Community Detection in Blockchain Social Networks

Sissi Xiaoxiao Wu, Zixian Wu, Shihui Chen, Gangqiang Li, and Shengli Zhang

***Abstract*—In this work, we consider community detection in blockchain networks. We specifically take the Bitcoin network and Ethereum network as two examples, where community detection serves in different ways. For the Bitcoin network, we modify the traditional community detection method and apply it to the transaction social network to cluster users with similar characteristics. For the Ethereum network, on the other hand, we define a bipartite social graph based on the smart contract transactions. A novel community detection algorithm which is designed for low-rank signals on graph can help find users' communities based on user-token subscription. Based on these results, two strategies are devised to deliver on-chain advertisements to those users in the same community. We implement the proposed algorithms on real data. By adopting the modified clustering algorithm, the community results in the Bitcoin network is basically consistent with the ground-truth of betting site community which has been announced to the public. At the meanwhile, we run the proposed strategy on real Ethereum data, visualize the results and implement an advertisement delivery on the Ropsten test net.**

***Index Terms*—Blockchain, Bitcoin, Ethereum, community detection, recommendation**

I. Introduction

EVER since Satoshi Nakamoto's Bitcoin white paper in the year of 2008[36], blockchain has been launched in many areas such as banking[20], network security, supply-chain management[48], internet-of-things (IoT)[12], financial cryptocurrency[34], serving as a decentralized ledger. Recently, governments in different countries begin to pay a huge attention to blockchain and put technologies involving blockchain into the cutting edge. In this work, instead of treating blockchain as a ledger, we try to study blockchain from a social media perspective. Specifically, we define the blockchain network as a decentralized social network, based on which we try different algorithms to analyze users' relationship underlying the ledger records. Our final goals are two-folded. First, we define different types of social networks for both Bitcoin and Ethereum. Second, by discovering users' clusters in the defined social graph we analyze users' behavior in both networks, and especially try to deliver advertisements in the Ethereum network.

Social network data is valuable as we can mine users' preferences from it and thus explore potential marketing. In traditional centralized social network, data is stored in a fusion center which is owned by the platform. Therefore, the platform monopolizes all data mining applications to make a big fortune. The blockchain network, however, decentralizes data among users and allows each user in the blockchain network fully access the whole piece of data and develops its own applications. Moreover, traceability ensured by the blockchain endorses the quality of the data, which further improves the efficiency of the data mining applications. The above nice properties make the blockchain as a social network promising in the future.

In this work, according to the way of recording the ledgers, Bitcoin and Ethereum are respectively defined as different kinds of social networks. In Bitcoin, every user can generate multiple private-public key pairs and the only purpose of transaction is to send BTC coins. The public key (also known as the address) is visible in the transaction block, as either the transaction input or the transaction output. Multiple input addresses and multiple output addresses may exist in the same Bitcoin transaction. In this context, one important pre-processing task of analyzing the ledger data in Bitcoin is to associate those addressees which belong to the same user and group them into a super-node in the social graph. This kind of operation is usually referred to "common spend" or "change address" heuristics. Based on the pre-processing results, we define the Bitcoin social network as an undirected graph where each super-node corresponds to a node in the graph and the edge weight of any two nodes in the graph is defined by historical coin-based transactions between any two super-nodes. Then we propose a specific clustering algorithm, which originated from the spectral clustering algorithm, to the Bitcoin social graph to find communities.

Ethereum, published in 2014 by Vitalik Buterin and launched in 2015, is the world's leading programmable blockchain as it has added a self-enforcing piece, i.e., the so called *smart contract*, which ensures a coordinated and enforced agreement between network participants by means of an organization or to create tokens[9]. In Ethereum, a transaction has only one input address and one output address and thus we can bypass the super-address pre-processing. Unlike Bitcoin, users in Ethereum interact with each other by not only a direct ETH coin transaction but also the smart contract transaction. In this work, we focus on the smart contract transactions and define the Ethereum social network as a bipartite social graph. Particularly, we are interested in those smart contract transactions specific for initial coin offering (ICO) events. Based on their bipartite graph, we introduce an effective community detection algorithm for low-rank signals to group users into different clusters[57]. These results can

This work is supported by the National Natural Science Foundation of China under Grant 61701315.

S. X. Wu, Z. Wu, S. Chen, G. Li, and S. Zhang are with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. E-mails: {xxwu.eesissi, zsl}@szu.edu.cn, {1900432037, 2172262956, ligangqiang2017}@emai.szu.edu.cn.

be further used for other purposes. For example, in both Bitcoin and Ethereum networks, if one person creates two user accounts, it is generally difficult to associate those accounts to the same person due to pseudo-anonymity. Our community detection algorithms may help do the association by analyzing the accounts' preferences, given that two accounts for the same user should share similarities. Also, the communities results may also be used to label users' potential preferences and thus provide a targeted referral service in blockchain.

A. Related work

There has been a branch of works in the literature on analyzing the Bitcoin transaction data and most of them focus on two issues: anonymization and de-anonymization. The review on these two issues can be found in Refs. [13, 44, 46, 50]. Therein, shared coin and send mixers in Refs. [5, 7, 30, 31, 60] are two basic anonymization approaches, while other approaches such as fair exchange[5], transaction remote release[51] and zero cash[49] are also popular. In this work, our main focus is on de-anonymization. In the literature, there are many approaches for implementing de-anonymization. For example, an early work in Ref. [5] first brought up the notion of "change address" and people realized that one can use the heuristics to associate addresses that involving common spending and one-time change. This approach is widely used as the first step to process Bitcoin data[2, 11, 17, 18, 21, 32, 45]. There are also more advanced approaches. In Refs. [6, 25, 62], the authors tried to de-anonymize user's identity of Bitcoin by linking the Bitcoin address with IP address. Ref. [53] summarized prior approaches of clustering addresses and implemented a modular framework to group users and classify them with different labels. Sometimes, off-chain information is also useful. For example, in Refs. [17, 44] the authors proposed to use off-chain information to guide different clustering models with a purpose of reducing the algorithm complexity. Ref. [15] and Ref. [35] proposed novel methodologies for analyzing Bitcoin users based on the observation of Bitcoin transactions over time.

In this work, we try to deanonymize a blockchain network by finding users' communities. Some prior works have been done for Bitcoin. The idea of treating Bitcoin as a social network has appeared in Ref. [52]. Therein, people usually used the notions of "user graph" and "transaction graph", and some analysis based on these two graphs has been elaborated in Ref. [44]. Ref. [29] studied the anonymity for Bitcoin by analyzing the transaction graph with the help of public data. Ref. [19] studied how different features of the date influence communities results on the "transaction graph" based on the ground truth of some known hack subnetworks. Authors in Ref. [43] and Ref. [22] extracted various features from these two graphs and pointed it out that features on the graph are crucial for the analysis results. Ref. [45] showed that a two-party community detection based on normalization mutual information could be used to re-identify users in Bitcoin. Our community detection approach on the Bitcoin data is based on the above works. Specifically, our approach has the following properties: 1) we study the "user graph" based on the super-address which is associated with addresses that involve common spending and one-time change; 2) we use historical BTC coin amount as the key feature to perform the community detection algorithm; 3) a modified clustering method is proposed for the Bitcoin social graph to find communities.

