Temporal-Amount Snapshot MultiGraph for Ethereum Transaction Tracking

Yunyi Xie[1,2], Jie Jin[1,2], Jian Zhang[1,2], Shanqing Yu[1,2], and Qi Xuan[1,2,3(✉)]

[1] Institute of Cyberspace Security, Zhejiang University of Technology,
Hangzhou 310023, China
`xuanqi@zjut.edu.cn`
[2] College of Information Engineering, Zhejiang University of Technology,
Hangzhou 310023, China
[3] PCL Research Center of Networks and Communications, Peng Cheng Laboratory,
Shenzhen 518000, China

Abstract. With the wide application of blockchain in the financial field, the rise of various types of cybercrimes has brought great challenges to the security of blockchain. In order to better understand this emerging market and explore more efficient countermeasures for effective supervision, it is imperative to track transactions on blockchain-based systems. Due to the openness of Ethereum, we can easily access the publicly available transaction records, model them as a complex network, and further study the problem of transaction tracking via link prediction, which provides a deeper understanding of Ethereum transactions from a network perspective. Specifically, we introduce an embedding based link prediction framework that is composed of temporal-amount snapshot multigraph (TASMG) and present temporal-amount walk (TAW). By taking the realistic rules and features of transaction networks into consideration, we propose TASMG to model Ethereum transaction records as a temporal-amount network and then present TAW to effectively embed accounts via their transaction records, which integrates temporal and amount information of the proposed network. Experimental results demonstrate the superiority of the proposed framework in learning more informative representations and could be an effective method for transaction tracking.

Keywords: Ethereum · Random walk · Network embedding · Temporal network · Link prediction

1 Introduction

Blockchain is a distributed ledger technology, which can record transactions among peers [1]. It could be described as a trusted database that has the characteristics of decentralized, anti-counterfeiting, as well as user tamper-ability and anonymity. With the support of the underlying blockchain technology, blockchain

platforms such as Bitcoin and Ethereum also take this opportunity to flourish and become world-renowned new digital currency trading platforms. As the largest public blockchain-based platform enabling smart contracts, Ethereum [2] has become a widely used financial application platform and the corresponding cryptocurrency *Ether* is the second-largest cryptocurrency.

However, along with the rapid development of blockchain technology, various types of cybercrimes have arisen endlessly and thus Ethereum has become a hotbed of various cybercrimes [3–5]. Due to the anonymity of the blockchain, criminals attempt to evade supervision and engage in illegal activities by injecting funds into the blockchain system. It's reported that Ethereum has suffered from a variety of scams, such as hacks, phishing, and Ponzi schemes, showing that cybercrimes have become a critical issue in Ethereum. In order to create a favorable investment environment and preserve the sustainable development of blockchain-based systems, it's imperative to pay more attention to research in this field for formulating effective supervision.

This paper focuses on one of the solutions of Ethereum illegal activities, namely transaction tracking. The so-called transaction tracking is to maintain transaction security, identify fraud gangs, trace capital flows, retrieve stolen money, and improve the regulatory system. Furthermore, transaction tracking helps ordinary investors or cryptocurrency companies check whether certain funds or transactions are associated with illegal entities or contaminated by suspicious paths. In a summary, transaction tracking is an effective regulatory measure to prevent crime. The issues of transaction tracking have been widely discussed and many methods have been proposed. However, compared with traditional scenarios, illegal activities on Ethereum behave very differently. Traditional illegal activities generally rely on phishing emails and websites to obtain sensitive information from users. As a result, existing methods that aim to detect emails or websites that contain phishing and fraudulent information and thus cannot be directly applied to solve the transaction tracking problem on blockchain platforms like Ethereum.

Thanks to the openness of Ethereum, the available access transaction records that contain rich historical information can be used to study Ethereum matters. Here, we model Ethereum transaction records as a transaction network for further understanding and study transaction tracking on Ethereum from a network perspective, where a node represents an account and each edge represents a particular transaction (containing some unique information such as amount values and timestamps of the transactions). Network embedding [6] could be used to extract meaningful information as to representation features from transaction networks and thus can benefit lots of useful downstream tasks such as node classification [7], link prediction [8,9], graph classification [10,11], community detection [12], etc. The problem of tracking and predicting transactions on Ethereum transaction networks can be modeled as a link prediction task.

