

Modeling and Understanding Ethereum Transaction Records via a Complex Network Approach

Dan Lin, Jiajing Wu¹, Senior Member, IEEE, Qi Yuan, and Zibin Zheng², Senior Member, IEEE

Abstract—As the largest public blockchain-based platform supporting smart contracts, Ethereum has accumulated a large number of user transaction records since its debut in 2014. Analysis of Ethereum transaction records, however, is still relatively unexplored till now. Modeling the transaction records as a static simple graph, existing methods are unable to accurately characterize the temporal and multiplex features of the edges. In this brief, we first model the Ethereum transaction records as a complex network by incorporating time and amount features of the transactions, and then design several flexible temporal walk strategies for random-walk based graph representation of this large-scale network. Experiments of temporal link prediction on real Ethereum data demonstrate that temporal information and multiplicity characteristic of edges are indispensable for accurate modeling and understanding of Ethereum transaction networks.

Index Terms—Ethereum, blockchain, complex networks, graph representation, cryptocurrency, transaction network.

I. INTRODUCTION

BEING the largest public blockchain-based platform supporting smart contracts, Ethereum [1] has attracted wide attention recently. To facilitate the implementation of smart contracts, Ethereum introduces the concept of *account*, which is formally an address. The corresponding cryptocurrency on Ethereum, known as *Ether*, can be transferred between accounts and used to compensate participant mining nodes. Current research about Ethereum mainly focus on security and performance issues of the blockchain technology [2], [3], [4], however, the interactive relationship between users and smart contracts is relatively unexplored till now.

Complex network is a universal tool to analyze real work systems from various fields [5], [6], [7], and it has been employed to model and analyze the huge transaction networks of blockchain-based systems. In 2018, Chen *et al.* [8] conducted the first systematic study to characterize Ethereum and obtain new observations with various metrics. Alqassem *et al.* [9] found the anti-social properties of Bitcoin system via transaction network data analysis. In

existing studies, transaction records are modeled as a simple graph, where multiple transactions between a pair of addresses are merged as a one-time transaction in the graph construction procedure.

Different from other large-scale complex networks, each edge in the Ethereum transaction network represents a particular Ether transaction, and thus contains some unique information such as the direction, amount value and timestamp of a particular transaction. It is essential to incorporate the aforementioned information for accurate modeling, characterization, and understanding of transaction network data. In addition, multiple transactions between two users are expected and it is more comprehensive to model a transaction network as a *multidigraph* rather than a simple graph. In graph theory, a multidigraph (in contrast to a simple graph) is a graph which is permitted to have self-loops and multiple edges (also called parallel edges). A multidigraph is a directed multigraph. At present, there is little research on data mining of multidigraphs, and existing methods modeling the transaction records as simple graph are inadequate to characterize the temporal and multiple features of transaction networks.

Random walk mechanism has been widely demonstrated to be an effective technique to measure local similarity of networks for a variety of domains and tasks [10], [11], [12]. Besides, random walk has also become a popular sampling method for the problem of graph representation, which aims to represent node features of large networks in a low dimensional space for graph analysis and mining [13]. Most existing methods of graph representation are designed for mining social networks and few studies have considered the scenarios of transaction data mining as most financial transaction data is not public. However, traditional methods for graph representation in social networks cannot be directly applied to analyze financial transaction networks, as the existing methods [13] ignore the multiplicity and temporal attributes of edges in transaction networks.

In this brief, we conduct the *first* work to understand Ethereum transaction records via random-walk based graph representation learning. In particular, we first model the Ethereum transaction records as a *Temporal Weighted Multidigraph*, and propose novel graph sampling and representation methods considering the temporal and weighted information of the graph. After that, we conduct link prediction experiments on realistic Ethereum data to evaluate the effectiveness of the proposed graph modeling and mining methods.