The above approaches are effective for Bitcoin while it is difficult to be directly carried forward to the Ethereum network, as Ethereum is mechanically different from Bitcoin. In Ethereum, there are two types of accounts: externally-owned accounts (EoAs) and contract accounts (CAs). In Refs. [10, 26], the authors pointed it out that existing methods such as discovering IP addresses and Bitcoin addresses clustering usually do not fit the Ethereum network due to the differences between both networks in the volatility of entry nodes and the way transactions are handled. Some works tried to use traditional clustering methods for Ethereum, such as support vector machine (SVM) and k-means in Ref. [8], the standard k-means in Refs. [23, 42], long short-term memory (LSTM) and cnonvolutional neural network (CNN) models in Ref. [23], affinity propagation k-medoids in Ref. [40], k-means clustering, agglomerative clustering, Birch clustering in Refs. [41, 54] and Neo4j in Ref. [10]. However, they basically equally treat EoAs and CAs as nodes in the transaction graph[10, 42, 54], while these two types of accounts are essentially different. Some works in Refs. [27, 40] also used side information to analyze the on-chain transactions. Therein Ref. [27] utilized the smart contract codes to analyze the smart contract nodes in the transaction graph. It is worth mentioning that our community detection approach in a bipartite graph differs from traditional one in a bipartite graph as we treat nodes on both sides of the bipartite graph as two parties of nodes, therefore utilizing the connections between those two parties to cluster nodes in one party, while the traditional approach usually treats all the nodes in both sides equally and find communities over all the nodes in both sides [1, 4, 61]. Our work on Ethereum data differs from the above work in five aspects: 1) we analyze the on-chain data without any off-chain side information; 2) we separately treat EoAs and CAs as different nodes and put them on two sides of a bipartite social graph; 3) we target on smart contract transactions which involve ICO events; 4) we apply a novel low-rank community detection algorithm based on graph signal processing (GSP) on the bipartite graph; 5) we utilize the clustering results to deliver on-chain advertisement in Ethereum.

The rest of the paper is organized as follows. In Section II, we show an example to define the Bitcoin social graph and then apply a novel clustering method on this graph. Based on this result, Section III demonstrates the difference between the Bitcoin social graph and the Ethereum social graph. We then define the bipartite Ethereum social graph and utilize a particular method to find communities on it. Simulation results are illustrated in Section IV where both Bitcoin data and Ethereum data are analyzed and compared. Moreover, we also test the on-chain advertisement for Ethereum based on the clustering results. We conclude and further discuss with simulation results in Section V.

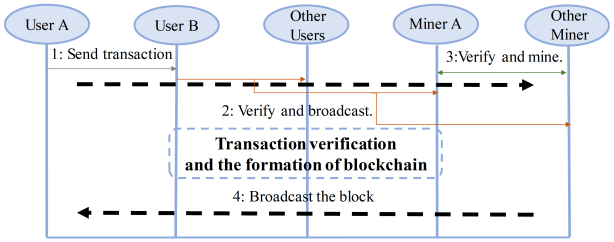

Fig. 1: The consensus process in Bitcoin network.

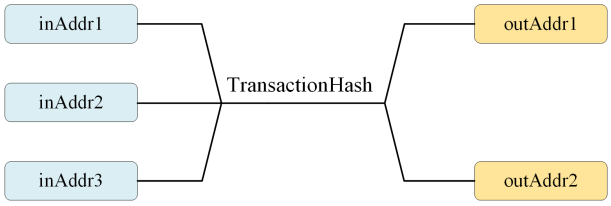

Fig. 2: The Bitcoin UTXO model: "inAddr" and "outAddr" are the abbreviations of the input and output addresses. We use the "transactionHash" to denote the transaction including these addresses.

II. Bitcoin Transactions Analysis

To start our work, we first introduce how Bitcoin verifies and records transactions. As shown in Fig. 1, in a Bitcoin network, when one node initiates a transaction, the transaction information will be signed and packaged, thus broadcasting to other nodes. Those nodes who receive the transaction will verify its legality and then help broadcast the verified transaction message. During the process of broadcasting, some of the received nodes are miners, who not only bear the responsibility of broadcasting transactions, but also undertake the task of "mining". The miner who has successfully mined by solving a difficult mathematical problem will get the right to write the ledger and add all transactions they have verified. When most of the nodes in the entire network agree on the same transactions, these transactions are recorded in the block.

A. The Bitcoin Social Network

As mentioned in the previous section, the only purpose of transaction in Bitcoin is to transfer BTC coins. To achieve this goal, each user generates a key pair (represented as the addresses in the transaction) to join the Bitcoin network and transfer BTC coins based on the so-called unspent transaction output (UTXO) model[36]. UTXO can be seen as an abstraction of electronic money, representing a chain of ownership implemented as a chain of digital signatures. In Fig. 2, we show a basic structure of a Bitcoin transaction where we can find multiple addresses in the input and output fields. Every address could contain multiple UTXO, wherein UTXO in the input addresses are consumed while in the output addresses are created.

A basic idea to define the Bitcoin social network is to let each address one-to-one correspond to a node in the social graph. However, a crucial problem here is that one user usually possess multiple addresses. Given that there are so many addresses in the blockchain, the dimensionality of the social network could be huge. To reduce the size of the graph, we try the following way to associate multiple addresses (key pairs) to a super-address in the social graph: 1) multiple input addresses of a transaction; 2) those bitcoin users that have a common change address. More details for such operations will be shown in the numerical experiments. After such a pre-processing, we define a social network where each node corresponds to a processed super-address.

B. Community Detection for Bitcoin

To well define the graph, we need to specify the edge weight between any two nodes (super-addresses). In fact, the edge weight between any two nodes could be defined following criterion in Refs. [19, 22, 43]. Herein, we extract features from the total transaction amount and set them as the edge weight. Then, we run a clustering method which is modified from the spectral clustering algorithm[39, 58] to cluster this Bitcoin social graph. Specifically, we denote the social graph by $\boldsymbol{G(V,E)}$, where $\boldsymbol{V}$ denotes the user nodes $(v_1, v_2, ...v_n)$ and $\boldsymbol{E}$ the edges. We let $\boldsymbol{W}$ denote the weight matrix where each entry w_{ij} is the edge weight between node v_i and node v_j. In this paper, we let a_{ij} be the historical total transaction amount between node i and node j and

$$w_{ij} = a_{ij} / \max_{i,j}\{a_{ij}\}.$$

Apparently, we will thus have an undirected graph with $w_{ij} = w_{ji}$. We then apply the following clustering algorithm in Algorithm 1 to find communities in the graph.

Remark 1: It is worth noting that the above clustering algorithm is quite similar to the well known spectral clustering algorithm except for that the similarity matrix is replaced by the weight matrix in our algorithm. This modification is meaningful since we re-define the "similarity" as that two nodes have a significant transaction relationship. This redefinition can help cluster users that have more connections with each other.

Algorithm 1 Clustering the Bitcoin social graph

Input: a set of nodes $\boldsymbol{V} = \{v_1, \cdots, v_n\}$, the number of clusters k and weight matrix $\boldsymbol{W}$ for all nodes in $\boldsymbol{S}$

Step 1: Define $\boldsymbol{D}$ to be the diagonal matrix whose (i,i)-element is the sum of $\boldsymbol{W}$'s i-th row. Letting $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, construct a matrix $\bar{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$.

Step 2: Find $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k$, the k largest eigenvectors of $\bar{\boldsymbol{L}}$ (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k]$ by stacking the eigenvectors in columns.

Step 3: Form the matrix $\boldsymbol{Y}$ from $\boldsymbol{X}$ by renormalizing each of $\boldsymbol{X}$'s rows to have unit length$(i.e., \boldsymbol{Y}_{ij} = \boldsymbol{X}_{ij}/(\sum_j \boldsymbol{X}_{ij}^2)^{1/2})$.

Step 4: Treating each row of $\boldsymbol{Y}$ as a point in $\mathbb{R}^k$, cluster them into k clusters via k-means or any other algorithm that attempts to minimize distortion.

Step 5: Finally, assign the original point $\boldsymbol{v}_i$ to cluster j if and only if row i of the matrix $\boldsymbol{Y}$ was assigned to cluster j.

Output: Partition nodes in $\boldsymbol{V}$ into k communities.