In this paper, we propose a transaction tracking framework on Ethereum from a network perspective. Firstly, we define a temporal-amount snapshot multigraph (TASMG) to model the Ethereum transaction records and make the

successive snapshots connect to reduce temporal loss. Furthermore, we introduce temporal-amount walk (TAW) to learn representations of accounts. For each account, the searching strategy depends on the amount transition probability and the temporal transition probability. Various experiments conducted on real-world Ethereum datasets demonstrate that the proposed framework can efficiently learn informative account representation, and solve the problem of transaction tracking.

The main contributions of this paper are summarized as follows.

- We study the matter of transaction tracking in Ethereum from a network perspective, which provides a deeper understanding of Ethereum transaction records and may contribute to the long-term development of the blockchain.
- We construct temporal-amount snapshot multigraph (TASMG) to retain Ethereum transactions' temporal and amount properties. Moreover, we propose a temporal-amount walk (TAW) to effectively embed accounts based on their transactions, which integrates temporal information and amount information of the Ethereum transaction networks.
- We carry on transaction tracking experiments on realistic Ethereum datasets, which demonstrate that our proposed framework significantly outperforms the baselines in transaction tracking. More precisely, our method combined with Ethereum datasets' properties can be superior to general methods.

The remainder of this work is organized as follows. In Sect. 2, we give a review of related work on blockchain transaction data analysis and summarize the related work on network embedding and link prediction. In Sect. 3, we demonstrate the framework for Ethereum transaction tracking. In Sect. 4, we conduct extensive experiments on real-world Ethereum datasets and compare our method with several network embedding techniques and similarity-based link prediction methods. Finally, we conclude our work in Sect. 5.

2   Related Work

2.1   Blockchain Transaction Analysis and Mining

In the traditional financial scenario, transaction records are sensitive information, which is usually private for security. Thanks to the openness of blockchain with publicly accessible transaction records, researchers can independently access Ethereum transaction records, which brings unprecedented opportunities for blockchain analysis. The study of blockchain transaction records has recently attracted considerable attention from different applications like graph analysis, price prediction, and anti market manipulation.

Wu et al. [13] proposed the concept of *Attributed Temporal Heterogeneous Motifs* and further addressed the issue of mixing detection using a detection model. Recently, they presented a network embedding model named *trans2vec* for transaction networks, which incorporates the transaction amount values and timestamps. It is worth noting that the model makes contributes to phishing

detection [4,5] and can be applied to other similar scenarios on transaction networks. Current researches on the blockchain mainly focus on cybercrimes, we argue that such preference may promote short-term security but may not conducive to the long-term development of the blockchain. In this paper, we consider the prediction of transactions and try to explore a deeper understanding of Ethereum transactions from a network perspective.

2.2   Network Embedding and Link Prediction

Network embedding has received extensive attention in the past decades [6]. Such embedding methods can transform each node into a low-dimensional vector, in which the structural information and topology properties of nodes are preserved as much as possible. Link prediction [14] aims at estimating the likelihood of the existence of links between nodes based on the current network information. Existing link prediction methods can be classified into several categories, e.g., similarity-based algorithms, maximum likelihood algorithms, and probabilistic models and most of the existing network embedding techniques can be applied to link prediction.

The earliest researches of network embedding are mainly based on random walk with Skip-Gram [15] model to learn node representation, where Deep-Walk [16] and Node2vec [17] are two classic examples using random walks on networks to obtain node representation. LINE [18] can optimize node's first-order and second-order proximity by designing specific objective function. Factorization based network embedding represents the connections between nodes in the form of a matrix and obtains the node embedding by factorizing this matrix. Graph Factorization (GF) [19] and HOPE [20] are two of the most notable factorization based methods. GCN based autoencoder [21], e.g., GAE and VGAE, has been widely used, which aims to minimize the reconstruction error of the output and input by its encoder and decoder.