II. FRAMEWORK

In order to analyze the Ethereum transaction records comprehensively, we propose a general framework which includes

Manuscript received December 26, 2019; accepted January 16, 2020. Date of publication January 21, 2020; date of current version November 4, 2020. This work was supported in part by the National Key Research and Development Program under Grant 2016YFB1000101, in part by the National Natural Science Foundation of China under Grant 61973325 and Grant 61503420, and in part by the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2016). This brief was recommended by Associate Editor F. C. M. Lau. (Corresponding author: Jiajing Wu.)

The authors are with the School of Data and Computer Science and the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510275, China (e-mail: wujiajing@mail.sysu.edu.cn).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSII.2020.2968376

1549-7747 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

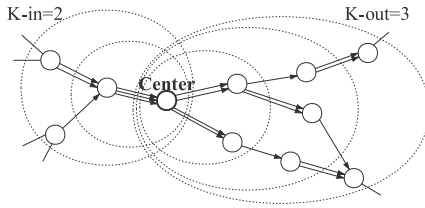


Fig. 1. A schematic illustration of a directed K -order subgraph.

four main procedures: (a) Data collection. We fetch transaction records of objective accounts from Ethereum through API of <https://etherscan.io/> (A block explorer and analytics platform for Ethereum); (b) Network construction. Based on collected transaction data, we construct a *Temporal Weighted Multidigraph* to represent the relationship of Ether transfer between accounts; (c) Random-walk based graph representation. We employ the graph representation algorithm based on designed random walk strategies to learn richer node representations; (d) Link prediction for evaluation. We conduct experiments on temporal link prediction problem to evaluate the effectiveness of our framework on Ethereum transaction networks.

III. DATA COLLECTION

Accounts on Ethereum can be divided into two categories, i.e., external owned accounts (EOA) which are similar to general bank accounts [14], and smart contract accounts which are source code files. In this brief, we focus on the transactions among EOAs for the reason that the Ether transfer records between them are publicly available on the blockchain. Besides, we only include the successful transactions with non-zero amount value into our dataset.

Since it is extremely time-consuming to process the whole Ethereum transaction network with more than two million EOAs [8], here we select a number of objective accounts and then obtain their transaction data through APIs of Etherscan. As shown in Fig. 1, centered by a particular objective account, we obtain a directed K -order subgraph. K -in and K -out are two parameters to control the depth of sampling inward and outward from the center, respectively.

On Ethereum, various related information of Ether transactions is stored as data packages. In details, the *TxHash* field is a unique identification of a transaction, the *Value* field in a transaction refers to the amount of money transferred, and the *Timestamp* field indicates when the transaction happens. Besides, the *From* and *To* fields denote the sender and recipient of the transaction. With the collected four-tuples (*From*, *To*, *Value*, *Timestamp*), we can construct a temporal weighted multidigraph.

In this brief, we conduct the first work to understand Ethereum transaction records from a complex network perspective, and we will open our collected dataset from Ethereum for future studies.

IV. NETWORK CONSTRUCTION

Before network construction, we first discuss some realistic rules and features of transaction networks like the Ethereum: (1) Transaction networks evolve continuously over time with

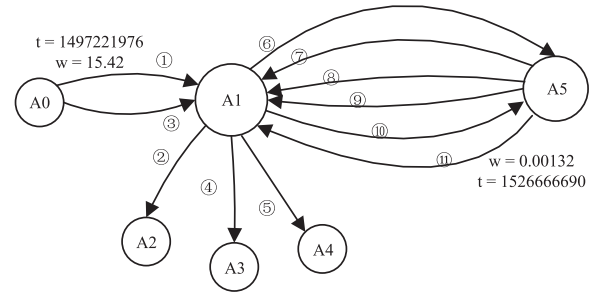


Fig. 2. An illustration for the Ethereum transaction network. Nodes are labeled by account addresses. Edges are attached by timestamp t and amount value w , and indexed in the increasing order of t .

additions of links, which is overlooked in most of the existing random-walk based graph algorithms; (2) The practical meaning of connections between accounts is not a one-off established relationship but a time-dependent event. Hence multiple edges need to be considered in transaction network modeling; (3) Unlike social networks, random walks on Ethereum transaction network are concrete, which represent realistic money transfer flows; (4) The amount value of a transaction reflects the similarity between the two involved accounts to some extent. In most cases, the larger amount of transaction, the closer relationship between two accounts. Fig. 2 demonstrates a microcosm of transaction activities on Ethereum.