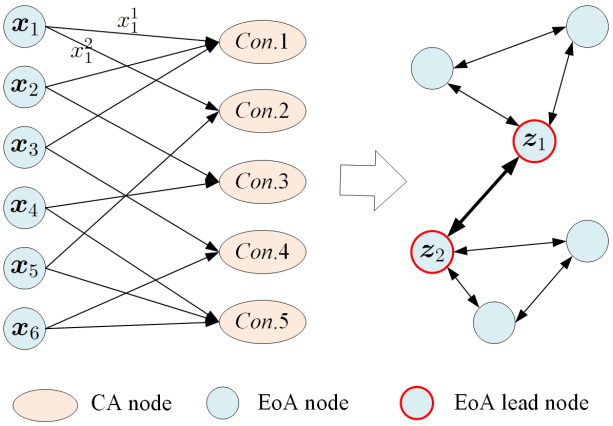

Fig. 3: Construct a social network based on different ICO tokens: x_i^t represents user i's balance for token t.

III. The Ethereum Social Network

As a blockchain network, Ethereum is different from Bitcoin in many aspects[3, 9, 59]. Particularly, Ethereum is not only a platform for providing ETH coin transactions, but also a programming language that enables users to build and publish distributed applications via the smart contract. In Ethereum, each user generates a pair of asymmetrically encrypted public key and private key to join the network. Each public key could be considered as a node in our Ethereum social network. In Ethereum, there are two types of accounts: externally-owned accounts (EoAs) and contract accounts (CAs). EoAs are considered as individual users in the external world while CAs are the contracts that could connect EoA users. Both EoAs and CAs are presented by unique hash addresses.

According to the properties of Ethereum, we define its social network as a bipartite graph, where EoA nodes and CA nodes are put into two sides of the graph; seen in Fig. 3. Each EoA node has their attention on different CA nodes. For example, in Fig. 3 (left), supposing that we have N EoA nodes and T CA nodes in the graph, for an EoA node $i \in \{1, ..., N\}$, we define $\boldsymbol{x}_i = [x_i^1, x_i^2, ..., x_i^T]$ and x_i^t is EoA node i's attention on CA node t. Herein, the CA nodes could be any smart contract in Ethereum. A typical example could be a token created by an ICO event, where a possible choice of x_i^t could be user i's transaction amount on token t. We remark that in this paper we use ICO events and token transaction amounts as features to define the bipartite graph, while it actually could be any other type of smart contracts which connect EoA nodes.

A. Algorithms and Strategies

Now our purpose is to perform a community detection on this bipartite graph and group all EoA nodes into clusters. In this subsection, we adopt the low-rank community detection algorithm in Ref. [57] to cluster the EoA nodes in the bipartite graph. The idea is to assume that all EoA nodes form a low-rank social sub-graph, where some lead EoA nodes will decide other nodes' attention on CA nodes. In this spirit, we partition the EoA node set of the bipartite graph into subsets with high edge densities. This could be done by applying a clustering algorithm on the low-rank output covariance matrix of the observed graph signal at EoA nodes. To proceed it, we regard this community detection problem as a problem of GSP, wherein the input of the graph $\boldsymbol{z} \in \mathbb{R}^R$ is on the EoA lead node (see Fig. 3) and it goes through a filter $\mathcal{H}(\boldsymbol{S})$:

$$\mathcal{H}(\boldsymbol{S}) := \sum_{\ell=0}^{L-1} h_\ell \boldsymbol{S}^\ell = \boldsymbol{V}(\sum_{\ell=0}^{L-1} h_\ell \boldsymbol{\Lambda}^\ell)\boldsymbol{V}^H \tag{1}$$

where $\boldsymbol{S}$ is the graph Laplacian matrix, L is the degree of the filter and R is the number of lead node. The output signal $\boldsymbol{x} \in \mathbb{R}^N$ is defined on all the EoA nodes and it is generated by

$$\boldsymbol{x} = \mathcal{H}(\boldsymbol{S})\boldsymbol{z} \ . \tag{2}$$

Herein, $\boldsymbol{V}$ and $\boldsymbol{\Lambda}$ are from a SVD decomposition of $\boldsymbol{S}$. The above equation means that in our graph model, the opinion of the EoA lead node decides all nodes' status. Based on the above model, the graph signal observed at all the EoA nodes can be expressed as

$$\boldsymbol{y}^t = \boldsymbol{x}^t + \boldsymbol{w}^t \text{ and } \boldsymbol{x}^t = \mathcal{H}(\boldsymbol{S})\boldsymbol{z}^t, \quad t = 1, ...T \tag{3}$$

where $\boldsymbol{y}^t$ is the observation of the graph signal which represents EoA nodes' attention on CA node t in our problem setting and $\boldsymbol{w}^t \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ is the noise. Notice that the input signal $\boldsymbol{z}^t$ is applied on only a subset of R EoA lead nodes and thus the number of variations in the excitation signal is limited to R mode. To further explore the graph structure, we let

$$\boldsymbol{C}_z = \mathbb{E}[\boldsymbol{z}^t(\boldsymbol{z}^t)^\mathsf{T}] = \boldsymbol{B}\boldsymbol{B}^\mathsf{T}, \tag{4}$$

where $\boldsymbol{B} \in \mathbb{R}^{N \times R}$ with $R < N$. Then, we can recover the community structure in $\boldsymbol{S}$ by applying Algorithm 2 on the empirical sampled covariance of the observed signal $\boldsymbol{y}^t$:

$$\hat{\boldsymbol{C}}_x = (1/T)\sum_{t=1}^{T} \boldsymbol{y}^t(\boldsymbol{y}^t)^\mathsf{T}, \tag{5}$$

which is an estimate of $\boldsymbol{C}_x = \mathcal{H}(\boldsymbol{S})\boldsymbol{B}\boldsymbol{B}^\mathsf{T}\mathcal{H}(\boldsymbol{S})^\mathsf{T}$.

An illustration of the algorithm model for Ethereum is depicted in Fig. 4. In practice, $\hat{\boldsymbol{C}}_x$ could be obtained by observing the graph signals from many instances t. For example, in the Ethereum social network, we consider each instance t as one CA node in the bipartite graph. Thus, we obtain the graph signal $\boldsymbol{y}^t$ by observing EoA nodes' attention on CA nodes and utilize it to detect communities of EoA nodes. For example, we could consider users' transaction amount on different tokens as their attention on such tokens.

Algorithm 2 Community detection from low-rank excitation

1: Input: Graph signals $\boldsymbol{y}_{t=1}^T$; desired number of clusters K.
2: Use $\boldsymbol{y}_{t=1}^T$ to compute the sample covariance $\hat{\boldsymbol{C}}_x$ as in (5).
3: Find the K eigenvectors to $\hat{\boldsymbol{C}}_x$ associated with the largest K eigenvalues. Denote the set of eigenvectors as $\hat{\boldsymbol{P}}_K \in \mathbb{R}^{N \times K}$.
4: Perform K-means clustering[56], which optimizes:

$$\min_{\mathcal{C}_1,...,\mathcal{C}_K} \sum_{i=1}^{K} \sum_{j \in \mathcal{C}_i} \|\hat{\boldsymbol{p}}_j - \frac{1}{|\mathcal{C}_i|} \sum_{q \in \mathcal{C}_i} \hat{\boldsymbol{p}}_q\|_2^2 \quad s.t. \quad \mathcal{C}_i \in V$$

where $\hat{\boldsymbol{p}}_j := [\hat{\boldsymbol{P}}_K]_i, : \in \mathbb{R}^K$.Let the solution be $\hat{\mathcal{C}}_1, ..., \hat{\mathcal{C}}_K$.
5:Output: Partition of $\boldsymbol{V}$ into K communities,$\hat{\mathcal{C}}_1, ..., \hat{\mathcal{C}}_K$.