Due to random walk based methods' high expressiveness and learning ability, and the diversity of Ethereum transactions, we model the general transaction network as a temporal-amount snapshot multigraph (TASMG), which combines transaction temporal and amount information. Furthermore, we propose several specific search strategies to perform random walk, namely temporal-amount walk (TAW), on the proposed TASMG. The random walk strategies reflect transaction amount and temporal properties simultaneously.

3   Methodology

3.1   Basic Definition

Network $G = (V, E)$ can be established by Ethereum transaction records, where V denotes the set of accounts (nodes) and E represents the transaction records (edges) with transaction temporal and amount information. According

to the given time interval ϵ, the transaction network G can be divided into snapshots $\{G_1, G_2, G_3, \cdots\}$. In this paper, we define TASMG to both retain the current time and nearby time snapshot, which can capture the changing tendency of accounts' transaction behavior. Further, we propose TAW to capture the temporal and amount properties of each account in TASMG. Figure 1 demonstrates the main steps of the proposed framework, including network construction, network embedding, and transaction tracking.
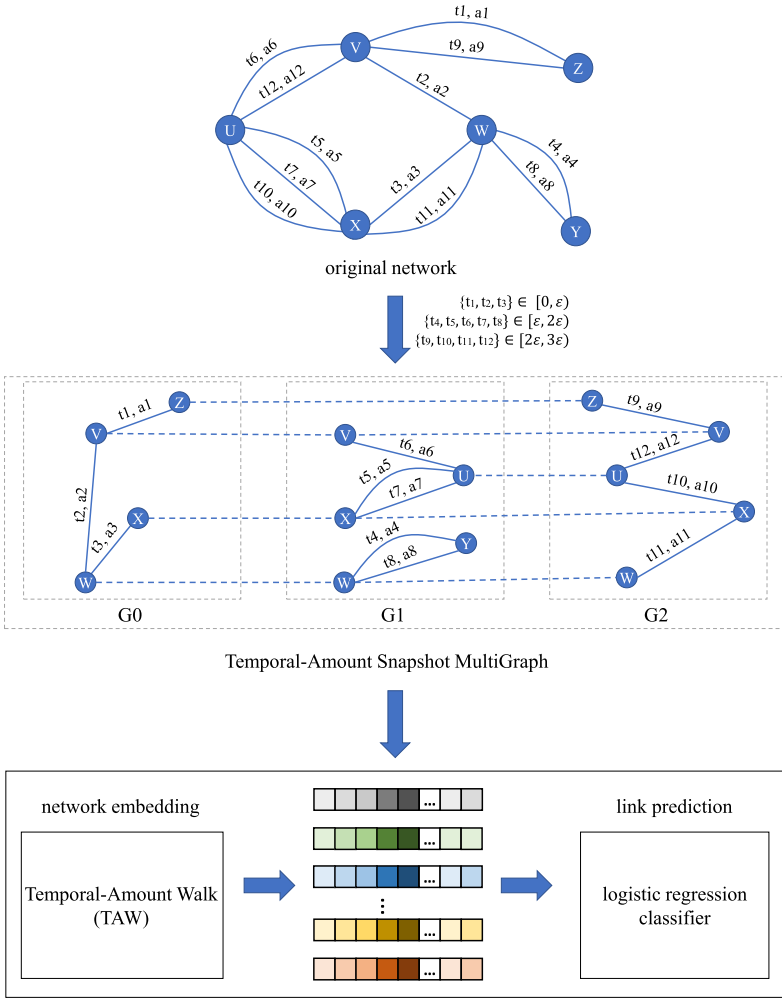

Fig. 1. The detailed framework for Ethereum transaction tracking.

