Xiaofeng Chen
Hongyang Yan
Qiben Yan
Xiangliang Zhang (Eds.)

LNCS 12486

Machine Learning for Cyber Security

Third International Conference, ML4CS 2020 Guangzhou, China, October 8–10, 2020 Proceedings, Part I

1 Part I

Springer

Attribute Propagation Enhanced Community Detection Model for Bitcoin De-anonymizing

Jiming Wang[2], Xueshuo Xie[1], Yaozheng Fang[2], Ye Lu[2], Tao Li[1,3](B),
and Guiling Wang[4]

[1] College of Computer Science, Nankai University, Tianjin 300350, China
litao@nankai.edu.cn
[2] College of Cyber Science, Nankai University, Tianjin 300350, China
[3] Tianjin Key Laboratory of Network and Data Security Technology,
Tianjin 300350, China
[4] New Jersey Institute of Technology, Newark, NJ 07102, USA

Abstract. Bitcoin is a kind of decentralized cryptocurrency on a peer-to-peer network. Anonymity makes Bitcoin widely used in online payment but it is a disadvantage for regulatory purposes. We aim to de-anonymize Bitcoin to assist regulation. Many previous studies have used heuristic clustering or machine learning to analyze historical transactions and identify user behaviors. However, the accuracy of user identification is not ideal. Heuristic clustering only uses the topological structure of the transaction graph and ignores many transaction information, and supervised machine learning methods are limited by the size of labeled datasets. To identify user behaviors, we propose a community detection model based on attribute propagation, combining the topological structure of the transaction graph and additional transaction information. We first parse the transaction data of public ledger and construct a bipartite graph to describe correlations between addresses and transactions. We also extract address attributes from historical transactions to construct an attributed graph with the previous bipartite graph. Then, we design an adaptive weighted attribute propagation algorithm named AWAP running on the attributed graph to classify bitcoin addresses, and further identify user behaviors. Extensive experiments highlight that the proposed detection model based on AWAP achieves 5% higher *accuracy* on average compared to state-of-the-art address classification methods in Bitcoin. AWAP also achieves 25% higher *F-score* on average compared to previous community detection algorithms on two datasets.

Keywords: Bitcoin anonymity · Community detection · Attribute propagation

1 Introduction

Bitcoin was proposed by Satoshi Nakamoto in 2008 [12]. As a global decentralized cryptocurrency, Bitcoin has received extensive attention because its anonymity

X. Chen et al. (Eds.): ML4CS 2020, LNCS 12486, pp. 607–622, 2020.
https://doi.org/10.1007/978-3-030-62223-7_53

can protect user privacy. In practice, users do not need any real-world identity registration information to join the bitcoin system, and each user is uniquely identified by a pseudonym. Bitcoin's anonymity ensures that the real identity of a trader is not revealed and thus attracts a large number of users for Bitcoin. However, anonymity also makes Bitcoin as circulating currency for many illegal activities. Bitcoin has been widely used in ransomware, thefts and scams [1,21], such as the black market Silk Road [3]. From the perspective of regulatory purposes, it is important and meaningful to understand the anonymity of the Bitcoin system. On the one hand, a healthy cryptocurrency system needs to support technical legal investigations to ensure the safety and legality of transactions. On the other hand, the cryptocurrency system should provide sufficient anonymity to protect user privacy. In this paper, we focus on the de-anonymization of Bitcoin to support regulation. The question we are exploring is how much anonymity the bitcoin system provides, and whether we can reveal user behaviors by analyzing their relationships through historical transactions.

In Bitcoin, a transaction is a transfer record between addresses. The transaction contains the connection between addresses and also some additional transaction information. Each transaction needs to be recorded on a public ledger to prove its validity. Therefore, a large number of transactions are published on the ledger. Using these public transactions, we can construct a transaction graph to track user transactions and further reveal user behaviors. Thus, the anonymity of the Bitcoin system is pseudo-anonymity. For Bitcoin de-anonymizing, graph-based classification methods of Bitcoin addresses can construct a transaction graph from public transaction records and use address heuristic clustering to complete address user mapping [18,29]. This mapping is conducive to transaction traceability and statistical feature analysis. But this method only considers the connection between addresses, ignoring many additional transaction information. Recently, some other methods use supervised learning [20] to classify bitcoin addresses. But these methods are limited by the size of the labeled dataset, so it is not suitable for large-scale analysis of historical transactions.