Ether transfer is one of the major activities happening on Ethereum. Here we abstract an Ether transfer transaction as a four-tuple (src, dst, w, t), which means the sender src transfers w Ether to the recipient dst at time t . To investigate the Ether transfer on Ethereum, we abstract the Ethereum transaction network as a *Temporal Weighted Multidigraph*:

Definition 1 [Temporal Weighted Multidigraph (TWMDG)]: Given a graph $G = (V, E)$, let V be the set of nodes and E be the set of edges. Each edge is unique and is represented as $e = (u, v, w, t)$, where u is the source node, v is the target node, w is the weight value and t is the timestamp. For the sake of simplicity, we define mapping functions $Src(e) = u$, $Dst(e) = v$, $W(e) = w$, $T(e) = t$ for $\forall e \in E$.

Based on collected four-tuples from Ethereum transaction records, we can build a TWMDG where each node represents a unique account and each edge represents a unique Ether transfer transaction.

V. RANDOM-WALK BASED GRAPH REPRESENTATION

Random-walk based graph representation methods have been proved to be scalable and effective for large graphs. As realistic transaction networks usually have a huge network size, we consider the random walk based method for graph analysis in this area. Before utilizing graph representation algorithm, we investigate the degree distribution of Ethereum transaction network, and find that the distribution of nodes in both the entire graph and the subgraph of Ethereum follow power-law (Fig. 3 plots the distribution of a subgraph). As discussed in a pioneering study [15], random-walk based sampling methods can well preserve structural properties of a graph with a power-law degree distribution. Inspired by this

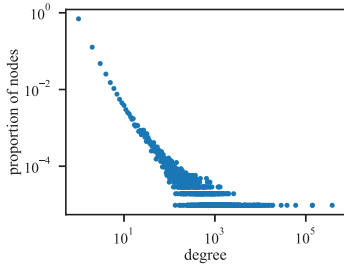


Fig. 3. The distribution of nodes appearing in Ethereum transaction subgraph follows a power-law.

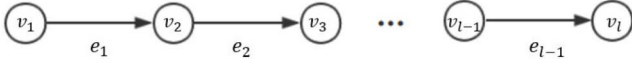


Fig. 4. Illustration of an l -length temporal walk.

brief, we employ graph representation method based on random walk to extract information from the Ethereum transaction network.

However, for the transaction networks discussed here, existing methods that ignore temporal information may sample a large number of invalid transaction sequences to derive node representations. For example, in Fig. 2, $\{A5, A1, A2\}$ is a possible random walk sequence in traditional methods DeepWalk [15] and node2vec [16], but it is not practical in a temporal graph as the transaction from A1 to A2 happens earlier. While in a recent work reported in [17], although temporal information is considered, the existence of multiple edges between points is neglected. For instance, the temporal walk from A0 to A1 is represented as a sequence of nodes $\{A0, A1\}$. However, whether A2 is possible for the next walk depends on whether the transaction path ① or ③ is sampled by the previous walk from A0 to A1.

In this brief, we represent an l -length temporal walk as a sequence of l nodes together with a sequence of $(l - 1)$ edges traversed in non-decreasing timestamps. This kind of temporal walk represents an actually feasible path for money flow in the transaction network. Therefore, the proposed method is expected to learn more meaningful and accurate time-dependent node representations that can capture more comprehensive properties from dynamic transaction networks. For a temporal weighted multidigraph discussed here, we define the concept of a *Temporal Walk* as follows.