Definition 1 (temporal-amount snapshot multigraph (TASMG)). *Divide $G = (V, E)$ into several snapshots according to time span ϵ, $G_t = (V_t, E_t)$*

be one of the snapshots. Nodes and edges in snapshot G_t are active between the timespan $[t\epsilon, (t+1)\epsilon)$, where time order $t \in \{0, 1, 2, \cdots\}$. Edge is represented as $e = (u, v, w, t)$, for $\forall e \in E$, $Src(e) = u$, $Dst(e) = v$, $W(e) = w$, $T(e) = t$, where u is the source node, v is the target node, w is the weight (transaction amount value) and t is the time accessibility. Self-connections are established when nodes exist in successive snapshots and on the premise that each snapshot is sorted in ascending order by time order t.

Self-connections in TASMG can make the random walk traverse across successive snapshots in non-decreasing order of edge's temporal information, which can capture the correlation between different snapshots and thus may result in more informative embeddings. For simplicity, we provide time accessibility for the edge between each pair-wise nodes and every two time slices linked nodes. Let $\eta_+ : \mathbb{R} \to \mathbb{Z}^+$ be a function that maps each node to an index based on the time order t, i.e., for a given node u in snapshot G_i, we have $\eta_+(u) = i$. In particular, we design a mapping function for each edge e in TASMG: $T(e) = \eta_+(v) - \eta_+(u) \in \{-1, 0, 1\}$, where v is the target node and u is the source node, to define the time accessibility of v from u, i.e., v is time accessible from u if and only if the corresponding $T(e) \geq 0$. Then, we define *successive edges* as shown below.

Definition 2 (Accessible Edge). *Given a TASMG $G = (V, E)$, the node v's accessible edges are defined as follows:*

$$L_t(v) = \{e \mid Src(e) = v, T(e) \geq 0\}$$

In the proposed network TASMG, a valid temporal walk is composed of accessible edges, i.e., a sequence of nodes connected by edges with non-decreasing $T(e)$. Next, we will present several sampling strategies by determining the selection probability for $e \in L_t(v)$.

3.2   Temporal Biased Walk

Based on the definitions in Sect. 3.1, we design a sampling strategy to choose accessible edges both in temporal and amount domains.

Temporal Transition Probability: The whole transaction network can be divided into different snapshots according to the time span, each snapshot represents a part of the network structure, and the gradual change of snapshot reflects the evolution process of the network. Ignoring the relevant information existing between two snapshots in successive time steps may result in the loss of temporal information. Therefore, we propose temporal transition probability to capture the behavior changes of nodes in different snapshots. In this case, the probability of selecting each edge $e \in L_t(c)$ can be given as follows:

$$P_T(e) = \frac{\psi_T(e)}{\sum_{e' \in L_t(c)} \psi_T(e')} \tag{1}$$

where $\psi_T(e)$ is expressed as

$$\psi_T(e) = \begin{cases} \alpha, & T(e) > 0 \\ 1 - \alpha, & T(e) = 0 \end{cases} \tag{2}$$

Here, the temporal bias α ($0.1 \leq \alpha \leq 0.9$) determines that the temporal walk will stay on the current snapshot or transfer to the next.

Amount Transition Probability: Apart from transaction time, the transaction amount value of edges also plays an essential role in transaction networks. In the following, we present unbiased, biased, and linear strategies from the amount domain, respectively.

- **Amount Unbiased Sampling (AUS).** This is the default setting in the amount domain, which assumes that each successive edge $e \in L_t(c)$ of node c has the same probability to be selected:

$$P_A(e) = \frac{1}{L_t(c)} \tag{3}$$

- **Amount Biased Sampling (ABS).** The amount value of each transaction indicates the importance of the interaction between the two accounts involved. In most cases, a higher value of transaction amount implies that there is a greater connection between the two accounts. Thus each edge $e \in L_t(c)$ can be assigned the selection probability:

$$P_A(e) = \frac{W(e)}{\sum_{e' \in L_t(c)} W(e')} \tag{4}$$

  where $W(e)$ is the transaction amount value between node c and its temporal neighbor x.