Furthermore, community detection based on graph is an important research topic in data mining. The goal of community detection is to classify closely connected nodes into a group, so that the nodes in the community are tightly connected, and the connections between the communities are sparse. Traditional community detection is based on the topological structure. In recent years, many studies have added node attributes to a graph to form an attributed graph. The community detection on attributed graph comprehensively considers the topological structure and node attributes, and widely used in user similarity analysis and content recommendation system in social networks. This work proposes a novel attribute propagation enhanced community detection method to complete the classification of Bitcoin addresses and further de-anonymize. Although promising, it is a challenging task to analyze the large-scale historical transactions in the Bitcoin system. One challenge is that we need to extract the features of a specific address from the massive historical transactions. Another challenge is that we need to comprehensively use topological structure and node attributes although attributes sometimes mismatch with topology and different types of attributes have different contributions to the result of community detection.

To address the above challenges, we design a novel model to de-anonymize in the bitcoin system based on attributed graph community detection. First, we parse the transaction data of public ledger and construct a bipartite graph [6] to describe correlations between addresses and transactions. We also extract address attributes from historical transactions and obtain an attributed graph. Then, we design an adaptive weighted attribute propagation algorithm named AWAP running on the attributed graph to classify bitcoin addresses. We treat the transaction attributed graph as a dynamic system. The attributes are transmitted between nodes using the topology as a medium to affect each other. We call the process *attribute propagation*. We use the propagation results to analyze the correlation between the nodes and further reveal user behaviors. Finally, we test the model on the benchmark labeled data set. The experimental results show that our method has higher accuracy and outperforms state-of-the-art methods. In summary, this paper makes the following contributions:

- We design a community detection model based on attribute propagation, comprehensively considering the topology of the graph and node attributes, and leveraging the attribute propagation to complete the node classification.
- We propose an adaptive weighted attribute propagation algorithm based on an attributed graph, which can maintain a dynamic attributed graph as the dynamic propagation of attributes in Bitcoin.
- We present a transaction parser to generate the features of bitcoin addresses and construct an attributed graph.

2 Background and Motivation

2.1 Bitcoin

The Bitcoin system can be viewed as a composition of large-scale transactions. Each transaction can hold multiple inputs and multiple outputs. When making a payment, a user signs the transaction with his private key to prove his ownership of the bitcoins. Transactions record relations between addresses and other information such as transaction fees and generation time of blocks. We can analyze the whole Bitcoin system by traversing all the transactions and extracting useful information including addresses relations and attributes from transactions .

The identities in Bitcoin are private keys. Each private key generates a public key and some addresses for public identification. These addresses assure user anonymity since they contain no links to a person. An address is called a pseudonym. According to [12], the pseudonym mechanism guarantees complete anonymity on two conditions. One is that the pseudonym has no connection with the real world, and the other is to use a new pseudonym for each transaction. But in fact, few people follow such rules [20]. In this way, using the information extracted from the Bitcoin transaction, we can potentially reveal the activities that the pseudonym participated in and eventually de-anonymize Bitcoin users.

2.2 Community Detection

Community detection is one of the major topics in data mining. Community detection helps discover the structural characteristics, such as functional modules of protein-protein interaction networks [16] or groups of people with similar interests in social networks [24]. Graph as a data structure is popularly used to model the structural relationship between objects in many applications. In addition to the topological structure, nodes are usually associated with attributes. We can add node attributes to a graph to form an attributed graph. Community detection on attributed graph aims to discover groups with common properties, such as similarity among group members or densely connected structure.

Most community detection methods only focus on graph topology or node attributes separately. Examples of topological structure-based methods include modularity [2], spectral clustering [10] and non-negative matrix factorization [8], while node attribute-based methods include k-SNAP [22]. Both topological structure and node attributes provide key information for community detection. It is unwise to ignore any of them. However, specified in [14], there is no evidence that topological structure and node attributes share the same characteristics in any case. In other words, node attributes may unexpectedly mismatch with topology. Different types of attributes have different degrees of contribution to community detection.

The Bitcoin system is dynamic, so we also treat attributed graph constructed from Bitcoin as a dynamic graph. The establishment of the connection between nodes is accompanied by the propagation of attributes. As connections increase, nodes receive attribute information from other nodes and send their own attribute to others. This process is similar to information propagation, which is a fundamental factor in the study of social networks [11].

2.3 Challenges and Goals

For the Bitcoin system, the data size of historical transactions is huge. It is time consuming to construct transaction graphs and extract address features. At the same time, the traditional bipartite address-transaction graph is not suitable for community detection. So the existing community detection algorithm is not ideal for de-anonymizing Bitcoin. Motivated by information propagation, we aim to address the following challenges:

- How to efficiently and accurately extract the features of a specific address from the massive bitcoin transaction history?
- How to construct an attributed graph that can be applied to community detection since bipartite address-transaction graph is not suitable for community detection?
- How to combine topological structure and node attributes together since node attributes sometimes mismatch with topology and different types of attributes have different contributions to community detection?