Definition 2 (Temporal Walk): In TWMDG, a temporal walk from node v_1 to v_l is an l -length path traversed in non-decreasing timestamps. Such a temporal walk is represented as a sequence of l nodes $walk_n = \{v_1, v_2, \dots, v_l\}$ together with a sequence of $(l - 1)$ edges $walk_e = \{e_1, e_2, \dots, e_{l-1}\}$, where $Src(e_i) = v_i$, $Dst(e_i) = v_{i+1}$ ($1 \leq i \leq (l - 1)$), and $T(e_i) \leq T(e_{i+1})$ ($1 \leq i \leq (l - 2)$). We define that nodes u and v are temporally connected if there exists a temporal path from u to v .

In order to sample valid random walks obeying the temporal constraint, we introduce a new concept called *Temporal Successive Edges* in TWMDG.

Definition 3 (Temporal Successive Edges): Given a temporal weighted multidigraph $G = (V, E)$, the temporal successive edges of a node u at time t is defined as follows:

$$L_t(u) = \{e \mid Src(e) = u, T(e) \geq t\}$$

For instance, in Fig. 2, let $t = T(e_{\textcircled{5}})$, then $L_t(A1) = \{e_{\textcircled{5}}, e_{\textcircled{6}}, e_{\textcircled{10}}\}$. The set of temporal successive edges plays the role of candidates for walkers to select possible successors.

Apart from the temporal constraint, we further develop biased searching strategies by considering more detailed transaction information. For the Ethereum transaction network discussed here, we abstract the transaction time and amount as the temporal and weighted information of a TWMDG. Consider a random walk that just traversed edge e_{i-1} , and is now stopping at node v_i at time $t = T(e_{i-1})$. The next node v_{i+1} of the random walk is decided by selecting a temporally valid edge e_i . We describe different sampling biases by formulating the selection probability for each temporal successive edge $e \in L_t(v_i)$.

From the perspective of temporal domain, we consider both unbiased and biased sampling strategies as follows.

- **Temporal Unbiased Sampling (TUS):** This is the default setting in the time domain, which assumes that each temporal successive edge $e \in L_t(v_i)$ of node v_i at time t has the same probability to be selected:

$$P_T(e) = \frac{1}{|L_t(v_i)|}. \quad (1)$$

- **Temporal Biased Sampling (TBS):** For financial transaction networks, the similarity between accounts is time-dependent and dynamic. On the one hand, a pair of accounts with frequent interactions are supposed to have a stronger relationship. Therefore, we let $\eta_- : \mathbb{R} \rightarrow \mathbb{Z}^+$ be a function that maps the timestamps of edges to a descending ranking. In this case, each edge $e \in L_t(v_i)$ will be assigned with a selection probability:

$$P_T(e) = \frac{\eta_-(T(e))}{\sum_{e' \in L_t(v_i)} \eta_-(T(e'))}. \quad (2)$$

where $T(e)$ denotes the timestamp of the edge e . This sampling method biases the selection towards edges that are closer in time to the previous edge. On the other hand, sampling the interactions among accounts in a large time interval may also be important for different domains of networks for the purpose of preserving global similarity in time domain. For such scenarios, we propose another strategy that favors edges appearing later to the previous timestamp. Let $\eta_+ : \mathbb{R} \rightarrow \mathbb{Z}^+$ be a function that maps the timestamps of edges to an ascending ranking. The probability of selecting each edge $e \in L_t(v_i)$ can be given as:

$$P_T(e) = \frac{\eta_+(T(e))}{\sum_{e' \in L_t(v_i)} \eta_+(T(e'))}. \quad (3)$$

Apart from the transaction time, the amount values of the edges (edge weights) also plays an essential role in financial transaction networks. In the following, we present unbiased and biased strategies from a weighted domain.

- **Weighted Unbiased Sampling (WUS).** Similar to TUS, this is the default setting in the amount domain and each edge $e \in L_t(v_i)$ has the same probability to be sampled:

$$P_W(e) = \frac{1}{|L_t(v_i)|}. \quad (4)$$

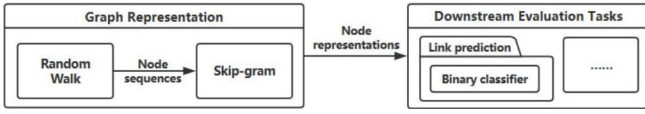


Fig. 5. The illustration of Ethereum transaction network modeling and analysis.