- **Amount Linear Sampling (ALS).** In order to avoid the extreme situation that the edge with a small amount value will never be sampled or the edge with a large amount value will always be sampled, we consider a linear mapping function to weaken the effects of the edge's transaction amount value. Thus we have:

$$P_A(e) = \frac{\omega_+(W(e))}{\sum_{e' \in L_t(c)} \omega_+(W(e'))} \tag{5}$$

  with $\omega_+(v)$ mapping the amount value of edge to an ascending ranking. In other words, $\omega_+(v)$ maps each edge's transaction amount value to an index with $\omega_+(e) = 1$ for the smallest transaction amount value.

Joint Transition Probability: Furthermore, we normalize the aforementioned temporal transition probability and amount transition probability, and then combine them as one. We set the unnormalized transition probability to $P(e)$ and then normalize it to the final transition probability for each edge $e \in L_t(c)$, where

$$P(e) = P_T(e)\, P_A(e) \tag{6}$$

3.3   Learning Network Embeddings

This paper determines to obtain the d-dimensional representation of each node from a mapping function $f : V \rightarrow \mathbb{R}^d$. Let $N(v)$ and $f_t(v)$ denote the temporal neighbors generated according to the given sampling strategy and the representation of node v in the snapshot G_t, respectively. The following objective function aims to maximize the log-probability of observing $N(v)$ and historical embedding $f_t(v)$ conditioned on the node v's representation:

$$\max_f \sum_{v \in V} \log(Pr(N(v), f_t(v) \,|\, f(v))) \tag{7}$$

According to the conditional independence assumption in the Skip-Gram model, we factorize the formula:

$$
\begin{aligned}
&\log(Pr(N(v), f_t(v) \,|\, f(v))) \\
&= \log(\prod_{u_i \in N(v)} Pr(u_i \,|\, f(v))) + \log(Pr(f_t(v) \,|\, f(v)))
\end{aligned} \tag{8}
$$

The definition of neighboring nodes is generally symmetric, i.e., the likelihood of a target node is independent of any other nodes. Therefore, we decompose the probability of the observation temporal neighborhood, and model the probability of each target neighborhood node pair as a softmax unit. Given the learning representation $f(v)$, the conditional probability of the observation node ui can be transformed as follows:

$$Pr(u_i \,|\, f(v)) = \frac{\exp(f(u_i)\, f(v))}{\sum_{n \in V} \exp(f(n)\, f(v))} \tag{9}$$

where $u_i \in N(v)$ is the ith near-neighbor of node v. With the above hypothesis, the objective function in Eq. (7) simplifies to:

$$
\begin{aligned}
\max_f \sum_{v \in V} \log(\prod_{u_i \in N(v)} &\frac{\exp(f(u_i)\, f(v))}{\sum_{n \in V} \exp(f(n)\, f(v))}) \\
&+ \log(Pr(f_t(v) \,|\, f(v)))
\end{aligned} \tag{10}
$$

4   Experimental Results

4.1   Dataset

As the largest public blockchain-based platform, Ethereum's [22] transaction records are completely public and researchers can easily obtain the objective account's historical transaction data through the API of Etherscan (`etherscan.io`). Due to the size of the total transaction records is extremely large [23], we ascertain a number of target accounts and then obtain their transactions from Ethereum transaction records to make subgraphs for subsequent experiments. As shown in Fig. 2, we randomly sample a centered
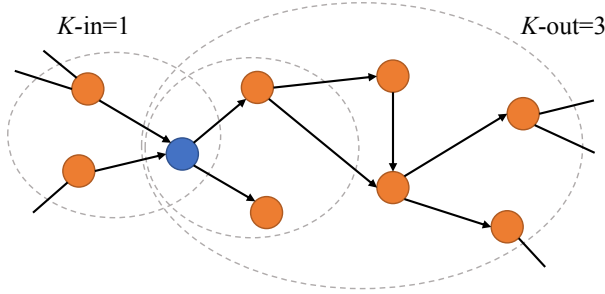
Fig. 2. Schematic illustration of K-order subgraph.

Table 1. Basic topological features of the three subgraphs. $|V|$ and $|E|$ are the numbers of nodes and edges, respectively, $\langle K \rangle$ is the average degree, and $\langle C \rangle$ is the average clustering coefficient.