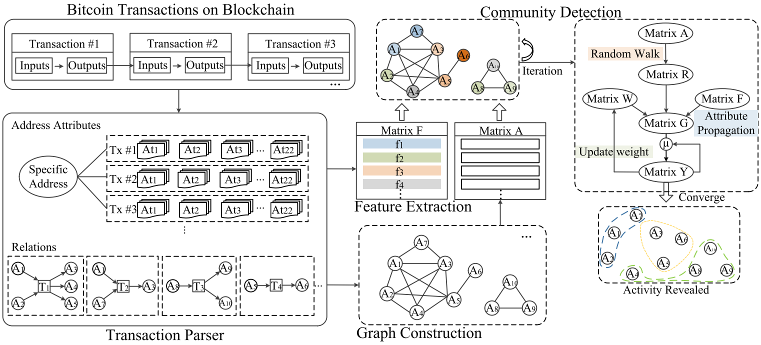


Fig. 1. An overview of our model.

3 System Model

3.1 Model Overview

In Fig. 1, our model starts with getting bitcoin transaction data on the blockchain. The transaction parser parses raw transaction data into the connections between addresses and some addresses attributes. The connections are described as a bipartite graph. In the graph construction stage, the bipartite graph is converted into a community graph suitable for community detection. In the feature extraction stage, we build a feature vector for a specific address based on the address attributes provided by the transaction parser. We then combine the community graph and the address features to form an attributed graph for community detection. The detection completes addresses classification and reveals user behaviors.

3.2 Transaction Parser

The size of Bitcoin data on the blockchain is huge. The transaction parser needs to efficiently obtain data and complete the parsing work. [18] employed a forked version of *bitcointools* using LevelDB. [5] used Armory to parse data on the blockchain. [7] designed a platform for parsing and analyzing blockchain. Taking into account the factors of time consumption and the specific information we need, we use the API provided by *blockchain.info* to implement our own parser. As mentioned in Background, there is lots of information in a transaction. Some information such as Transaction Hash, ScriptSig is not what we need. Parser needs to filter out useless information. Parser also extracts connections between addresses and constructs a bipartite address-transaction graph.

3.3 Graph Construction

The upper part of Fig. 2 shows the bipartite address-transactions graph constructed by the parser, where A represents an address and T represents a transaction. The bipartite graph can show the relationship between addresses and

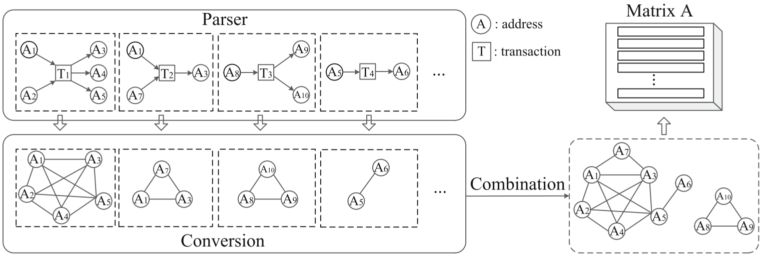


Fig. 2. The construction process of a community graph

transactions. However, it cannot be applied in community detection, because the nodes in a community graph should be of the same type. So we convert the bipartite graph to the lower part of Fig. 2. Our construction principle is to delete transaction nodes T and increase the edges between related nodes. We convert a directed bipartite graph into an undirected graph. So the process may lose some information. But our goal is to classify the nodes. The direction of the edges between the nodes is not important for the result, as long as the edges can reflect the connections between nodes.

We use a 2-tuple $G = (V, E)$ to represent a community graph, where $V = \{v_i | i \in [1, N]\}$ is the set of nodes, $E = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ is the set of edges. The graph here is an undirected graph. The adjacency matrix A of the graph G can be computed as:

$$A_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \text{ or } (v_j, v_i) \in E, \text{ or } i = j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

3.4 Features Extraction

In this part, we use the address attributes provided by the parser to generate address features vector. The process of feature extraction is shown in Fig. 3. What features we extract depends on the characteristics of different types of transactions. For example, mining pool transactions have no inputs, and gambling transactions often have many inputs, so we select the average number of inputs of transactions as an address feature. In other words, the address features we choose can reflect the transaction behaviors. Eventually, We select 22 features for each address. Some of these features are shown in Table 1.

Considering the attribute propagation process, we extend the feature value vector to a binary-valued vector to speed up the propagation. We record all the vectors in the feature matrix F, then a 3-tuple $G = (V, E, F)$ can be used to represent an attributed graph.