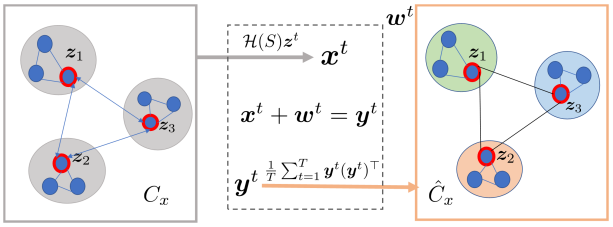

Fig. 4: An algorithm model for Ethereum.

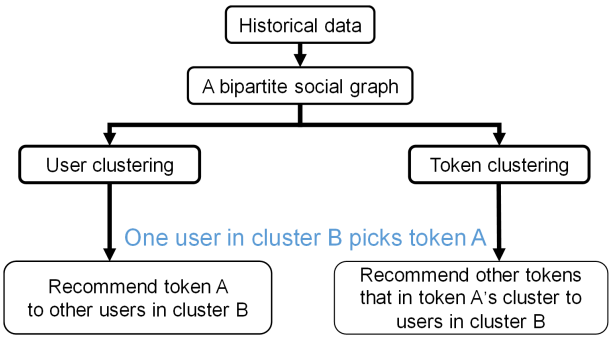

Fig. 5: A flow chart for the recommendation system.

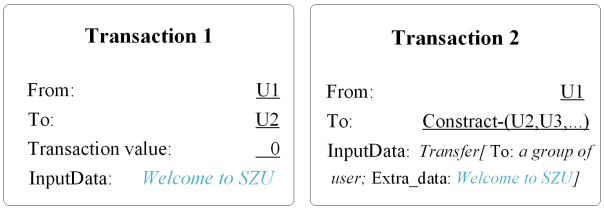

Fig. 6: Coin transaction (left) and smart contract transaction (right). Ui denotes User i.

B. Group Tokens by Users Subscription

In previous discussion, we discuss how to find communities of EoA nodes by using graph signal processing. In fact, this process can be reversed to cluster tokens by users' subscription. That is, we exchange the position of the EoA nodes and CA nodes and observe the graph signal at each CA node, which represents all EoA nodes' attention on a specific CA node. This equals to transposing $\hat{\boldsymbol{C}}_x$ in Algorithm 2 and performing the same clustering method on the new $\hat{\boldsymbol{C}}_x$. Numerical results are shown in Section IV-B.

At the end of this part, two remarks are in order. First, the relationship between user clustering and token clustering is analog to that between the user-based collaborative filtering and the item-based collaborative filtering. Second, the token clustering result can also be used to recommend tokens to EoA nodes. A flow chart for the recommendation system is shown in Fig. 5. We will introduce the detailed recommendation process in the next subsection.

C. Advertisement Strategies

The proposed community detection algorithms can help find EoA users sharing the same interests on CA nodes (by user clustering), as well as CA nodes favored by groups of EoA nodes (by token clustering). We then discuss 1) how to deliver

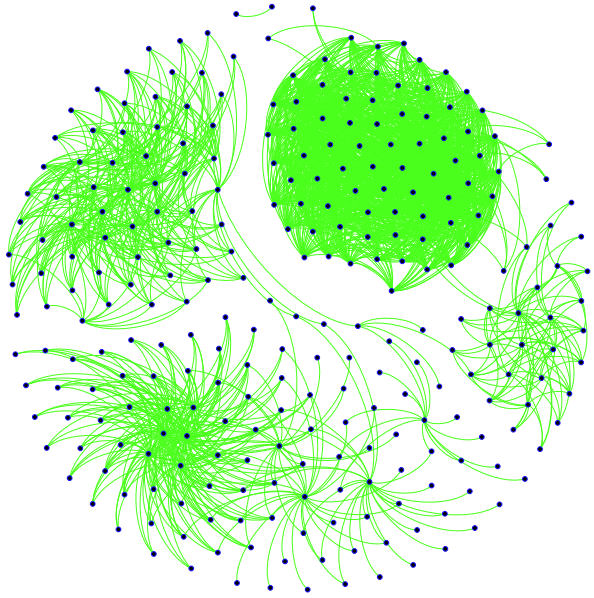

Fig. 7: Graph representation after associating the common spend addresses.

advertisements to a user whose community members have shown interest on a specific token, and 2) how to recommend other tokens to a user who has shown interest on a specific token. Specifically, we may resort to the "InputData" field in the transaction script to serve our purpose. In Fig. 6, we show two types of transactions in Ethereum. Transaction 1 is an ETH coin transaction (left) and Transaction 2 is a smart contract transaction (right). For both transactions, there is an "InputData" field in the script which can be used to run functions or send messages. We therefore design two on-chain advertisement strategies. One approach is to send a small amount ETH coin (could be zero) to the target user and attach a recommendation message in the "InputData" field in this coin transaction. This implementation can only be done in a one-to-one manner and one has to cost some gas to send the message. Another approach is similar to the so-called "airdrop"[14, 33], wherein new ICO project distributes part of their tokens for free to a community to advertise their ICO project. "Airdrop" could be done via smart contract in a group message manner and no extra ETH coin is consumed except for the gas. Notice that to successfully deliver the message, one has to negotiate with the wallet company to register their token in the target user's list. Otherwise, users can not see the new-added token, as well as the advertisement message. We remark that our design differs from the original "airdrop" in two aspects: 1) we have resorted to a community detection to target potential users; 2) we utilize the "InputData" field to send the advertisement message.

IV. Dataset and Numerical Results

In this section, numerical results are provided for both Bitcoin data and Ethereum data. The data sets are downloaded from the actual blockchain systems while we also utilize some known pre-processed results based on the real blockchain data.

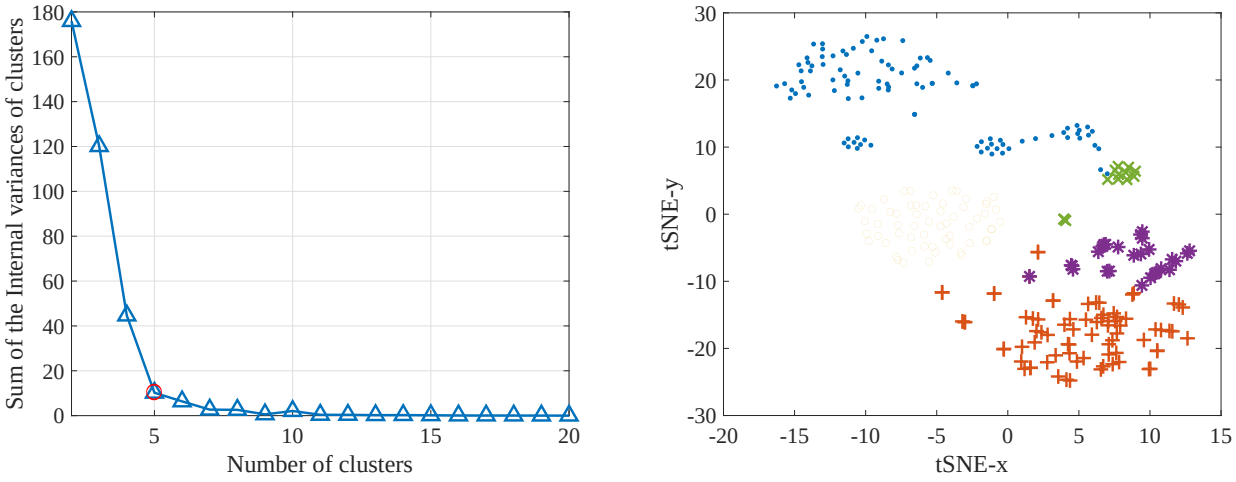

Fig. 8: Bitcoin clustering: (Left) The k-var curve, where the number of token's clusters is found by the Elbow method; (Right) t-SNE Clustering results.