	$|V|$	$|E|$	$\langle K \rangle$	$\langle C \rangle$
EthereumG1	2100	6995	6.662	0.211
EthereumG2	5762	9098	3.158	0.112
EthereumG3	10269	28431	5.537	0.147

account to obtain its local structure information and then extract K-order subgraph [24]. K-in and K-out are two parameters to control the depth of sampling inward and outward from the center, respectively.

In the following experiments, we collect three subgraphs for transaction tracking with different sizes from the whole Ethereum transaction records, i.e., randomly select different center accounts and collect three subgraphs with K-in = 1, K-out = 3. A summary of these networks is listed in Table 1.

4.2 Baselines and Experimental Setup

It is assumed that node pairs with higher similarity values are more likely to have interactions, and thus we calculate the similarity value for each node pair, e.g., Common Neighbors (CN), Adamic–Adar Index (AA), Resource Allocation Index (RA). Furthermore, we compare the performance of TAW with several baseline methods including GF, HOPE, GAE, VGAE, LINE, DeepWalk, and Node2vec. GF and HOPE are two of the most notable factorization based embedding methods, while GAE and VGAE are the GCN based autoencoder models. LINE preserves the first-order and second-order proximity between nodes and in this work, the final representation for each node is created by second-order representations. DeepWalk and Node2vec belong to the same category of Skip-Gram based methods. The parameters for both the node sampling and optimization steps of the two random walk based embedding methods, including ours, are set exactly the same: number of walks per node $w = 10$, the length of walk $l = 80$,

and the size of context window $k = 5$. Since Node2vec requires the in-out and return hyperparameters, the ranges of its hyper parameters in this paper are set to $p, q \in \{0.5, 1, 2\}$. For our proposed method TAW, we take one month as the time span to construct TASMG for each dataset and temporal bias α varies in $[0.1, 0.9]$.

For all methods, the number of dimensions of output vector representations d is set to 128. After learning embedding for each node, we use hadamard[1] operation to calculate the vectors of the corresponding edges on the embedding vectors of the learned pairs of nodes. And we train a one-vs-rest logistic regression classifier to classify the links in the test set. Experiments are repeated for 5 random seed initializations and we report the average performance.

4.3   Metrics

- **AUC:** It can be interpreted as the probability that a randomly chosen missing link is given a higher score than a randomly chosen nonexistent link. If among n independent comparisons, there are n' times that the missing link gets a higher score and n'' times they get the same score, the AUC value is

$$\text{AUC} = \frac{n' + 0.5n''}{n}$$

  If all the scores are generated from an independent and identical distribution, the AUC value should be about 0.5. The degree to which the value exceeds 0.5 indicates how much better the algorithm performs than pure chance.
- **Precision:** Given the ranking of the non-observed links, the precision is defined as the ratio of relevant items selected to the number of items selected, which indicates how many predictions are accurate, from the perspective of prediction results. That is to say, if we take the top-L links as the predicted ones, among which L_r links are right, then the Precision equals L_r/L. Clearly, higher precision means higher prediction accuracy.

4.4   Results and Discussions

Transaction tracking can be modeled as a link prediction task in Ethereum, which goal is to predict the occurrence of links in a given network based on observed information. Before the experiments, we hide some account connections in the trading network. We aim to track these missing connections by using all the remaining information to predict the transaction records of the account. We first randomly hide 20% of links in the original network as positive samples of the training set and use the remaining to train all methods. Then we stochastically sample an equal number of node pairs with no link as negative samples of the training set. The test set consists of two parts, one of which contains all the hidden edges as the positive samples, and the other contains the unconnected pairwise nodes sampled randomly as the negative samples.

---

[1] $[f(u) \cdot f(v)]_i = f_i(u) * f_i(v)$.