- **Weighted Biased Sampling (WBS):** As illustrated in the Introduction, the weight value of each transaction indicates the significance of interactions between the two accounts involved. For most instances, a higher value of transaction amount implies a larger similarity between the two accounts. Thus each edge $e \in L_t(v_i)$ can be assigned the selection probability:

$$P_W(e) = \frac{W(e)}{\sum_{e' \in L_t(v_i)} W(e')}. \quad (5)$$

To prevent the extreme situation where edges with small weights would never be sampled, we consider a linear mapping function to weaken the effects of edge weights. Thus we have

$$P_W(e) = \frac{\eta_+(W(e))}{\sum_{e' \in L_t(v_i)} \eta_+(W(e'))}. \quad (6)$$

Furthermore, we combine the aforementioned sampling probabilities from both temporal and weighted domains, i.e., P_T and P_W , by $P(e) = P_T(e)^\alpha P_W(e)^{(1-\alpha)}$ ($0 \leq \alpha \leq 1$) for $\forall e \in L_t(v_i)$. Here $\alpha = 0.5$ is the default value for balancing between time domain and amount domain. After the random walk generation with the temporal constraint and flexible biased strategies, the second part is an update procedure based on SkipGram [18], [19], which learns node representations as a maximum likelihood optimization problem.

VI. LINK PREDICTION FOR EVALUATION

In the previous section, we discuss different walking strategies for graph representation learning, which aims to preserve network properties for downstream tasks. Usually, link prediction and node classification are two common downstream tasks for evaluation of graph representation learning [13]. Due to the unavailability of ground truth of Ethereum accounts, we evaluate our temporal-random-walk-based graph embedding by link prediction on realistic Ethereum data, as Fig. 5 shows. In fact, link prediction is also a valuable issue in blockchain systems. A series studies on Ethereum have witnessed various illegal behaviors or scams such as phishing, Ponzi [20], money laundry [21], etc., and the temporal link prediction can help people track the real-time process of Ether flow or prevent illegal users' cashing process [22].

The task of link prediction aims to predict the occurrence of links in a given graph on the basis of observed information. In the temporal link prediction problem, unlike the static link prediction where links have no timestamp, we use the existing links in the past (50% of the earlier edges) as the training data to predict the occurrences of links in the future (the remainder). Shown in Table I, we randomly select center accounts and collect three subgraphs with different size ($K - in = K - out = K$) from Ethereum for learning node embedding vectors $\Phi(v)$ for $\forall v \in V$ via random-walk based graph representation learning. Next, we use a binary classifier for supervised link

TABLE I
STATISTICS OF DATASETS

Dataset	Address of Center Node	K	#Accounts	#Transactions
EthereumG1	0x51faeda318982f439e80012fb45d2b017ddccdbe	3	3,832	208,927
EthereumG2	0x5e247060f48eeb64367250ed03ff5091bba47fd1	4	10,628	208,533
EthereumG3	0x51faeda318982f439e80012fb45d2b017ddccdbe	4	26,175	677,785

TABLE II
PERFORMANCES OF DIFFERENT GRAPH MODELING AND WALKING METHODS FOR LINK PREDICTION

Metrics(%)	EthereumG1		EthereumG2		EthereumG3	
	AUC	AP	AUC	AP	AUC	AP
<i>Directed graph + unbiased walk</i>	82.71	76.69	85.91	82.13	79.92	77.72
<i>Directed graph + biased walk</i>	83.03	76.94	86.30	82.47	82.20	79.99
<i>TWMDG + unbiased temporal walk</i>	87.73	83.73	92.85	90.29	93.00	90.78
<i>TWMDG + biased temporal walk</i>	89.55	85.58	93.36	90.94	93.83	91.89

prediction, where node pairs (src, dst) existing in the training graph act as positive samples of the training set. We randomly sample an equal number of node pairs with no link as negative samples. Then we obtain features of a directed link from nodes v_i to v_j by concatenating their node representations, i.e., $F_{i,j} = [\Phi(v_i), \Phi(v_j)]$. If $i \neq j$, $F_{i,j} \neq F_{j,i}$. Finally, we train a support vector classifier to classify the links in the test set (50% of the later edges).