3.5 Community Detection

Next we define the process of attribute propagation. We calculate the probability that the attribute of a node propagates to another by random walk. The one

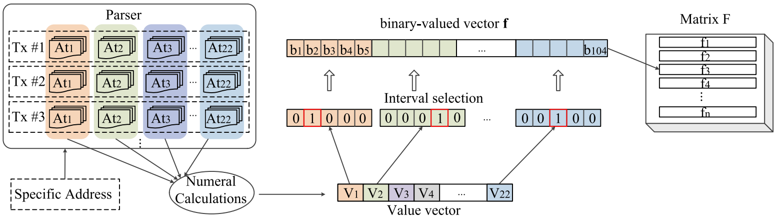


Fig. 3. The process of feature extraction

Table 1. Some features extracted from bitcoin transactions

Features	Description
n_{tx}	The number of transactions
n_{spent}	The number of spent transactions
BTC_{spent}	Total spent bitcoin
Fee	Transaction fee
$Balance$	The balance in an address
$Lifetime$	The duration between the first transaction and the last transaction
f_{tx}	The frequency of transactions

step transition probability P_{ij} denotes the probability of a node i arriving at node j at time $t = 1$ given that i started at time $t = 0$.

$$P_{ij} = P_{t=1|0}(j|i) = \frac{A_{ij}}{\sum_{a=1}^{N} A_i a} \tag{2}$$

Further, we treat t as a random variable from $[0, \infty)$ and follows a geometric distribution, then the t step transition probability:

$$P_{t|0}(j|i) = \sum_{s=0}^{\infty} P_{s|0}(j|i) \cdot P(t = s) = \sum_{s=0}^{\infty} P_{s|0}(j|i) \cdot \lambda(1 - \lambda)^{s} \tag{3}$$

We assume the start point of a random walk is chosen at random, the probability that node j receives attribute from node i can be calculated as:

$$P_{0|t}(i|j) = \frac{P_{t|0}(j|i)}{\sum_{a=1}^{N} P_{t|0}(j|a)} \tag{4}$$

We obtain matrix R where $R_{ij} = P_{0|t}(i|j)$. The attribute propagation can be written in matrix form G:

$$G = F^{\top} R \tag{5}$$

As mentioned in Sect. 2, attributes may mismatch with topology and different types of attributes have different degrees of contribution to community detection, we use an adaptive weight matrix W to control the contribution of attributes. W is a diagonal matrix with $W_{ii} = w_i$, $\forall i \in [1, m]$. m is the dimension of feature vectors. We initialize $w_1, ..., w_m = 1.0$. Then G is rewritten as:

$$G = WF^\top R \tag{6}$$

Specified in [11], the key element of community detection based on information propagation is the assumption of community consistency. When the propagation reaches stability, nodes in the same community are likely receive the same amount of attribute propagation. We use ϕ_i to denote the attribute node i receive. Then, we can get μ_k:

$$\mu_k = \frac{\sum_{i=1}^{m} \phi_i}{m} \tag{7}$$

μ_k denotes the expectation of attribute propagation received by nodes in the community C_k. m denotes the number of nodes in the community C_k. Considering a membership matrix Y and K is the number of communities, we have:

$$E[\phi_i] = \sum_{k=1}^{K} \mu_k \cdot Y_{ik},\ where\ Y_{ik} = \begin{cases} 1, & i \in C_k \\ 0, & i \notin C_k \end{cases} \tag{8}$$

where $E[\phi_i]$ denotes the expectation of attribute propagation received by node i. We use g_i to denote the i-th row of matrix G, which represents the actual attribute propagation obtained by node i. Obviously, $g_i = \phi_i$. Then, community detection based on attribute propagation can be obtained by solving the following optimization:

$$\underset{Y, \mu_k}{\arg\min} \sum_{i=1}^{N} \| g_i - E[\phi_i] \|_2^2 \tag{9}$$

Y is the result we want, indicating the relationship between the node and the community. First, we solve μ_k with Y and (7):

$$\mu_k = \sum_{i=1}^{N} \frac{g_i \cdot Y_{ik}}{\sum_{j=1}^{N} Y_{jk}} \tag{10}$$

Take μ_k to (8) and (9), we can set matrix Y by calculating the first K eigenvectors of matrix $R^\top W F^\top F W^\top R$ according to [25].

