the remaining 20% of the original data set, which will be used as the test dataset. After the training of the classifier is completed, the test dataset will be used as its input to obtain the experimental prediction results for subsequent comparative analysis.

50/50 split between phising and non-phising addresses

4.2 Metric

For the evaluation of the results of the binary classification model, we have adopted three evaluation metrics commonly used in machine learning here: Precision, Recall, and F1-score.

$$Precision = \frac{true\ positive}{true\ positive\ +\ false\ positive} \tag{3}$$

$$Recall = \frac{true\ positive}{true\ positive\ +\ false\ negative} \tag{4}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision\ +\ Recall} \tag{5}$$

Precision indicates how many predictions are accurate, from the perspective of prediction results. Recall indicates how many of the true positive classes were successfully recalled, i.e classified as predicted positive classes by model. F1-score is a comprehensive metric that combines the impact of precision and recall.

4.3 Baselines and Experimental Setup

Specifically, we intend to show the value and potentiality of our proposed framework from figuring out the following research questions: 1) Whether the proposed framework can extract latent feature information of the transaction network as addresses' features for subsequent phishing classification. 2) Whether

the proposed model reflects stronger adaptability for the phishing detection on Ethereum through more consideration on transaction records. So we compare our model with several baseline methods including Node2Vec, WL-kernel, and original Graph2Vec. Node2Vec is derived from the Word2Vec algorithm in the field of natural language processing. It samples through a specific random walk, generating a corresponding sequence of context for each node. Then treat these sequences as text and import them into the Skip-gram model in Word2Vec to get the embedding of central node. In the experiments, we set parameters of Node2Vec as recommended by the original paper [8] with walks per node $r = 10$, context size $k = 10$, walk length $l = 5$, while $p = 0.25$ and $q = 0.75$ for random walk.

WL-kernel is a method inspired by the Weisfeiler-Lehman test of graph isomorphism. Its key idea is iterate over each labeled node in graph and its neighbors to build a multiset label. In the end, we compare the similarity of graphs by counting the co-occurrences of labels in them after iteration. And the parameter settings is height parameter $h = 3$.

For all the methods, we all use a common embedding dimension of 256. And the hyperparameters of classifier are adjusted well based on 5-fold cross validation on the training set.

4.4 Results and Discussions

Table 1. Performance comparison of various methods.

Method	Metric		
	Precision	Recall	F1-measure
node2vec	0.57	0.58	0.58
WL-kernel	0.65	0.62	0.63
Graph2Vec	0.68	0.67	0.68
Line_Graph2Vec	**0.69**	**0.77**	**0.73**

Classification Performance. Figure 1 shows the performance comparison of various methods. We can find out some interesting phenomenon that meets our expectation. First, we pay attention to low-order substructure embedding technique, Node2vec. According to the result, using Node2vec to obtain graph embeddings performs the worst among all methods. We can find out the proposed approach outperforms it by more than 15%. Note that Node2vec is a representation learning method based on a lower-dimensional structure, i.e at the node level. It utilizes the similarity information of the neighbor node to train the representation vector, which means that the embeddings acquired by this kind

of method could only extract local information. Thus it is hard to satisfy our demand for mining deeper structure information in the transaction network.

Secondly, WL-kernel is a graph classification strategy based on graph kernel. As a graph classification algorithm, it also learns representation vectors based on graph substructure information. Compared with the Node2vec method, as a method considered at the network level, WL-kernel can further obtain relatively global structural information, so it could have better performance in the classification task. However, it is still not as good as the results of the Graph2Vec model, for the reason that Graph2Vec is more scalable and meticulous in the way of generating rooted subgraphs, extracting more precise features in the end. Moreover, it cannot deal with the label information of nodes or edges, so the behavioral characteristics that we have summarized according to the transaction patterns of phishing scammers could never play the role we expect.

Lastly, we compare the proposed framework incorporating the transaction direction information and the original Graph2Vec model, observing the impact that the transaction records may cause on the classification effect. As can be seen from the Table 1, our framework has better performance in all evaluation metrics, especially the recall has been greatly improved, which indicates that framework is provided with certain practical value. It shows the fact that the transaction pattern of phishing scammers we consider about is actually an important feature for identifying the addresses. So integrating the transaction direction information into the training of subgraphs representations could contribute to boosting the performance of framework. In addition, it shows that in the process of training, if more features of the transaction network can be considered, the training results of the model can give us more help in solving the anomaly detection problem. Moreover, it also reflects that the research perspective is still worth more exploring.