a) Settings: In the experiments, we compare our proposed walking strategies with directed DeepWalk [15] and directed node2vec [16] implemented by OpenNE [23]. For the random walk based representation methods, we have several hyperparameters: the node representation dimension d , the size of window k , the length of walk l , and walks per node r . In general, we set $d = 128$, and $k = 4$. Specifically, we set $r = 20$, $l = 10$ for EthereumG1, $r = 10$, $l = 10$ for EthereumG2, $r = 10$, $l = 20$ for EthereumG3. For the purpose of comparisons, we consider two widely discussed random walk based methods, namely DeepWalk [15] and node2vec [16], labeled as *Directed graph+unbiased walk* setting and *Directed graph+biased walk* settings in Table II, respectively. Besides, we apply temporal walk with TUS and WUS for *TWMDG+unbiased temporal walk* and temporal walk with TBS and WBS for *TWMDG+biased temporal walk* ($\alpha = 0.5$).

b) Results: Table II compares the performance of various methods on temporal directed link prediction in terms of Area Under Curve (AUC) and Average Precision (AP). Modeling the transaction data as a TWMDG without any bias overwhelmingly outperforms traditional directed graph. Temporal walk with biases of both time and amount domains leads to better performance than unbiased temporal walk.

c) Discussions of Results: The results manifest that the temporal information as well as the multiplicity characteristic of edges in TWMDG are very important and meaningful for analysis and understanding of financial transaction networks. The comparisons also demonstrate that the rich information

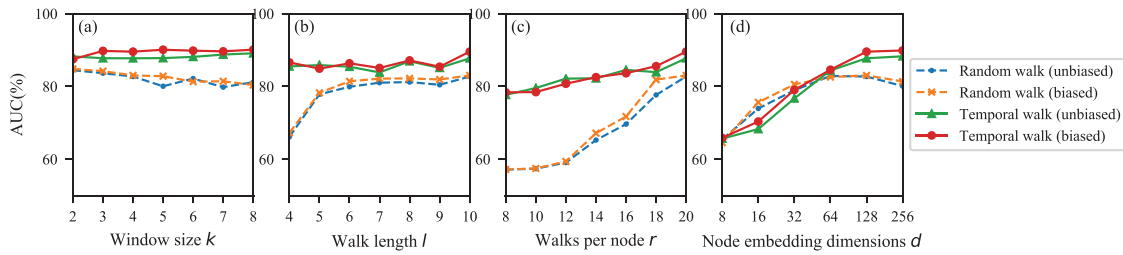


Fig. 6. Performance in terms of Area Under Curve (AUC) under varying hyperparameters, when (a) fixing $l = 10$, $r = 20$, $d = 128$, and varying k from 2 to 8; (b) fixing $k = 4$, $r = 20$, $d = 128$, and varying l from 4 to 10; (c) fixing $k = 4$, $l = 10$, $d = 128$, and varying r from 8 to 20; (d) fixing $k = 4$, $l = 10$, $r = 20$, and varying d from 8 to 256.

from time and amount domains does help us obtain a more comprehensive representation for predictive tasks.

d) Parameter Analysis: To further illustrate the superiority of biased temporal walk, we compare the performance on EthereumG1 with varying value of node representation dimension d , walk length l , walks per node r and window size k . Results in Fig. 6 demonstrate that: (1) Temporal walk with or without additional biases consistently outperform unbiased and biased random walk under different settings of k , l , r ; (2) Random-walk based methods are more sensitive to two hyperparameters, walk length l and walks per node r , while temporal-walk based methods can always achieve promising results with a wide range of both l and r ; (3) Interestingly, with an increase of d , the performance of temporal-walk based methods monotonically improves but performance of random-walk based methods degrades with d larger than 64, which implies that temporal walk can embed richer helpful information and thus requiring a larger value of d for data representation.