Then, during the iteration, we update μ and Y with:

$$\mu_k^{t+1} = \sum_{i=1}^{N} \frac{g_i \cdot Y_{ik}^t}{\sum_{j=1}^{N} Y_{jk}^t} \tag{11}$$

$$Y_{ik}^{t+1} = \begin{cases} 1, & \forall m \in [1, K], \|\mu_k^t - g_i\| \leq \|\mu_m^t - g_i\| \\ 0, & otherwise \end{cases} \tag{12}$$

Finally, we use a vote mechanism, similar to [30], to adjust attribute weights. We assume $w_1^t, ...w_m^t$ are the attribute weights in the t^{th} iteration. The weight of attribute a_i in the $(t+1)^{th}$ iteration is computed as:

$$w_i^{t+1} = \frac{1}{2}(w_i^t + \Delta w_i^t) \tag{13}$$

We use a vote mechanism to accurately calculate Δw_i^t. If nodes in the same community share the same value of attribute a_i, it means attribute a_i can reflect the characteristics of the community, then the weight w_i of a_i increase. If nodes in the same community have a random distribution on values of attribute a_i, the weight w_i of a_i decrease. The vote process can be computed as:

$$vote_i(v_p, v_q) = \begin{cases} 1, & if\ v_p, v_q\ share\ the\ same\ value\ on\ a_i \\ 0, & otherwise \end{cases} \tag{14}$$

and Δw_i^t is calculated as

$$\Delta w_i^t = \frac{\sum_{j=1}^{k} \sum_{v \in V_j} vote_i(c_j, v)}{\frac{1}{m} \sum_{p=1}^{m} \sum_{j=1}^{k} \sum_{v \in V_j} vote_p(c_j, v)} \tag{15}$$

where V_j denotes the nodes in community j and c_j denotes a virtual node with expectation attributes of community j. The algorithm of community detection based on AWAP is summarized in Algorithm 1.

Algorithm 1. Adaptive Weighted Attribute Propagation

```
Input: adjacency matrix $A$; feature matrix $F$; number of clusters $K$;parameter $\lambda$;
Output: detected communities indicated by $Y$;
Initialize $w_1 = w_2 = ... = w_m = 1.0$;
Calculate $R$ matrix with random walk(4) ;
Calculate $G$ with (6);
Calculate the first $K$ eigenvectors and initialize $Y$;
initialize $\mu$ with (10);
while not converged do
    update $Y$ with $\mu$ and $W$ by (12);
    update $\mu$ with $Y$ by (11);
    update weights $w_1, w_2, ..., w_m$ with (13);
end while
Return $Y$;
```

4 Evaluation

In this section, we demonstrate the effectiveness of our proposed model in terms of bitcoin address classification and community detection. We choose 4 Bitcoin address classification methods and 7 previous community detection algorithms as the baselines. The evaluation is concerning the following questions:

- How does the Bitcoin address classification performance when using AWAP as the community detection compared with state-of-the-art methods?
- How does the AWAP performance in community detection compared with previous community detection algorithms?
- Why AWAP can increase the accuracy, F-score, and Jaccard?
- How do parameters and weight distribution influence performance?

4.1 Experimental Setup

Datasets. For Bitcoin de-anonymizing, our model can run directly on the raw Bitcoin transaction data. But in this paper, we select a labeled dataset to compare the performance of our model with other methods. We use a five categories bitcoin addresses dataset in [6]. And, we also use the parser to extract address feature vectors from historical transactions to further improve the dataset. Finally, our dataset consists: Exchange: 10413 addresses; Gambling: 10479 addresses; DarkNet Marketplace: 10593 addresses; Mining Pool: 10498 addresses; Service: 10597 addresses. For each address in the dataset, there is a 104-dimensional feature vector.

Measured Metrics. We use 3 metrics *F-score*, *Jaccard similarity* and *NMI* to evaluate the performance of the detected communities C using the ground-truth communities C^*, and 3 metrics *Accuracy*, *Precision* and F_1-score to evaluate the performance of bitcoin address classification.

$$F_{score}(C, C^*) = \sum_{C_i \in C} \frac{|C_i|}{\sum_{C_j \in C} |C_j|} \max_{C_j^* \in C^*} F_{score}(C_i, C_j^*) \tag{16}$$

$$Jac(C, C^*) = \sum_{C_j^* \in C^*} \frac{\max_{C_i \in C} Jac(C_i, C_j^*)}{2\,|C^*|} + \sum_{C_i \in C} \frac{\max_{C_i^* \in C^*} Jac(C_i, C_j^*)}{2\,|C|} \tag{17}$$

$$NMI(C, C^*) = \frac{\sum_{C_i, C_j^*} p(C_i, C_J^*)(log\ p(C_i, C_j^*) - log\ p(C_i)p(C_j))}{\max(H(C), H(C^*))} \tag{18}$$

$H(C)$ is the entropy of the community C. *Accuracy* is the proportion of correct predictions to total predictions. *Precision* is the proportion of positive predictions to the total positive predictions. *Recall* represents a measure of the completeness of a classifier. F_1-score is the harmonic mean of *Precision* and *Recall.*

$$F_1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{19}$$

The 6 metrics all take values from $[0, 1]$, and larger values indicate better results.