Starting from the motivation inspired by the strategy in the field of graph classification, we propose a framework to solve the problem of phishing detection via learning features from the representation transaction subgraphs. After designing based on the analysis of the transaction network and the patterns of phishing behavior, our proposed framework showed great competitiveness in the final experimental results. And the superiority and differences revealed by the prediction results presented in performance comparison also strongly illustrate that our idea of using transaction network to represent certain addresses can indeed uncover richer valuable information from the different dimensions of phishing detection problem.

Impact of Transaction Graph. In the first step of our proposed framework, according to the transaction records, we build a second-order transaction graph for each target address to represent it. To evaluate whether the second-order graph could certainly contribute to providing adequate information on the transaction network surrounding the target address, we set a comparison between it and the first-order transaction graph. Note that the first-order transaction graph only contains the first-order transaction information between the target address

and its first-order transaction neighbors. As shown in Table 2, the framework applying the second-order transaction graph has much better performance in all metrics. The result indicates that because of the limited topology of the first-order graph, our model based on Graph2Vec fails to extract enough features for the downstream classification task. Moreover, the second-order transaction graph remains more information of those transaction neighbors, which contributes to defining the transaction behavior pattern of the target address. Thus the second-order transaction graph is the best choice for our framework.

Impact of 2nd-order transaction subgraph

Table 2. Performance Comparison of different order of transaction graph.

Graph setting	Metric		
	Precision	Recall	F1-measure
First-order	0.57	0.52	0.54
Second-order	**0.69**	**0.77**	**0.73**

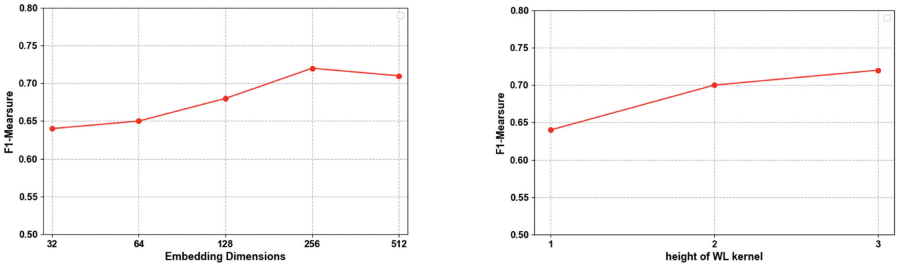

Fig. 4. Parameter sensitivity

Parameter Sensitivity. For the proposed framework, there are some parameters that influence the results. In Fig. 4, considering F1-Measure as the performance metric, we evaluate the effects of the chosen key parameters: embedding dimension and height of the WL kernel. When a parameter is adjusted for evaluating, other parameters are set as default. In order to evaluate how embedding dimensions could impact the detection performance, we gradually increase the dimensions from 32 to 512. From Fig. 4, we can observe that, as the increasing of embedding dimension, our framework has achieved better performance for classification. It indicates that larger dimensions could retain richer information that explored by the algorithm for the classification task. Besides for height of the WL kernel, we similarly adjust it from 1 to 3 and compare the detection results. Referring to Fig. 4, increasing the height of WL kernel can apparently

boost the final performance. Note that the height of WL kernel actually means the degree of rooted subgraphs that used to represent all the graphs. The results indicate that with a larger degree of rooted subgraphs, more features of network structure can be considered for representation learning.

Larger degree of rooted subgraphs produced better results

5 Conclusion

In this paper, we presented a new perspective to solve the problem of phishing detection on Ethereum. Firstly we utilized the surrounding transaction network of the addresses to characterize their positions and behaviors in the transaction network. Then through the analysis of the behavior pattern of the phishing scammers, we integrated the Ether flow feature information and build up an improved verification framework. In the final experiment, we found out from the comparison and analysis of the results that the network representation has the potential to extract more meta information from transaction network. Undoubtedly, the conclusion shows great potential value of the representation features extracted by graph embedding from the surrounding transaction network. In the follow-up work, we will make more use of transaction records collected, introducing attention mechanisms that could redistribute the weight of information in transaction network, to make the model better adapt to the Ethereum environment and phishing detection problem. At the same time, with the development of algorithms in the field of graph classification, there are more ideas and methods worthy of new attempts in this research direction.