Table 2. Performances of different methods for transaction tracking.

| Metrics	EthereumG1		EthereumG2		EthereumG3	
	AP	AUC	AP	AUC	AP	AUC
CN	0.6907	0.6848	0.5134	0.4921	0.6826	0.6881
AA	0.7367	0.7002	0.5909	0.5099	0.6979	0.6912
RA	0.7378	0.7007	0.5909	0.5099	0.6986	0.6914
Jaccard	0.5088	0.6097	0.4523	0.4588	0.6234	0.6800
GF	0.7827	0.6821	0.7377	0.7050	0.7946	0.6662
HOPE	0.7698	0.6578	0.8300	0.7580	0.8714	0.8089
LINE	0.7761	0.7521	0.8627	0.8237	0.6371	0.6370
DeepWalk	0.6159	0.6637	0.6138	0.6307	0.7755	0.8024
Node2vec	0.6877	0.7149	0.6939	0.6990	0.8239	0.8501
GAE	0.7911	0.6752	0.5828	0.3729	0.8703	0.7885
VGAE	0.8179	0.7184	0.6683	0.4719	0.8934	0.8278
TAW	**0.8774**	**0.8819**	**0.9180**	**0.9134**	**0.9115**	**0.9108**

Transaction Tracking Performance. The experimental results of transaction tracking are given in Table 2. We find that similarity indices, i.e., CN, Jaccard, RA and AA, perform poorly among all methods. These similarity indices could only extract incomplete information, which is hard for mining deep structure information of transaction networks. We further observe that the sparser of network, the harder it for GCN-based methods to predict the appearance of links. GF, HOPE, and LINE achieve poor performance on most networks, indicating that preserving higher-order proximity is not conducive to predicting unobserved transactions. Here, the random walk based methods outperform other methods, indicating that random walks are especially useful when approximate node centrality and similarity in Ethereum transaction networks. However, it is still not as good as the results of TAW, for the reason that our proposed transaction tracking framework is more meticulous in the way of generating TASMG and extracting more precise features. Overall, the evaluation indicates that TAW achieves clear performance gains over the baselines for transaction tracking problem, which is reasonable since our method with flexible walking strategies is able to learn the similarity between nodes more effectively. Moreover, the outstanding performance of our proposed transaction tracking framework also suggests that TASMG, which better combines temporal and amount properties, can indeed uncover richer valuable information from the different dimensions of Ethereum transaction records.

Parameter Sensitivity. Table 3 shows the difference between different amount sampling strategies. Interestingly, using a biased strategy seems to improve slightly on the tested datasets, which implies that the probability of transaction occurrence is positively correlated with the amount value of the transaction

between the two accounts. We thus can infer that the larger transaction amount value of the two accounts, the closer relationship between the two. In addition, since α is newly introduced in our method, we focus on the influence of temporal bias α on the proposed framework. In order to evaluate how temporal bias α could impact the transaction tracking performance, we gradually vary the temporal bias α in $[0.1, 0.9]$. The results are shown in Fig. 3, where we can observe that, overall, as the increasing of temporal bias α, our framework has achieved better performance for transaction tracking.

Table 3. Performances of different amount sampling strategies.

Metrics	EthereumG1		EthereumG2		EthereumG3	
	AP	AUC	AP	AUC	AP	AUC
AUS	0.8774	0.8819	0.9180	0.9134	0.9115	0.9108
ABS	**0.8852**	**0.8836**	**0.9287**	0.9184	**0.9189**	**0.9158**
ALS	0.8798	0.8811	0.9257	**0.9199**	0.9150	0.9137
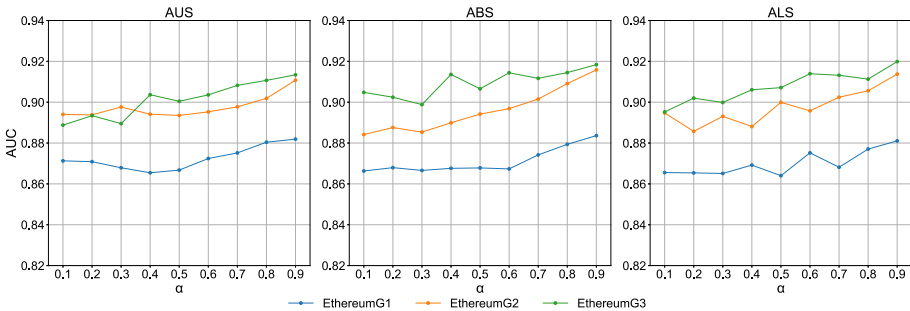
Fig. 3. Performance in terms of AUC under temporal bias α which change is in $[0.1, 0.9]$.