A. Numerical Results for the Bitcoin Data

The Bitcoin data we studied comes from the website *http://vo.elte.hu/bitcoin*, where the raw Bitcoin data is processed and compressed into several documents; more details could be found in Ref. [16]. In our experiment, we used the document "txhash.txt" which is a list of transaction IDs (indexed by the website) and hash pairs to record the hash for each transactions in chronological order. We intercepted the block data from block number $250,000$ to $252,000$, whose time interval is between "2013-08-03 12:36:23" and "2013-08-13 18:11:30". By searching transaction hash in "txhash.txt" we found that these transaction IDs range from $21,490,941$ to $22,003,698$. With these IDs, we can search documents "txin.txt" and "txout.txt" to find the input addresses and output addresses. Herein, "txin.txt" records each transaction's input addresses with the amount of Satoshis[1] and "txout.txt" records each transaction's output addresses with the amount of Satoshis. Within this time interval, we can extract in total $512,756$ transactions involving $515,765$ addresses. It is worth noting that at this moment, the addresses may be duplicated.

Our next step is to pre-process the data by associating the addresses using the heuristic of "common spend" and "change address". To process the data by "common spend", we utilized the data set "contraction.txt" from the website *http://vo.elte.hu/bitcoin*, which is a list of addresses possibly belonging to the same user. The basic idea of this process is that any two input addresses which belong to the same "user" appear as inputs in the same transaction at least once. After this processing, we have in total $132,431$ transactions and $65,811$ identified unique users (super-addresses) left. To better analyze the details of some key users, we assume that the "change address" is used rarely and thus we can eliminate the addresses whose occurrence (appears in the transaction input or output) is less than 30. This process significantly reduces the size of the graph to 3930 transactions and 279 users. Notice that some of the users are the combinations of several common spend addresses, and others are the change addresses which appeared in the output of transactions but never appeared in the inputs of a transaction. In Fig. 7, we plot the graph representation by using the Gephi software[24], where the continuous graph layout algorithm ForceAltas 2 is adopted to visualize the graph. Note that herein we do not consider to use any features to define the edge weights. The edge weight is either '0' (no transaction) or '1' (with transaction). From the plot we see that after associating common spend, the graph is well clustered while the dimension of the graph is still large.

We then evaluate and visualize the community detection results based on the social graph we have defined for the Bitcoin data. First, we run the Elbow method in Ref. [55] to determine the optimal number of clusters k. Fig. 8 (left) shows that the "inflection point" is $k = 5$ and thus we consider there are 5 clusters in our example. Interestingly, this is also roughly consistent with the results in Fig. 7 although they have defined different edge weight. We then run Algorithm 1 under a random initialization with $k = 5$ and find that the number of nodes in each cluster are 101, 78, 60, 27, and 13, respectively. A 2D visualization of the clustering results are shown in Fig. 8 (right), where a machine learning algorithm called t-SNE[28] is used for a nonlinear dimensionality reduction. The results show that the target users are indeed clustered in the compressed 2D space. We use two parameters to evaluate the community results. One is the Silhouette score, which combines the two factors of cohesion and resolution to evaluate the clustering results[47]. The other one is the Modularity score which is usually used to measure the structural of network communities[38]. In this experiment, we have the Silhouette score 0.5871 and theModularity score 0.3733. Normally the Silhouette score is at a range of $[-1.0, 1.0]$ and the range of the Modularity score is $[-0.5, 1.0]$. The more two scores approach to 1, the better quality of the network partition. The Modularity score around $0.3 \sim 0.7$ is considered as a good clustering result[37]. We track the nodes of the gambling website and find that the gambling website nodes and other 55 nodes that had

[1]The satoshi is currently the smallest unit of the bitcoin currency recorded on the block chain.

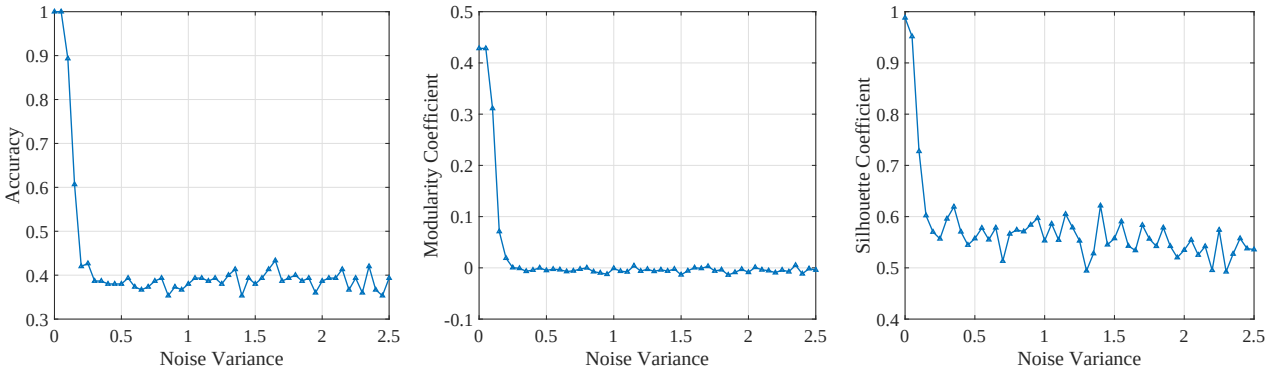

Fig. 9: How noise affects the recovery of the communities.

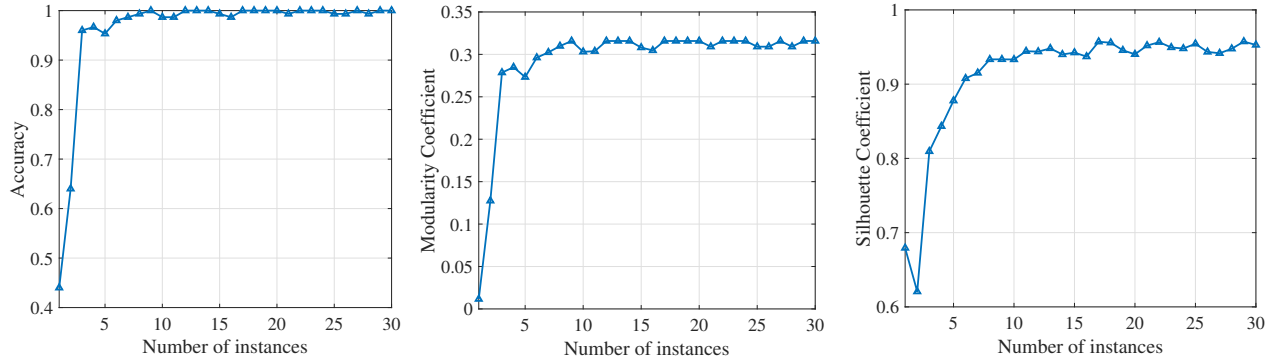

Fig. 10: Number of instances versus the recovery accuracy.

transactions with the gambling nodes have been all clustered into the same cluster. This cluster has in total 101 nodes.

B. Numerical Results for the Ethereum Data

1) Synthetic data test

To verify our model, we first generate synthetic data to test how the proposed method works for a known graph with given input signal. Specifically, a graph $\mathcal{G}(N, K, P_a, P_b)$ is generated where N is the number of nodes, K is the number of communities, P_a is the probability of node connection within the community and P_b is the probability of node connection between communities. We then define the graph filter as $\mathcal{H}(\boldsymbol{S}) = (1-\alpha\boldsymbol{S})^{L-1}$ where $\boldsymbol{S}$ is the Laplace matrix of $\mathcal{G}$ and L is the order of the graph filter. The graph signal $\boldsymbol{y}_t$ is thus generated following (3) where we set $\boldsymbol{z} = \boldsymbol{B}\boldsymbol{\alpha}$ with $\boldsymbol{B}$ having R non-zero rows and each row having $[Rd/N]$ ones where d is the degree of the graph and we generate $\boldsymbol{\alpha} \sim \mathcal{N}(0, \boldsymbol{I})$ to get different instances. Given the structure of the graph and input signal $\boldsymbol{z}$, we use three different parameters to evaluate the recovery of the graph. Herein, the Silhouette score and Modularity score have defined before, and the recovery rate is defined as the percentage of nodes that are recovered in the correct cluster.