Using line model

Acknowledgements. The work described in this paper was supported by the National Natural Science Foundation of China (61973325, 61503420) and the Fundamental Research Funds for the Central Universities under Grant No.17lgpy120.

References

1. Abdelhamid, N., Ayesh, A., Thabtah, F.: Phishing detection based associative classification data mining. Expert Syst. Appl. **41**, 5948–5959 (2014)
2. Adhikari, B., Zhang, Y., Ramakrishnan, N., Prakash, B.A.: Distributed representations of subgraphs. In: Proceedings of the 2017 IEEE International Conference on Data Mining Workshops, New Orleans, LA, USA, pp. 111–117. IEEE (2017)
3. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, New York, NY, USA, pp. 635–644. Association for Computing Machinery (2011)
4. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: Proceedings of the Association for the Advance of Artificial Intelligence, Phoenix, Arizona, USA, pp. 1145–1152. AAAI Press (2016)
5. Chang, T., Svetinovic, D.: Improving Bitcoin ownership identification using transaction patterns analysis. IEEE Trans. Syst. Man Cybern. Syst. **50**, 1–12 (2020)
6. Chau, D.H., Nachenberg, C., Wilhelm, J., Wright, A., Faloutsos, C.: Polonium: tera-scale graph mining and inference for malware detection. In: Proceedings of the 2011 SIAM International Conference on Data Mining. Mesa, Arizona, USA, pp. 131–142 (2011)

7. Duvenaud, D., et al.: Convolutional networks on graphs for learning molecular fingerprints. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 2224–2232. MIT Press, Cambridge (2015)
8. 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, New York, NY, USA, pp. 855–864. Association for Computing Machinery (2016)
9. Han, Q., Wu, J.W., Zheng, Z.Z.: Long-range dependence, multi-fractality and volume-return causality of Ether market. Chaos Interdisc. J. Nonlinear Sci. **30**, 011101 (2020)
10. Khonji, M., Iraqi, Y., Jones, A.: Phishing detection: a literature survey. IEEE Commun. Surv. Tutor. **15**, 2091–2121 (2013)
11. Lin, D., Wu, J., Yuan, Q., Zheng, Z.: Modeling and understanding Ethereum transaction records via a complex network approach. IEEE Trans. Circ. Syst. **67**, 1 (2020). II-Express Briefs
12. Moghimi, M., Varjani, A.Y.: New rule-based phishing detection method. Expert Syst. Appl. **53**, 231–242 (2016)
13. Monamo, P., Marivate, V., Twala, B.: Unsupervised learning for robust Bitcoin fraud detection. In: Proceedings of the Information Security for South Africa, Johannesburg, South Africa, pp. 129–134. IEEE (2016)
14. Sahingoz, O.K., Buber, E., Demir, O., Diri, B.: Machine learning based phishing detection from URLs. Expert Syst. Appl. **117**, 345–357 (2019)
15. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. J. Mach. Learn. Res. **12**, 2539–2561 (2011)
16. Wu, J., et al.: Who are the phishers? Phishing scam detection on Ethereum via network embedding. arXiv preprint arXiv:1911.09259 (2019)
17. Xie, S., Zheng, Z., Chen, W., Wu, J., Dai, H.N., Imran, M.: Blockchain for cloud exchange: a survey. Comput. Electric. Eng. **81**, 106526 (2019)
18. Zheng, P., Zheng, Z., Dai, H.: Xblock-ETH: extracting and exploring blockchain data from Ethereum. arXiv preprint arXiv:1911.00169 (2019)
19. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends. In: Proceedings of the 2017 IEEE International Congress on Big Data, Los Alamitos, CA, USA, pp. 557–564. IEEE (2017)
20. Zouina, M., Outtaj, B.: A novel lightweight URL phishing detection system using SVM and similarity index. Hum. Centric Comput. Inf. Sci. **7**, 17 (2017)