5   Conclusion

In this paper, we model a general transaction network as a temporal-amount snapshot multigraph (TASMG), which provides a novel network perspective for a deeper understanding of Ethereum transactions. Furthermore, we implement a random walk using specific search strategies to characterize accounts' behaviors on the proposed TASMG, namely temporal-amount walk (TAW). Experimental results demonstrate the effectiveness of our proposed transaction tracking framework, and indicate that TASMG has the potential to extract more information from Ethereum transaction records. Though Ethereum transaction records are publicly available, it's still relatively unexplored till now. For future work, we

plan to apply deep learning methods to expand our methods and further extend the current framework to analyze more illegal activities on Ethereum and create a safe trading environment for Ethereum.

References

1. Swan, M.: Blockchain: Blueprint for a New Economy. O'Reilly Media, Inc. Sebastopol (2015)
2. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., Wang, F.-Y.: Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Trans. Syst. Man Cybern. Syst. **49**(11), 2266–2277 (2019)
3. Yuan, Z., Yuan, Q., Wu, J.: Phishing detection on Ethereum via learning representation of transaction subgraphs. In: Zheng, Z., Dai, H.-N., Fu, X., Chen, B. (eds.) BlockSys 2020. CCIS, vol. 1267, pp. 178–191. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-9213-3_14
4. Wu, J., Yuan, Q., Lin, D., You, W., Chen, W., Chen, C., Zheng, Z.: Who are the phishers? Phishing scam detection on Ethereum via network embedding. In: IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020)
5. Yuan, Q., Huang, B., Zhang, J., Wu, J., Zhang, H., Zhang, X.: Detecting phishing scams on Ethereum based on transaction records. In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2020)
6. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. IEEE Trans. Knowl. Data Eng. **31**(5), 833–852 (2018)
7. Rossi, R., Neville, J.: Time-evolving relational classification and ensemble methods. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012. LNCS (LNAI), vol. 7301, pp. 1–13. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30217-6_1
8. Fu, C., et al.: Link weight prediction using supervised learning methods and its application to yelp layered network. IEEE Trans. Knowl. Data Eng. **30**(8), 1507–1518 (2018)
9. Zhang, J., Zheng, J., Chen, J., Xuan, Q.: Hyper-substructure enhanced link predictor. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2305–2308 (2020)
10. Xuan, Q., et al.: Subgraph networks with application to structural feature space expansion. IEEE Trans. Knowl. Data Eng. **33**(6), 2776–2789 (2021). https://doi.org/10.1109/TKDE.2019.2957755
11. Zhou, J., Shen, J., Yu, S., Chen, G., Xuan, Q.: M-Evolve: structural-mapping-based data augmentation for graph classification. IEEE Trans. Network Sci. Eng. **8**(1), 190–200 (2020)

12. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
13. Wu, J., et al.: Detecting mixing services via mining bitcoin transaction network with hybrid motifs. IEEE Trans. Syst. Man Cybern. Syst. (2021)
14. Lü, L., Zhou, T.: Link prediction in complex networks: a survey. Physica A: Stat. Mech. Appl. **390**(6), 1150–1170 (2011)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
16. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
17. 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, pp. 855–864 (2016)
18. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077 (2015)
19. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J.: Distributed large-scale natural graph factorization. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 37–48 (2013)
20. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1105–1114 (2016)
21. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
22. Wood, G., et al.: Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper **151**(2014), 1–32 (2014)
23. Chen, T., et al.: Understanding Ethereum via graph analysis. ACM Trans. Internet Technol. (TOIT) **20**(2), 1–32 (2020)
24. Lin, D., Wu, J., Yuan, Q., Zheng, Z.: Modeling and understanding Ethereum transaction records via a complex network approach. IEEE Tran. Circ. Syst. II: Express Briefs **67**(11), 2737–2741 (2020)