VII. CONCLUSION AND DISCUSSION

In this brief, we proposed to model the Ethereum transaction network as a new defined network model called temporal weighted multidigraph, and refine the definition of temporal walk considering the network dynamics and the multiplicity of edges to obtain graph representation more accurately. Experiments on the task of temporal link prediction demonstrated the effectiveness of the proposed framework. The limitations of the proposed approach is that the graph representation model based on temporal random walk is not an inductive method, which can not obtain the representations for the newly added nodes. For future work, we plan to develop more specific graph representation methods to investigate various predictive tasks on Ethereum and extend the current framework to analyze other large-scale temporal or domain-dependent networks.

REFERENCES

- [1] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.
- [2] A. Aldweesh, M. Alharby, E. Solaiman, and A. van Moorsel, "Performance benchmarking of smart contracts to assess miner incentives in ethereum," in *Proc. 14th Eur. Dependable Comput. Conf. (EDCC)*, Iasi, Romania, Sep. 2018, pp. 144–149.
- [3] W. Chen *et al.*, "Cooperative and distributed computation offloading for blockchain-empowered industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8433–8446, Oct. 2019.
- [4] C. Chen, J. Wu, H. Lin, W. Chen, and Z. Zheng, "A secure and efficient blockchain-based data trading approach for Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9110–9121, Sep. 2019.
- [5] Z. Chen, J. Wu, Y. Xia, and X. Zhang, "Robustness of interdependent power grids and communication networks: A complex network perspective," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 1, pp. 115–119, Jan. 2018.
- [6] G. Chen, Y. Lou, and L. Wang, "A comparative study on controllability robustness of complex networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 5, pp. 828–832, May 2019.
- [7] J. Zhou, X. Yu, and J.-A. Lu, "Node importance in controlled complex networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 3, pp. 437–441, Mar. 2019.
- [8] T. Chen *et al.*, "Understanding ethereum via graph analysis," in *Proc. IEEE INFOCOM*, Honolulu, HI, USA, Apr. 2018, pp. 1484–1492.
- [9] I. Alqassem, I. Rahwan, and D. Svetinovic, "The anti-social system properties: Bitcoin network data analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 21–31, Jan. 2020.
- [10] F. Spitzer, *Principles of Random Walk*. New York, NY, USA: Springer, 2013.
- [11] W. Liu and L. Lü, "Link prediction based on local random walk," *Europhys. Lett. Assoc.*, vol. 89, no. 5, Mar. 2010, Art. no. 58007.
- [12] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," in *Proc. Nat. Acad. Sci.*, vol. 105, no. 4, pp. 1118–1123, 2008.
- [13] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.
- [14] W. Chen and Z. Zheng, "Blockchain data analysis: A review of status, trends and challenges," *J. Comput. Res. Dev.*, vol. 55, no. 9, pp. 1853–1870, 2018.
- [15] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. ACM SIGKDD*, Aug. 2014, pp. 701–710.
- [16] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD*, Aug. 2016, pp. 855–864.
- [17] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, "Continuous-time dynamic network embeddings," in *Proc. WWW Companion Web Conf.*, Lyon, France, Apr. 2018, pp. 969–976.
- [18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: ArXiv:1301.3781.
- [19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2013, pp. 3111–3119.
- [20] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, Apr. 2018, pp. 1409–1418.
- [21] M. Möser, R. Böhme, and D. Breuker, "An inquiry into money laundering tools in the bitcoin ecosystem," in *Proc. APWG eCrime Res. Summit*, Sep. 2013, pp. 1–14.
- [22] L. Cai and B. Wang, "Research on tracking and tracing bitcoin fund flows," in *Proc. IEEE 4th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Chongqing, China, Dec. 2018, pp. 1495–1499.
- [23] THUNLP. (2017). *Openne: An Open Source Toolkit for Network Embedding*. [Online]. Available: <https://github.com/thunlp/openne>