Configuration. Our experiment is tested on the machine with Windows 10, Intel Core 2.20 GHz CPUs, and 16 GB of RAM. Our parser step is implemented in Python 3.7.2. and community detection algorithm is implemented in Matlab.

Table 2. Bitcoin address classification results

Algorithm	Accuracy	Precision	F_1-score
Logistic regression	0.85	0.87	0.85
LightGBM	0.92	**0.92**	**0.91**
BAGC	0.55	0.47	0.50
CP	0.89	0.71	0.79
AWAP	**0.94**	0.85	0.89

Table 3. Community quality comparison on Citeseer and Cora

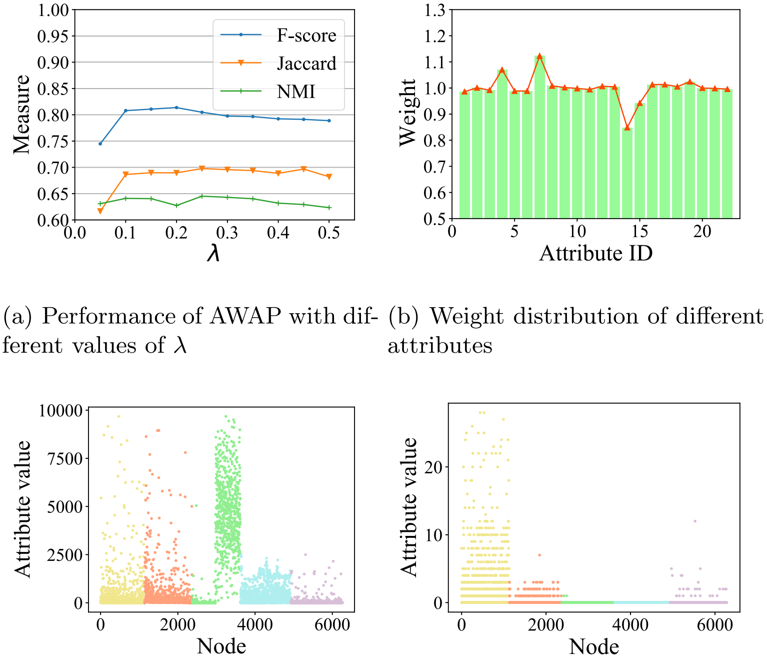
Algorithm	Information	Citeseer			Cora		
		F-score	Jaccard	NMI	F-score	Jaccard	NMI
CNM	Topology	0.1735	0.1094	0.2290	0.4210	0.2315	0.1491
DeepWalk	Topology	0.2699	0.2481	0.0878	0.3917	0.3612	0.3270
Big-CLAM	Topology	0.5114	0.0872	0.2197	0.4829	0.2340	0.2919
Circles	Topology + Attributes	0.3405	0.1867	0.0024	0.3595	0.1810	0.0064
CP	Topology + Attributes	0.6918	0.4991	0.4314	0.6770	0.5168	0.4863
CODICIL	Topology + Attributes	0.5953	0.4041	0.3392	0.5857	0.3947	0.4254
CESNA	Topology + Attributes	0.5240	0.1158	0.1158	0.6059	0.3254	0.4671
AWAP	Topology + Attributes	**0.7134**	**0.4570**	**0.5205**	**0.7583**	**0.5875**	**0.5683**

4.2 Model Performance

Address Classification Performance. We summarize the address classification results of different methods in Table 2. We also use some other community detection algorithms BAGC [26] and CP [11] to classify bitcoin addresses. The results show that our method has higher accuracy, while precision and F_1-score are inferior to LightGBM. The reason why accuracy is high and precision is slightly lower may be because the classification of most addresses is correct, but one of the categories of addresses is wrong. We further analyze accuracy, precision, and F_1-score of the five address types, and find that some Services addresses are identified as DarkNet addresses, resulting in a decrease in the accuracy of Exchange and Mining, and precision of Exchange.

Community Detection Performance. We evaluate our model from the perspective of address classification above, and this part we evaluate the community detection results of our community detection algorithm (AWAP). We conduct investigation on Citeseer and Cora. We consider three metrics: *F-score*, *Jaccard similarity* and *NMI*. *F-score* mainly describes the accuracy of detected communities, while *Jaccard* is a statistic used for comparing the similarity of detected communities and the ground truth, and *NMI* offers an entropy measure of the overall matching. The results are shown in Table 3.