Fig. 9 shows how noise corrupted in the observed graph signal effects the recovery accuracy. Herein, we set $N = 150, P_a = 0.89, P_b = 0.11, R = 15$ and generate 1000 instances as the input signal. We vary the noise variance to see the recovery performance. The results tell that in the noise-free case, we can 100% recover the communities, while as the noise level increases, the recovery accuracy deteriorates. In Fig. 10, we set $N = 150, P_a = 0.8, P_b = 0.2, R = 15$ and assume the noiseless case. We vary the number of instances of the input signal to see how it affects the recovery accuracy. Apparently, the results tell that we need to observe sufficient instances to recover the community information. The synthetic data test in Fig. 9 and Fig. 10 demonstrates that the proposed model can find communities given that the noise in the observation model is not so heavy and we have data for sufficient instances. Motivated by this, in the real data test, we utilize user-token subscription information in the Ethereum network to find the EoA users' communities.

2) Real data test

In this part, we focus on the token transactions in the Ethereum network to show a toy example of the data processing. To set up a meaningful example, we screen out the user-token pairs by three steps. In step one, we pick 20 top market capitalization tokens from website *https://etherscan.io* on the date of July 1st 2019, and record the top 100 user addresses of each token. In step two, we extract aforementioned users' transactions, recorded all the tokens they have owned at the current moment (that would be much more than 20 tokens mentioned above). After this operation, there are 141 users with 1837 tokens in total. Then, in the last step, to maintain an appropriate level of attention for each token, we first delete those users who have subscribed less than 20 tokens and then

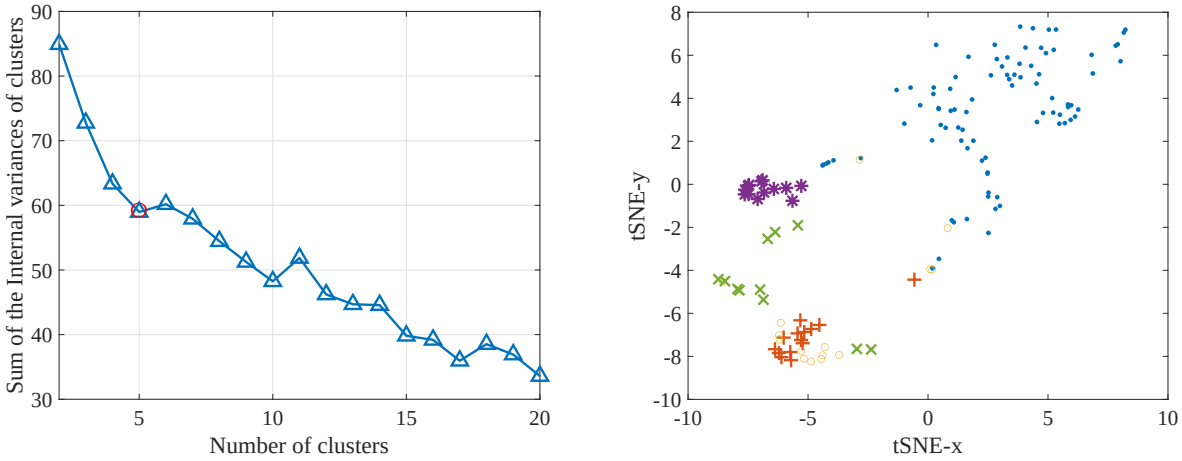

Fig. 11: User clustering: (Left) The k-var curve, where the number of user's clusters is found by the Elbow method; (Right) t-SNE Clustering results.

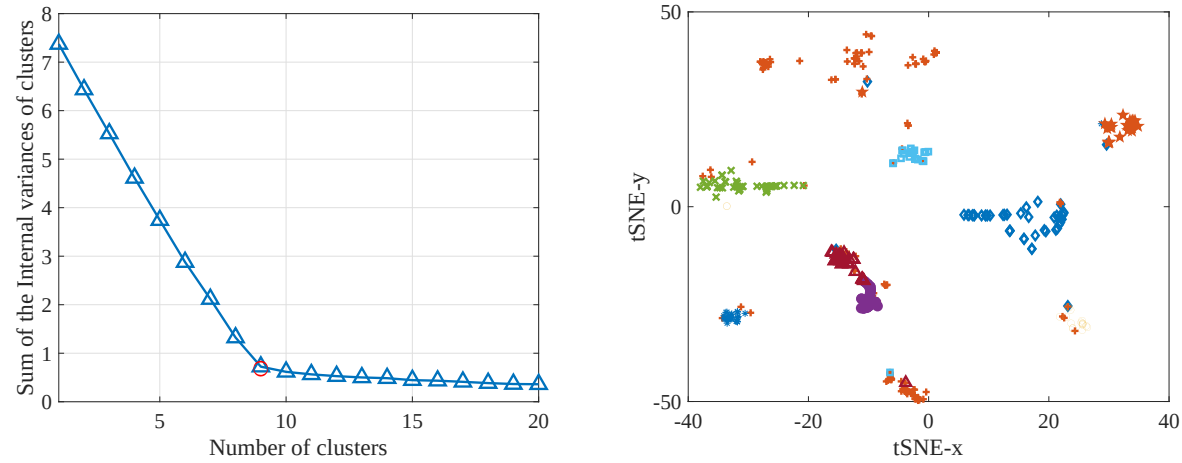

Fig. 12: Token clustering: (Left) The k-var curve, where the number of token's clusters is found by the Elbow method; (Right) t-SNE Clustering results.

delete those tokens which have been subscribed by less than 60 EoA users. In the end, we have selected 141 users who have focused on 21 tokens to form the bipartite graph. This graph corresponds to a user-token matrix $\boldsymbol{A} \in \mathbb{R}^{141\times 21}$ with entry A_{ij} denoting user i's transaction amount on token j. To further process $\boldsymbol{A}$, we unify the token value to ETH value by multiplying their currency to ETH on July 1st 2019. Therefore, in the bipartite graph, the edge weight between the user and the token is defined as the amount of ETH value this user has owned on this token at the current moment. We remark that such a data process can be applied to a much bigger size of data and we can thus analyze a large size graph at one shot. Herein we focus on a small size graph so that we can have a more clear description of the process.

• *The user clustering:* After a row normalization and a column normalization to matrix $\boldsymbol{A}$, we apply Algorithm 2 to perform the community detection. In particular, the Elbow method is used to determine the optimal number of clusters for clustering[55]. Fig. 11 (left) shows that the "inflection point" is $k = 5$ and thus we consider there are 5 clusters in our toy example. Fig. 11 (right) shows a 2D visualization of the clustering results, where t-SNE is used for a nonlinear dimensionality reduction. We also calculate the Silhouette score and the Modularity score for this case, which are 0.6586 and 0.5999, respectively.