The results show that our AWAP outperforms the baselines. By comparison, it is shown that the algorithm that comprehensively considers the topology



(a) Performance of AWAP with different values of λ

(b) Weight distribution of different attributes

(c) the distribution of attribute Total Sent BTC in nodes

(d) the distribution of attribute Transaction Fee in nodes

Fig. 4. The performance on: (a) different parameters; (b) weight distribution; (c) attribute distribution; (d) attribute distribution.

and node attributes is indeed superior to the algorithm that only considers the topology. Dynamically adjusting attribute weights can also improve the performance of community detection. Our method increases the contribution of specific attributes and reduces the contribution of attributes that mismatch with the topology or do not match the expected attributes of the community. We will analyze the distribution of values of different weighted attributes in different communities in the Discuss section. On the other hand, the method of attribute propagation is based on information propagation. The topology is used as a medium to transfer node attributes. Naturally, the topology structure and the node attributes are combined together. For the topologically independent node, the node attributes are automatically compared with the community expected attributes.

Discussion. In the experiment, we fixed the value of λ to 0.2. In this part, we first investigate the influence of λ on the performance of our AWAP. The value of λ is varied from 0.05 to 0.5 with step size of 0.05. The results are summarized in Fig. 4(a). We can observe that the three metrics all reach the maximum value when the value of λ is between 0.2 and 0.25 and then decrease slowly overall when λ get larger. This may be because when λ is at this interval, the attributes are more easily and thoroughly propagated in the topology, which is conducive to improve the effectiveness of community detection.

Next, we discuss the attribute weights determined by AWAP. Aggregating 104-dimensional feature weight vectors to calculate the weights of 22 attributes, we get Fig. 4(b). Two attributes with the highest weights are the number of payback transactions N_{pt} and bitcoins an address total sent BTC_{sent} while two attributes with the lowest are related to transaction fees Fee. Figure4(c) and Fig. 4(d) are the value distribution of BTC_{sent} and Fee in nodes respectively. Different colors indicate different communities, from left to right are Exchange, Gambling, DarkNet, Mining and Services. BTC_{sent} usually reflects the scale of transactions. We can observe that BTC_{sent} of DarkNet is concentrated in the high-value area, BTC_{sent} of Services is generally low, and the other three types of attribute values also have their own distributions. BTC_{sent} can distinguish DarkNet and Services very well which is a good basis for community detection. So its weight is high. Fee often has a fixed lower limit in a transaction, and few addresses are willing to pay higher fees to the miners. In the figure, 96% of nodes have a Fee less than 10. Different types of addresses have similar Fee. So it is not able to provide a basis for community detection.

5 Related Work

Bitcoin de-anonymization methods often use heuristic clustering to construct the one-to-many mapping from entities to addresses based on the properties of the Bitcoin protocol. [17] derive two topological structures from Bitcoin's public transaction history and combine these structures with external information to investigate an alleged theft of Bitcoins. [18] use multiple inputs heuristic clustering on the full bitcoin transaction graph and answer a variety of questions about the typical behavior of users. [23] use transaction-specific features to achieve 70% accuracy for classifying addresses into several types. [15] introduces the notion of transaction motifs and finally achieve more than 80% accuracy. [6] analyze the information revealed by the pattern of transactions in the neighborhood of a given entity transaction and achieve 85% accuracy for the Logistic Regression algorithm and 92% for LightGBM.

Based on the graph, the algorithm of community detection can be categorized into two types. One only considers the topology of the graph, while the other comprehensively considers the topological structure and node attributes. [2] uses a heuristic method to extract the community structure of large networks based on modularity optimization. [27] present BIGCLAM, an overlapping community detection method that scales to large networks of millions of nodes and edges based on topology. [4] presents a hierarchical agglomeration algorithm for detecting community structure. [13] proposes DeepWalk which is a structure-only representation learning method. DeepWalk uses local information obtained from truncated random walks to learn latent representations.

A model is proposed for detecting circles that combine network structure as well as user profile information [9]. They learn members and user profile similarity metric for each circle. A Bayesian probabilistic model (BAGC) for attributed graph clustering is proposed in [26]. The model provides a principled and natural framework for capturing both structural and attribute aspects of a graph, avoiding the artificial design of a distance measure. [28] develop CESNA for overlapping community detection which has a linear runtime in the network size. [11] treats a network with a dynamic system and uses the principle of information propagation to integrate the structure and contents in a network. [19] design a mechanism for fusing content and link similarity. They present a biased edge sampling procedure and finally get an edge set.

6 Conclusion

In this paper, we formulate the Bitcoin de-anonymity as a problem of Bitcoin addresses classification to explore user trading behaviors. To achieve this goal, we first construct an attributed graph based on bitcoin historical transactions. And then, we propose a community detection method based on attribute propagation that comprehensively uses the topological structure and node attributes of the attributed graph. The method also dynamically adjusts weights of different attributes. Our approach provides sound results on public datasets. Our proposed model can efficiently resolve the problems of bitcoin addresses classification and community detection. An interesting direction is to apply more community detection algorithms to explore bitcoin user behaviors and further contribute to the healthy development of Bitcoin.