• *The token clustering:* For the previous user clustering process, graph signals $\boldsymbol{y}_{t=1}^{T} \in \mathbb{R}^{N}$ represents the subscription of all users on token-t $(t = 1, ..., T)$, which can be considered as the features of EoA users. To proceed the token clustering, we redefine $\boldsymbol{y}_{n=1}^{N} \in R^{M}$ that can be interpreted as user-n $(n = 1, ..., N)$'s subscription on all tokens. We pre-process the data in a similar way as that in the user clustering process and delete the tokens which are not subscribed by any users. The new bipartite graph is thus based on the token-user matrix $\boldsymbol{A} \in \mathbb{R}^{1811\times 141}$. The sample covariance $\hat{\boldsymbol{C}}_x$ can also be calculated as in (5). We then apply Algorithm 2 to cluster the tokens. Through the Elbow method, Fig. 12 (left) shows that the "inflection point" of the token-user graph is $k = 9$. Thus we consider there are 9 clusters in the token-user graph. By the use of t-SNE [28], the clustering results are shown in Fig.

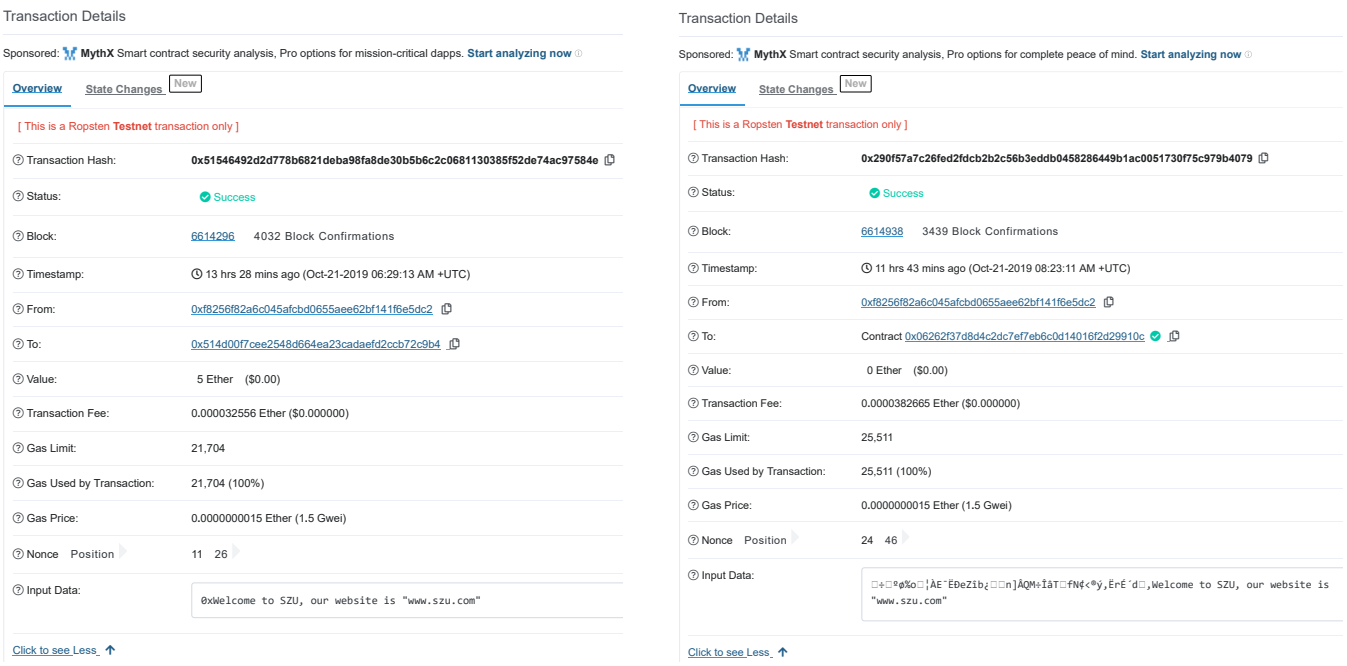

Fig. 13: View advertisements in Ropsten. These two transactions can be viewed at links:
(Left) https://ropsten.etherscan.io/tx/0x51546492d2d778b6821deba98fa8de30b5b6c2c0681130385f52de74ac97584e,
(Right) https://ropsten.etherscan.io/tx/0x290f57a7c26fed2fdcb2b2c56b3eddb0458286449b1ac0051730f75c979b4079.

12 (right). The figure displays that the target CA tokens are clustered as 9 groups. The Silhouette score and Modularity score are 0.5517 and 0.5027, respectively. We remark here that for both the user clustering and token clustering t-SNE results, we could see that most of the nodes are well clustered while there are still few nodes wrongly distributed, owing to the noise in the observation model.

3) *Implementation of the On-chain Advertisement*

In this part, we show the implementation of the two on-chain advertisement strategies. Our experiment is done on the Ropsten test net, which is a testing environment for Ethereum. Therein, we have used the MEW module (c.f., https://www.myetherwallet.com/) to build transactions, the Remix module(c.f., https://remix.ethereum.org/) for deployment contracts and website *https://etherscan.io* to view the block.

In the left of Fig. 13, we implement how to deliver advertisement in an ETH transaction. We use MEW to build the transaction directly while adding an advertisement message in the input field. Note that to visualize the message we need to convert the string information into hexadecimal. The website *https://etherscan.io* shows us the message in the block. This block will be synchronized to the users' wallet and the wallet will push the message to the target user. Delivering advertisement via smart contract is shown in the right screen of Fig. 13. Therein, a smart contract transaction is generated by the ICO initiator to send message to a group of users. In this approach, the ICO initiator needs to cooperate with the wallet company to register their token address, as well as pushing the valid advertisement message to target users. Experiments show that both strategies are valid in the testing environment.

V. Conclusion

In this work, we have considered community detection on blockchain networks. We respectively studied Bitcoin and Ethereum networks. In particular, for Bitcoin we defined the social network based on transactions and proposed a modified clustering method for the transaction graph. For Ethereum, a bipartite social graph was defined and a novel low-rank clustering method was adopted to cluster users in this graph. We implemented both methods for real blockchain data, visualized and analyzed the community results. We also demonstrated advertisement strategies for delivering on-chain advertisements in the Ethereum network. Our work verified the possibility of applying community detection in different blockchain networks, given that the observation model is not too noisy and sufficient data is provided. How to reduce the effect of heavy noise would be our next task to conquer.

References

[1] T. Alzahrani and K. J. Horadam, "Community detection in bipartite networks: Algorithms and case studies," in *Complex systems and networks*. Springer, 2016, pp. 25–50.

[2] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 34–51.

[3] A. M. Antonopoulos and G. Wood, *Mastering ethereum: building smart contracts and dapps*. O'Reilly Media, 2018.

[4] M. J. Barber, "Modularity and community detection in bipartite networks," *Physical Review E*, vol. 76, no. 6, p. 066102, 2007.

[5] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better—how to make bitcoin a better currency," in *International Conference on Financial Cryptography and Data Security*. Springer, 2012, pp. 399–414.

[6] A. Biryukov and I. Pustogarov, "Bitcoin over tor isn't a good idea," in *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 122–134.

[7] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 486–504.

[8] E. Brinckman, A. Kuehlkamp, J. Nabrzyski, and I. J. Taylor, "Techniques and applications for crawling, ingesting and analyzing blockchain data," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2019, pp. 717–722.

[9] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, p. 37, 2014.

[10] W. Chan and A. Olmsted, "Ethereum transaction graph analysis," in *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2017, pp. 498–500.

[11] T.-H. Chang and D. Svetinovic, "Improving bitcoin ownership identification using transaction patterns analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[12] M. Conoscenti, A. Vetro, and J. C. De Martin, "Blockchain for the internet of things: A systematic literature review," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–6.

[13] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[14] M. Di Angelo and G. Salzer, "Collateral use of deployment code for smart contracts in ethereum," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2019.

[15] J. DuPont and A. C. Squicciarini, "Toward de-anonymizing bitcoin by mapping users location," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*. ACM, 2015, pp. 139–141.

[16] H. Eotvos Lorand University, Budapest. ELTE bitcoin project website and resources. http://vo.elte.hu/bitcoin. Accessed Jan 10, 2020.

[17] D. Ermilov, M. Panov, and Y. Yanovich, "Automatic bitcoin address clustering," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 461–466.

[18] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *arXiv preprint arXiv:1502.01657*, 2015.

[19] D. Goldsmith, K. Grauer, and Y. Shmalo, "Analyzing hack subnetworks in the bitcoin transaction graph," *arXiv preprint arXiv:1910.13415*, 2019.

[20] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Financial Innovation*, vol. 2, no. 1, p. 24, 2016.

[21] M. Harrigan and C. Fretter, "The unreasonable effectiveness of address clustering," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 2016, pp. 368–373.

[22] D. Hossain, "Community detection and observation in large-scale transaction-based networks," Ph.D. dissertation, Technische Universität Berlin, 2018.

[23] T. H.-D. Huang, P.-W. Hong, Y.-T. Lee, Y.-L. Wang, C.-L. Lok, and H.-Y. Kao, "Soc: hunting the underground inside story of the ethereum social-network opinion and comment," *arXiv preprint arXiv:1811.11136*, 2018.

[24] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software," *PloS one*, vol. 9, no. 6, p. e98679, 2014.

[25] D. Kaminsky, "Black ops of tcp/ip," *Black Hat USA*, vol. 44, 2011.

[26] R. Klusman and T. Dijkhuizen, "Deanonymisation in ethereum using existing methods for bitcoin," 2018.

[27] S. Linoy, N. Stakhanova, and A. Matyukhina, "Exploring ethereum's blockchain anonymity using smart contract code attribution."

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[29] G. Maxwell, "Anonymity of bitcoin transactions: An analysis of mixing services," in *Proceedings of Münster Bitcoin Conference*, 2013, pp. 17–18.

[30] ——, "Coinjoin: Bitcoin privacy for the real world," in *Post on Bitcoin forum*, 2013.

[31] ——, "Coinswap: Transaction graph disjoint trustless trading," *CoinSwap: Transactiongraphdisjointtrustlesstrading (October 2013)*, 2013.

[32] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: characterizing payments among men with no names," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 127–140.

[33] B. Mills, "X2-ventures."

[34] M. H. Miraz and M. Ali, "Applications of blockchain technology beyond cryptocurrency," *arXiv preprint arXiv:1801.03528*, 2018.

[35] J. V. Monaco, "Identifying bitcoin users by transaction behavior," in *Biometric and Surveillance Technology for Human and Activity Identification XII*, vol. 9457. International Society for Optics and Photonics, 2015, p. 945704.

[36] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[37] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[38] ——, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[39] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[40] R. Norvill, B. B. F. Pontiveros, R. State, I. Awan, and A. Cullen, "Automated labeling of unknown contracts in ethereum," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2017, pp. 1–6.

[41] J. Payette, S. Schwager, and J. Murphy, "Characterizing the ethereum address space," 2017.

[42] M. Petrov, "Identification of unusual wallets on ethereum platform," 2019.

[43] T. Pham and S. Lee, "Anomaly detection in bitcoin network using unsupervised learning methods," *arXiv preprint arXiv:1611.03941*, 2016.

[44] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[45] C. Remy, B. Rym, and L. Matthieu, "Tracking bitcoin users activity using community detection on a network of weak signals," in *International conference on complex networks and their applications*. Springer, 2017, pp. 166–177.

[46] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 6–24.

[47] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[48] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain management," *International Journal of Production Research*,

vol. 57, no. 7, pp. 2117–2135, 2019.
[49] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, pp. 459–474.
[50] Q. ShenTu and J. Yu, "Research on anonymization and de-anonymization in the bitcoin system," *arXiv preprint arXiv:1510.07782*, 2015.
[51] ——, "Transaction remote release (trr): A new anonymization technology for bitcoin," *arXiv preprint arXiv:1509.06160*, 2015.
[52] S. Somin, G. Gordon, and Y. Altshuler, "Social signals in the ethereum trading network," *arXiv preprint arXiv:1805.12097*, 2018.
[53] M. Spagnuolo, F. Maggi, and S. Zanero, "Bitiodine: Extracting intelligence from the bitcoin network," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 457–468.
[54] H. Sun, N. Ruan, and H. Liu, "Ethereum analysis via node clustering," in *International Conference on Network and System Security*. Springer, 2019, pp. 114–129.
[55] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.
[56] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
[57] H.-T. Wai, S. Segarra, A. E. Ozdaglar, A. Scaglione, and A. Jadbabaie, "Community detection from low-rank excitations of a graph filter," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4044–4048.
[58] L. Wang, S. Ding, and H. Jia, "An improvement of spectral clustering via message passing and density sensitive similarity," *IEEE Access*, vol. 7, pp. 101 054–101 062, 2019.
[59] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
[60] Y. Yanovich, P. Mischenko, and A. Ostrovskiy, "Shared send untangling in bitcoin," *The Bitfury Group white paper*, 2016.
[61] C. Zhou, L. Feng, and Q. Zhao, "A novel community detection method in bipartite networks," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 1679–1693, 2018.
[62] J. Zhu, P. Liu, and L. He, "Mining information on bitcoin network data," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 999–1003.

Sissi Xiaoxiao Wu received the B.Eng. degree in electronic information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2005, the M.Phil. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2009, and the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong (CUHK), Hong Kong, in 2013.,From December 2013 to November 2015, she was a Postdoctoral Fellow in the Department of Systems Engineering and Engineering Management, CUHK. From December 2015 to March 2017, she was a Postdoctoral Fellow in the Signal, Information, Networks and Energy Laboratory supervised by Prof. A. Scaglione of Arizona State University, Tempe, AZ, USA. Since March 2017, she has been an Assistant Professor at the Department of Communication and Information Engineering, Shenzhen University, Shenzhen, China. Her research interests are in wireless communication theory, optimization theory, stochastic process, and channel coding theory, and with a recent

Zixian Wu is currently pursuing a M.Eng degree in communication and information engineering at Shenzhen University. Prior, he received his B.Eng. degree in electronic and information engineering from Shenzhen University, China, in 2019. His research interests are in machine learning, data mining.

Shihui Chen is currently pursuing a M.Eng. degree in Electronics and Communication Engineering at Shenzhen University. Prior, he received his B.Eng. degree in the measurement and control technology and instrument from the Nanchang Institute of Technology, Nanchang, China, in 2017. His research interests are in data mining and blockchain technology.

Gangqiang Li is currently pursuing a PhD degree in communication and information engineering at Shenzhen University. Prior, he received his B.Eng. degree in electronic engineering from the Henan University of Urban Construction, Pingdingshan, China, in 2014, and the M.Eng. degree in Control Science and Engineering, Shenzhen University, Shenzhen, China, in 2017. His research interests are in machine learning, data mining and distributed protocols.

Shengli Zhang [corresponding author] received the B.Eng. degree in electronic engineering and the M.Eng. degree in communication and information engineering from the University of Science and Technology of China, Hefei, China, in 2002 and 2005, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, China, in 2008. After that, he joined the Communication Engineering Department, Shenzhen University. He is currently a Full Professor. From March 2014 to March 2015, he was a Visiting Associate Professor with Stanford University. He is the pioneer of Physical-layer network coding. He has authored or coauthored more than 20 IEEE top journal papers and ACM top conference papers, including IEEE JSAC, IEEE TWC, IEEE TMC, IEEE TCom, and ACM Mobicom. His research interests include physical layer network coding, interference cancellation, and cooperative wireless networks. He served as an Editor for the IEEE TVT, IEEE WCL, and IET Communications. He has also served as TPC member in several IEEE conferences, including IEEE Globecom2016, Globecom2014, ICC2015, ICC2014, WCNC2012, and WCNC2014.