References

1. Bartoletti, M., Pes, B., Serusi, S.: Data mining for detecting bitcoin Ponzi schemes. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), pp. 75–84. IEEE (2018)
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
3. Christin, N.: Traveling the silk road: a measurement analysis of a large anonymous online marketplace. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 213–224 (2013)
4. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
5. Fleder, M., Kester, M.S., Pillai, S.: Bitcoin transaction graph analysis. arXiv preprint arXiv:1502.01657 (2015)
6. Jourdan, M., Blandin, S., Wynter, L., Deshpande, P.: Characterizing entities in the bitcoin blockchain. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 55–62. IEEE (2018)
7. Kalodner, H., Goldfeder, S., Chator, A., Möser, M., Narayanan, A.: BlockSci: design and applications of a blockchain analysis platform. arXiv preprint arXiv:1709.02489 (2017)

8. Kamuhanda, D., He, K.: A nonnegative matrix factorization approach for multiple local community detection. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 642–649. IEEE (2018)
9. Leskovec, J., Mcauley, J.J.: Learning to discover social circles in ego networks. In: Advances in Neural Information Processing Systems, pp. 539–547 (2012)
10. Li, X., Kao, B., Ren, Z., Yin, D.: Spectral clustering in heterogeneous information networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4221–4228 (2019)
11. Liu, L., Xu, L., Wangy, Z., Chen, E.: Community detection based on structure and content: a content propagation perspective. In: 2015 IEEE International Conference on Data Mining, pp. 271–280. IEEE (2015)
12. Nakamoto, S., Bitcoin, A.: A peer-to-peer electronic cash system (2008). https://bitcoin.org/bitcoin.pdf
13. 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)
14. Qin, M., Jin, D., Lei, K., Gabrys, B., Musial-Gabrys, K.: Adaptive community detection incorporating topology and content in social networks. Knowl. Based Syst. **161**, 342–356 (2018)
15. Ranshous, S., et al.: Exchange pattern mining in the bitcoin transaction directed hypergraph. In: Brenner, M., et al. (eds.) FC 2017. LNCS, vol. 10323, pp. 248–263. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70278-0_16
16. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabási, A.L.: Hierarchical organization of modularity in metabolic networks. Science **297**(5586), 1551–1555 (2002)
17. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) Security and Privacy in Social Networks, pp. 197–223. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
18. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_2
19. Ruan, Y., Fuhry, D., Parthasarathy, S.: Efficient community detection in large networks using content and links. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1089–1098 (2013)
20. Shao, W., Li, H., Chen, M., Jia, C., Liu, C., Wang, Z.: Identifying bitcoin users using deep neural network. In: Vaidya, J., Li, J. (eds.) ICA3PP 2018. LNCS, vol. 11337, pp. 178–192. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05063-4_15
21. Sun Yin, H.H., Langenheldt, K., Harlev, M., Mukkamala, R.R., Vatrapu, R.: Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain. J. Manag. Inf. Syst. **36**(1), 37–73 (2019)
22. Tian, Y., Hankins, R.A., Patel, J.M.: Efficient aggregation for graph summarization. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 567–580 (2008)
23. Toyoda, K., Ohtsuki, T., Mathiopoulos, P.T.: Multi-class bitcoin-enabled service identification based on transaction history summarization. In: 2018 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), pp. 1153–1160. IEEE (2018)

24. Watts, D.J., Dodds, P.S., Newman, M.E.: Identity and search in social networks. Science **296**(5571), 1302–1305 (2002)
25. Xu, L., White, M., Schuurmans, D.: Optimal reverse prediction: a unified perspective on supervised, unsupervised and semi-supervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1137–1144 (2009)
26. Xu, Z., Ke, Y., Wang, Y., Cheng, H., Cheng, J.: A model-based approach to attributed graph clustering. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 505–516 (2012)
27. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 587–596 (2013)
28. Yang, J., McAuley, J., Leskovec, J.: Community detection in networks with node attributes. In: 2013 IEEE 13th International Conference on Data Mining, pp. 1151–1156. IEEE (2013)
29. Zhao, C., Guan, Y.: A graph-based investigation of bitcoin transactions. In: Peterson, G., Shenoi, S. (eds.) DigitalForensics 2015. IAICT, vol. 462, pp. 79–95. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24123-4_5
30. Zhou, Y., Cheng, H., Yu, J.X.: Graph clustering based on structural/attribute similarities. Proc. VLDB Endow. **2**(1), 718–729 (2009)